COLLABORATIVE LEARNING: THEORY, ALGORITHMS, AND APPLICATIONS

By

Kaixiang Lin

A DISSERTATION

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

Computer Science — Doctor of Philosophy

2020

ABSTRACT

COLLABORATIVE LEARNING: THEORY, ALGORITHMS, AND APPLICATIONS

By

Kaixiang Lin

Human intelligence prospers with the advantage of collaboration. To solve one or a set of challenging tasks, we can effectively interact with peers, fuse knowledge from different sources, continuously inspire, contribute, and develop the expertise for the benefit of the shared objectives. Human collaboration is flexible, adaptive, and scalable in terms of various cooperative constructions, collaborating across interdisciplinary, even seemingly unrelated domains, and building large-scale disciplined organizations for extremely complex tasks. On the other hand, while machine intelligence achieved tremendous success in the past decade, the ability to collaboratively solve complicated tasks is still limited compared to human intelligence.

In this dissertation, we study the problem of collaborative learning - building flexible, generalizable, and scalable collaborative strategies to facilitate the efficiency of learning one or a set of objectives. Towards achieving this goal, we investigate the following concrete and fundamental problems: 1. In the context of multi-task learning, can we enforce flexible forms of interactions from multiple tasks and adaptively incorporate human expert knowledge to guide the collaboration? 2. In reinforcement learning, can we design collaborative methods that effectively collaborate among heterogeneous learning agents to improve the sample-efficiency? 3. In multi-agent learning, can we develop a scalable collaborative strategy to coordinate a massive number of learning agents accomplishing a shared task? 4. In federated learning, can we have provable benefit from increasing the number of collaborative learning agents?

This thesis provides the first line of research to view the above learning fields in a unified framework, which includes novel algorithms for flexible, adaptive collaboration, real-world applications using scalable collaborative learning solutions, and fundamental theories for propelling the understanding of collaborative learning.

ACKNOWLEDGMENTS

First and foremost, I would like to thank my advisor, Dr. Jiayu Zhou, for his advice, encouragement, inspirations, and endless support for my research and career. Throughout the past five years at Michigan State University, Dr. Zhou has always influenced me with his curiosity, passion, and persistence of research. He is willing to discuss the grant picture of the research and provide constructive suggestions in the technical details. Meanwhile, despite being creative and productive, he also gives me the freedom to work on a variety of problems, even some are not aligned with his interest. I would like to thank Drs. Jiliang Tang, Zhaojian Li, and Anil K. Jain for being on my thesis committee.

I'm very happy to have had the opportunity to collaborate with the wonderful group of colleagues, faculty, and researchers throughout my Ph.D. For the work presented in this dissertation, I enjoyed working with Dr. Jianpeng Xu, Dr. Inci M. Baytas, Dr. Shuiwang Ji, Dr. Shu Wang, Renyu Zhao, Dr. Zhe Xu, Zhaonan Qu, Dr. Zhaojian Li, Dr. Zhengyuan Zhou and Dr. Jiayu Zhou. I thank them for their contributions and for everything they have taught me. Besides the work presented in this thesis, I also had the pleasure of working with many outstanding researchers, including Liyang Xie, Dr. Fei Wang, Dr. Pang-Ning Tan, Fengyi Tang, Ikechukwu Uchendu, Boyang Liu, Ding Wang, Zhuangdi Zhu, and Dr. Bo Dai. I would like to thank all of my amazing colleagues in ILLIDAN lab: Qi Wang, Dr. Inci M. Baytas, Liyang Xie, Mengying Sun, Fengyi Tang, Boyang Liu, Zhuangdi Zhu, Junyuan Hong, Xitong Zhang and Ikechukwu Uchendu for a collaborative, friendly, and productive environment.

I also want to express my sincere thanks to the amazing colleagues I met during the internships, including Dr. Pinghua Gong, Wei Chen, Guojun Wu, Zhengtian Xu, Hongyu

Zheng, Jintao Ke, Huaxiu Yao, Dan Wang, Lili Cao, Lingkai Yang, Qiqi Wang, Dr. Yaguang Li, Dr. Peng Wang, Dr. Jie Wang, Chao Tao, Dr. Jia Chen, and Dr. Youjie Zhou. Many thanks to Dr. Pinghua Gong, Dr. Peng Wang, for hosting me as an intern at Didi Chuxing in 2017 and 2018. I am also most thankful to Dr. Jia Chen and Dr. Youjie Zhou for their patience and endless help during my internship at Google in 2019.

Finally, I thank my parents, for their unconditional love and support.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ALGORITHMS

Chapter 1

Introduction

Human intelligence is remarkable at collaboration. Besides independent learning, our learning process is highly improved by summarizing what has been learned, communicating it with peers, and subsequently fusing knowledge from different sources to assist the current learning goal. This *collaborative learning* procedure ensures that the knowledge is shared, continuously refined, and concluded from different perspectives to construct a increasingly profound understanding, which can significant improve the the learning efficiency.

On the other hand, machine intelligence still pales in comparison to human in some aspects, despite its phenomenal development in recent years: they in general designed for one specific task, with an isolated, data inefficient, and computationally expensive learning paradigm.

The research goal presented in this dissertation is to build an intelligent system with multiple learning agents that collaboratively resolves one or a set of tasks more efficiently. In particular, we tackle the following challenges in various domains of collaborative learning.

- **Flexible and interactive collaboration.** How can models of multiple learning agents interact to leverage the knowledge from related tasks in a flexible, stable, and interactive way? More concretely, how can we incorporate higher-order interactions into the multiple learning models during training? How can we continuously guide the learning of multiple models and selectively solicit the human expert knowledge to escort

their collaboration interactively?

- **Heterogeneous collaboration.** One limitation in collaborative learning is that the learning models in general, have a homogeneous structure. How can we design collaborative strategies among heterogeneous learning agents to improve the sample-efficiency?

- **Large-scale collaboration.** In practice, an effective and efficient collaboration among a large amount of learning agents is desired. How can we scale the collaboration to thousands of agents?

- **Theoretical guarantee of collaboration.** Besides the practical algorithms and applications, what are the theoretical advantages of collaborative learning? Does the learning benefit from more learning agents?

1.1 Dissertation Contributions

To resolve the aforementioned challenges of collaborative learning, this thesis presents how the collaboration is achieved to improve sample-efficiency in various scenarios. More concretely, the contributions of this thesis are summarized in the following sections.

1.1.1 Model-driven collaboration

We discuss model-driven collaboration in the context of multi-task learning. The first part in this Chapter discusses how do we capture the high-order feature interactions among related tasks collaboratively. Traditional multi-task learning with linear models are widely used in various data mining and machine learning algorithms. One major limitation of

such models is the lack of capability to capture predictive information from interactions between features. While introducing high-order feature interaction terms can overcome this limitation, this approach dramatically increases the model complexity and imposes significant challenges in the learning against overfitting. When there are multiple related learning tasks, feature interactions from these tasks are usually related and modeling such relatedness is the key to improve their generalization. Here, we present a novel Multi-Task feature Interaction Learning (MTIL) framework to exploit the task relatedness from high-order feature interactions. Specifically, we collectively represent the feature interactions from multiple tasks as a tensor, and prior knowledge of task relatedness can be incorporated into different structured regularizations on this tensor. We formulate two concrete approaches under this framework, namely the shared interaction approach and the embedded interaction approach. The former assumes tasks share the same set of interactions, and the latter assumes feature interactions from multiple tasks share a common subspace. We have provided efficient algorithms for solving the two formulations.

The second part in this chapter investigates soliciting and incorporating task relatedness information from human expert to the model, which guides the direction of the model-based collaboration. In the center of MTL algorithms is how the relatedness of tasks are modeled and encoded in learning formulations to facilitate knowledge transfer. Among the MTL algorithms, the multi-task relationship learning (MTRL) attracted much attention in the community because it learns task relationship from data to guide knowledge transfer, instead of imposing a prior task relatedness assumption. However, this method heavily depends on the quality of training data. When there is insufficient training data or the data is too noisy, the algorithm could learn an inaccurate task relationship that misleads the learning towards suboptimal models. To address the aforementioned challenge, we propose a novel interactive

multi-task relationship learning (iMTRL) framework that efficiently solicits partial order knowledge of task relationship from human experts, effectively incorporates the knowledge in a proposed knowledge-aware MTRL formulation. We propose an efficient optimization algorithm for kMTRL and comprehensively study query strategies that identify the critical pairs that are most influential to the learning. We present extensive empirical studies on both synthetic and real datasets to demonstrate the effectiveness of proposed framework.

1.1.2 Data-driven collaboration

In Chapter 3, we discuss data-driven collaboration in the context of reinforcement learning and use the data as a medium to facilitate collaboration among multiple learning agents, which can then largely improve the sample-efficiency.

In this chapter, we first leverage the knowledge distillation to enforce the collaboration among heterogeneous learning agents. The idea of knowledge transfer has led to many advances in machine learning and data mining, but significant challenges remain, especially when it comes to reinforcement learning, heterogeneous model structures, and different learning tasks. Motivated by human collaborative learning, we propose a collaborative deep reinforcement learning (CDRL) framework that performs adaptive knowledge transfer among heterogeneous learning agents. Specifically, the proposed CDRL conducts a novel deep knowledge distillation method to address the heterogeneity among different learning tasks with a deep alignment network. Furthermore, we present an efficient collaborative Asynchronous Advantage Actor-Critic (cA3C) algorithm to incorporate deep knowledge distillation into the online training of agents, and demonstrate the effectiveness of the CDRL framework using extensive empirical evaluation on OpenAI gym.

In addition to knowledge transfer among different tasks, we can further coordinate

different homogeneous learning agents for the same task, which further advances more stable optimization and sample-efficient learning. The main idea is an off-policy learning framework that disentangles exploration and exploitation in reinforcement learning, which build upon the connection between imitation learning and reinforcement learning. The state-of-the-art estimates the optimal action values while it usually involves an extensive search over the state-action space and unstable optimization. Towards the sample-efficient RL, we propose ranking policy gradient (RPG), a policy gradient method that learns the optimal rank of a set of discrete actions. To accelerate the learning of policy gradient methods, we establish the equivalence between maximizing the lower bound of return and imitating a near-optimal policy without accessing any oracles. These results lead to a general off-policy learning framework, which preserves the optimality, reduces variance, and improves the sample-efficiency. We conduct extensive experiments showing that when consolidating with the off-policy learning framework, RPG substantially reduces the sample complexity, comparing to the state-of-the-art.

1.1.3 Large-scale Collaborative Multi-agent Learning

In this chapter, we apply collaborative multi-agent reinforcement learning to a real-world fleet management application, which is an essential component for online ride-sharing platforms. Large-scale online ride-sharing platforms have substantially transformed our lives by reallocating transportation resources to alleviate traffic congestion and promote transportation efficiency. An efficient fleet management strategy not only can significantly improve the utilization of transportation resources but also increase the revenue and customer satisfaction. It is a challenging task to design an effective fleet management strategy that can adapt to an environment involving complex dynamics between demand and supply. Existing studies usu-

ally work on a simplified problem setting that can hardly capture the complicated stochastic demand-supply variations in high-dimensional space. We propose to tackle the large-scale fleet management problem using reinforcement learning, and propose a contextual multi-agent reinforcement learning framework including two concrete algorithms, namely contextual deep Q-learning and contextual multi-agent actor-critic, to achieve explicit coordination among a large number of agents adaptive to different contexts. We show significant improvements of the proposed framework over state-of-the-art approaches through extensive empirical studies.

1.1.4 The Provable Advantage of Collaborative Learning

Previously, we propose the heuristic collaborative approach to coordinate a large number of learning agents to resolve a real-world application. In addition, we would like to provide a rigorous answer to whether there is a provable benefit from increasing the number of collaborative learning agents. We investigate this problem in federated learning, which is a critical scenario in both industry and academia. Federated learning (FL) learns a model jointly from a set of participating devices without sharing each other's privately held data. The characteristics of non-*iid* data across the network, low device participation, and the mandate that data remain private bring challenges in understanding the convergence of FL algorithms, particularly in regards to how convergence scales with the number of participating devices. Here, we focus on Federated Averaging (FedAvg)–the most widely used and effective FL algorithm in use today–and provide a comprehensive study of its convergence rate. Although FedAvg has recently been studied by an emerging line of literature, it remains open as to how FedAvg's convergence scales with the number of participating devices in the FL setting–a crucial question whose answer would shed light on the performance of FedAvg in large FL systems. We fill this gap by establishing convergence guarantees for FedAvg under

three classes of problems: strongly convex smooth, convex smooth, and overparameterized strongly convex smooth problems. We show that FedAvg enjoys linear speedup in each case, although with different convergence rates. For each class, we also characterize the corresponding convergence rates for the Nesterov accelerated FedAvg algorithm in the FL setting: to the best of our knowledge, these are the first linear speedup guarantees for FedAvg when Nesterov acceleration is used. To accelerate FedAvg, we also design a new momentum-based FL algorithm that further improves the convergence rate in overparameterized linear regression problems. Empirical studies of the algorithms in various settings have supported our theoretical results.

1.2 Dissertation Structure

The remainder of this dissertation is organized as follows. We introduce the background of collaborative learning in Chapter 2. In Chapter 3, we start with learning linear models for multiple tasks while incorporating flexible forms of interactions and develop an interactive approach to solicit human expert knowledge for model collaborations. This chapter was previously published as "Multi-task Feature Interaction Learning" [115] and "Interactive Multi-task Relationship Learning" [117]. In Chapter 4, we present data-driven collaboration methods to interact among heterogeneous learning agents, which can largely improve the sample-efficiency of reinforcement learning algorithms. The materials in this chapter are based on "Collaborative Deep Reinforcement Learning" [114] and "Ranking Policy Gradient" [118]. In Chapter 5, we study a real-world application and design a coordination strategy that can scale to a large number of learning agents. The materials in this chapter were published as "Efficient large-scale fleet management via multi-agent deep reinforcement learning" [116]. In

Chapter 6, we present rigorous theories on the improvement of convergence rates with respect to the increasing number of collaborative learning agents, which advocate the advantage of collaborative learning. The materials in this chapter are based on "Federated Learning's Blessing: FedAvg has Linear Speedup" [157]. We conclude this dissertation in Chapter 7.

Chapter 2

Background

In this chapter, we first give a coherent definition of *collaborative learning* used in throughout this dissertation, then we discuss connections and discrepancies among four specific scenarios under this overarching framework.

2.1 Collaborative Learning Problem Formulation

In disciplines of cognitive science, education and psychology, *collaborative learning*, a situation in which a group of people learn to achieve a set of tasks together, has been advocated throughout previous studies [50]. Motivated by the phenomenal success of human collaborative learning, we study the collaborative learning in the domain of artificial intelligence. We first provide a general definition of collaborative learning in this thesis.

Definition 1 (Collaborative learning)**.** *Collaborative learning is a general learning paradigm that multiple learning agents collaborate to solve one or a set of tasks.*

Here, we would like to clarify the several terminologies used in Definition 1.

- **multiple**: in contrast to individual learning, collaborative learning here covers a wide range of learning: from a small scale such as a pair of learning agents to large-scale such as thousands of learning agents.

- **learning agents**: The learning agent refers to a machine learning model that behaves differently from each other. For example, learning agents can be parameterized by different deep neural networks. The neural network can have different domains or architectures. The central requirement is that each learning agent can learn individually and conduct decision making independently.

- **collaborate**: the interaction among different learning agents. The strategy of this interaction is the central design of the collaborative learning algorithm.

- **solve one or a set of tasks**: In machine learning, solving one or a set of tasks refers to optimizing one or several objective functions that generalize well to the unseen scenarios.

More concretely, we provide the problem formulation of collaborative learning as follows:

$$\min_{\mathbf{W}=\{\mathbf{w}_i\}_{i=1}^K} \sum_{i=1}^K F_i(\mathbf{W}) \quad \text{s.t. } \mathbf{w}_i \in \mathcal{C}_i(\mathbf{W}) \quad \forall i = 1, \ldots, K \tag{2.1}$$

where $F_i, i = 1, \ldots, K$ refers to the set of tasks we want to solve. The model parameter $\mathbf{w}_i$ denotes the learning agents. It is worth noting that $\mathbf{w}_i$ is not necessarily represented by a single instance, e.g., a neural network, a decision tree, etc. We use $\mathbf{w}_i$ to denote all variables that need to be determined for a decision process, which constructs a mapping from the input of task i to the action, such as regression, classification, etc. We use the set $\mathcal{C}_i, \forall i = 1, \ldots, K$ denotes the interactions between learning agent i and others, which can encode various types of collaboration strategies into the learning process as we will discuss shortly. For simplicity, we denote the union of models of all learning agents as $\mathbf{W} = \{\mathbf{w}_i\}_{i=1}^K$. The rationale of collaborative learning is that the proper design of interactions $\mathcal{C}$ among the learning agents

facilitates the optimization of objectives.

It is worth noting that the collaboration set is a more general expression comparing to the regularization. The regularization has a specific form on enforcing the formulation while the set of collaboration can integrate more flexible algorithmic designs of interaction. In this thesis, despite differentiations exist in terms of how different learning agents interact, we follow the common practice and use cooperation and collaboration interchangeably [50].

2.2 A Taxonomy of Collaboration

In this section, we present different categories of collaborations, which leads to several subfields in the machine learning community. We discuss the connections and discrepancies of those related subfields and explore the possible advantages of organizing them in a unified view.

2.2.1 Model-Driven Collaboration

The first category of collaborative learning is model-driven collaboration, which directly enforces the interaction of learning agents in the parameter space. From the perspective of transfer learning, these approaches implement knowledge transfer from introducing inductive bias during the learning. It specifically specify the conditions of learned solution needs to be satisfied, such as sparsity or low-rank property. In this case, the collaboration constrain reduces to the various regularizations and the collaborative learning reduces to multi-task learning and federated learning. More concretely, we set $\mathcal{C}_i = \mathcal{R}(\mathbf{W})$, where $\mathcal{R}(\cdot)$ is the regularization added to the $\mathbf{W}$. For example, under the situations that $\mathbf{W}$ is a matrix (each learning agents' model is a vector), a common regularization is trace norm $\mathcal{R}(\mathbf{W}) = \{\mathbf{W}|\|\mathbf{W}\|_{tr}$ that controls the subspace of multiple models.

Multi-Task Learning (MTL) is a principled learning paradigm that leverages useful information contained in multiple related tasks to help improve the generalization performance of all the tasks [226]. The goal of MTL is to learn K functions for the tasks such that $f_k(\mathbf{x}_{ik}) = y_{ik}$, based on the assumption that all task functions are related to some extent, where each function f_k is parameterized by $\mathbf{w}_k$. The general multi-task learning formulation is given by:

$$\min_{\mathbf{W}} \sum_{k=1}^{K} F_k(\mathbf{w}_k) + \lambda \mathcal{R}(\mathbf{W}) \tag{2.2}$$

Another field that falls into model-driven collaboration is federated learning. Federated learning (FL) learns a single model jointly from a set of learning agents. In general, each learning agent corresponds to a local device and the training is performed sharing each other's privately held data. As for now, the prevalent collaboration strategy is the aggregation of all learning agents' models. The challenge of federated learning is the practical constraints on collaboration: to reduce the communication cost (the frequency of collaboration), deal with system heterogeneity, and understand the theoretical properties of this simple collaborative strategy. We will provide rigorous answers to those questions in Chapter 6.

2.2.2 Data-driven Collaboration

One limitation for the traditional model-based collaboration is that the model structure restricted due to the usage of inductive transfer. To overcome this issue, the data-driven collaboration leverages the techniques such as knowledge distillation, mimic learning.

In this case, the data-driven collaboration constrain is given by

$$\mathcal{C}_i(\mathbf{w}_i) = \{\arg\min_{\mathbf{w}_i} \ell(\mathbf{w}_i, f_{\mathbf{w}_j}(\mathbf{x}), y), \forall(\mathbf{x}, y) \in B\},$$

where B denotes the replay buffer that contain a set of selected data according to the task-specific criteria. Notice that the interaction between learning agents now are conducted through the data collected in B. Since the other learning agent's model labeled the data in B, it contains information learned in agent j, which is then distilled to agent i through loss function $\ell(\cdot)$. In this way, we can empower a flexible network structure among different agents, thus achieve collaboration among heterogeneous learning agents. These approaches will be introduced in Chapter 4.

2.2.3 Collaborative Multi-agent Learning

In collaborative multi-agent learning, the multiple learning agents interact with others to achieve a common task. Each learning agent can perform the learning process individually while the We emphasize this problem as a distinct type of collaboration since the agents can adapt their collaborations through the environment feedback, though this trial and error can be computationally intractable. To improve the sample-efficiency in this scenario, we can enforce a task-specific model-driven or data-driven approach during the learning. We provide a concrete real-world application to demonstrate this category in Chapter 5.

Chapter 3

Model-Driven Collaborative Learning

In this chapter, we discuss model-driven collaboration in the context of multi-task learning. More specifically, we first proposed a novel Multi-Task feature Interaction Learning (MTIL) framework to exploit the task relatedness from high-order feature interactions, which provides better generalization performance by inductive transfer among tasks via shared representations of feature interactions. We formulate two concrete approaches under this framework and provide efficient algorithms: the shared interaction approach and the embedded interaction approach. The former assumes tasks share the same set of interactions, and the latter assumes feature interactions from multiple tasks come from a shared subspace. We have provided efficient algorithms for solving the two approaches. Secondly, the classical multi-task relationship learning could learn an inaccurate task relationship when there are insufficient training data or the data is too noisy, and would mislead the learning towards suboptimal models. In this chapter, we proposed a novel interactive multi-task relationship learning (iMTRL) framework that efficiently solicits partial order knowledge of task relationship from human experts, effectively incorporates the knowledge in a proposed knowledge-aware MTRL formulation. We proposed efficient optimization algorithm for kMTRL and comprehensively study query strategies that identify the critical pairs that are most influential to the learning.

3.1 Multi-Task Feature Interaction Learning

3.1.1 Introduction

Linear models are simple yet powerful machine learning and data mining models that are widely used in many applications. Due to the additive nature of the linear models, it can fully unleash the power of feature engineering, allowing crafted features to be easily integrated into the learning system. This is a desired property in many practical applications, in which high-quality features are the key to predictive performance. Moreover, efficient parallel algorithms are readily available to learn linear models from large-scale datasets. Despite its attractive properties, one apparent limitation of such models is that they can only learn a set of individual effects of features contributing to the response, due to its linear additive property. Thus when a part of the response is derived from interactions between features, such models would not be able to detect such non-linear predictive information, thereby leading to poor predictive performance.

In practice, high-order feature interactions are common in many domains. For example, in genetics studies, environmental effects and genetic-environmental interaction are found to have strong relationship with the variability in adoptee aggressivity, conduct disorder and adult antisocial behavior [29]. Similarly, the interaction effects between continuance commitment and affective commitment was found in predicting annexed absences [177]. Also, a recent study of depression found that genotype, sex, environmental risk and their interaction have combined influence on depression symptoms [52]. It is also reported that the interaction of brain-derived neurotrophic factor and early life stress exposure are identified in predicting syndromal depression and anxiety, and associated alterations in cognition [63]. In biomedical studies, many human diseases are a result of complicated interactions among

genetic variants and environmental factors [79]. One intuitive solution to overcome this limitation is to augment interaction terms into linear models, explicitly modeling the effects from the interactions. However, this will dramatically increase the model complexity and lead to poor generalization performance when there is limited amount of data [35, 39, 124, 158, 216].

On the other hand, when there are multiple related learning tasks, the multi-task learning (MTL) paradigm [10, 19, 33] has offered a principled way to improve the generalization performance of such learning tasks by leveraging the relatedness among tasks and performing inductive transfer among them. The past decade has witnessed a great amount of success in applying MTL to tackle problems where large amount of labeled data are not available or creating such datasets incurs prohibitive cost. Such problems are especially prevalent in biological and medical domains, where MTL has achieved significant success, including data analysis on genotype and gene expression [101], breast cancer diagnosis [228] and progression modeling of Alzheimer's Disease [68], etc. The MTL improves generalization performance by learning a shared representation from all tasks, which serves as the agent for knowledge transfer. Structured regularization has provided an effective means of modeling such shared representation and encoding various types of domain knowledge on tasks [10, 89, 142, 199]. The attractive benefits provided by MTL make it an ideal scheme when learning problems involve multiple related tasks with feature interactions, because tasks may be related with each other by shared structures on feature interactions. For example, predicting various cognitive functions may involve a shared set of interactions among brain regions.

However, many existing MTL frameworks are based on linear models [10] in the original input space. Thus they cannot be directly applied to explore task relatedness in the form of high-order feature interactions. On the other hand, although traditional nonlinear MTL methods based on neural networks (e.g., [13]) can exploit non-linear feature interactions

to some extends, it is generally difficult to encode prior knowledge on task relatedness to such models. In this chapter, we propose a novel multi-task feature interaction learning framework, which learns a set of related tasks by exploiting task relatedness in the form of shared representations in both the original input space and the interaction space among features. We study two concrete approaches under this framework, according to different prior knowledge about the relatedness via feature interactions. The *shared interaction approach* assumes that there are only a small number of interactions that are relevant to the predictions, and all tasks share the same set of interactions; the *embedded interaction approach* assumes that, for each task, the feature interactions are derived from a low-dimensional subspace that is shared across different tasks. We have provided formulations and efficient algorithms for both approaches. We conduct empirical studies on both synthetic and real datasets to demonstrate the effectiveness of the proposed framework on leveraging feature interactions from tasks. The contributions of this paper are three folds:

- Our novel framework has extended the MTL paradigm, for the first time, to allow high-order representations to be shared among tasks, by exploiting predictive information from feature interactions.
- We proposed two novel approaches under our framework to model different task relatedness over feature interactions.
- Our comprehensive empirical studies on both synthetic and real data have led to practical insights of the proposed framework.

The remainder of this paper is organized as follows: Section 3.1.2 reviews related work of MTL and models involving feature interactions. Section 3.1.3 introduces the framework for

MTIL. The two approaches under MTIL have been given in 3.1.4. Section 6.6 presents the experimental results on both synthetic and real datasets.

3.1.2 Related Work

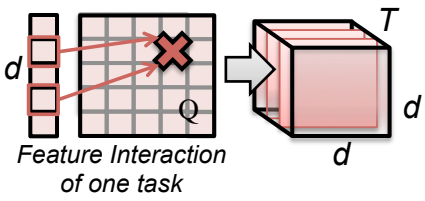
The proposed research is related to existing work on MTL and feature interaction learning. In this section, we briefly summarize the these related work and show how our work advances these areas.

Multi-Task Learning. MTL has been extensive studied over the last two decades. In the center of most MTL algorithms is how task relationships are assumed and encoded into the learning formulations. The concept of learning multiple related tasks in parallel was first introduced in [33]. It was demonstrated in multiple real-world applications that adding a shared representation in neural network tasks can help others get better models. Such discovery had inspired many subsequent research efforts in the community and applications in diverse application domains. Among these studies, the regularized MTL framework has been pioneered by [55]. The regularization scheme can easily integrate various task relationship into existing learning formulations to couple MTL, thus providing a flexible multi-task extension to existing algorithms. It is well adopted and is soon generalized to a rich family of MTL algorithms.

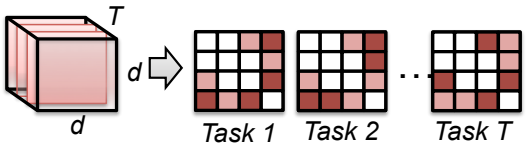
MTL via Regularization. Among the work in the regularization based MTL scheme, there are many different assumptions about how tasks are related, leading to different regularization terms in the formulation. For example, one common assumption is that the tasks share a subset of features, and the task relatedness can be captured by imposing a group sparsity penalty on the models to achieve simultaneous feature selection across tasks [199, 142].

Another common assumption is that the models of tasks come from the same subspace, leading to a low-rank structure within the model matrix. Directly penalizing the rank function leads to NP-hard problems, and one convex alternative is to penalize the convex envelop of the rank function, i.e., trace norm. This encourages low-rank by introducing sparsity to the singular values of the model matrix [89]. In [10], the authors studied a MTL formulation that learns a common feature mapping for the tasks and assumed all tasks share the same features after the mapping. The authors have shown that this assumption can also be equivalently expressed by a low- rank regularization on the model. There are many more formulations that fall into this category of formulation to capture task relatedness by designing different shared representation and regularization terms, such as cluster structures [232], tree/graph structures [101, 38], etc. However, to the best of our knowledge, all of these formulations do not consider feature interactions in the model, and extensions to consider interactions are not straightforward. In this work, we will extend the MTL framework to enable knowledge transfer not only in the original input space, but also in higher- order feature interaction space.

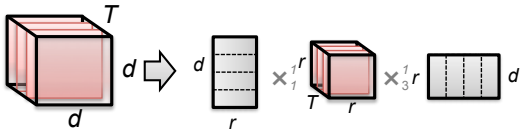
Multilinear MTL. The use of tensor in MTL has shown to be very effective in representing structural information underlying in MTL problems. In [162], Romera-Paredes *et al.* proposed a multilinear multitask (MLMTL) framework that arranges parameters of linear effects from all tasks into a tensor $\mathcal{W}$, by which they are able to represent the multi-modal relationships among tasks. In a dataset containing multi-modal relationships, tasks can be referenced by multiple indices. In MLMTL, the authors employed a regularizer on $\mathcal{W}$ to induce a low-rank structure to transfer knowledge among tasks. The optimization problem contains the minimization of tensor's rank, which leads to solving a non-convex problem. Thus the authors develop an alternating algorithm, employing the Tucker decomposition and convex



a) tensor representation of feature interactions



b) structured sparsity of an interaction tensor



c) low-rank structure of an interaction tensor

Figure 3.1: Illustration of MTL with feature interactions. (a) the feature interactions from multiple tasks can be collectively represented as a tensor $\mathcal{Q}$; group sparse structures (c) and low-rank structures (b) in feature interactions can be used to facilitate multi-task models.

relaxation using tensor trace norm. Although the authors also used a tensor representation in MTL, the learning formulations, implications, as well as the meaning of such the tensor is fundamentally different from those in our work. The proposed MTIL framework utilizes tensor to capture the relatedness among tasks and transfer knowledge through high-order feature interactions, which cannot be achieved by any existing MTL formulations. Note that the tensor in MLMTL is indexed by multi-modal tasks. In MTIL, the tensor is indexed by features and tasks, which is clearly different from the aforementioned work. In the proposed embedded interaction approach for MTIL, however, we face a similar challenge in MLMTL to seek a solution involving a low-rank tensor.

Feature Interaction

In many machine learning tasks, we are interested in learning a linear predictive model. Given the input feature vector of a sample, the response is given by a linear combination of

these features, i.e., a weighted sum of the features. Because of this reason we call them linear effects. There are strong evidences found in many complex applications that, in addition to the linear effects, there are also effects from high-order interactions between such features. As a result, there are considerable efforts from both academia and industry aiming at addressing this limitation by removing the additive assumption and including interaction effects.

To overcome the dimensionality issues introduced by interaction effects, two types of heredity constraints have been studied [20]; namely strong hierarchy in which an interaction effect can be selected into the model only if both of its corresponding linear effects have been selected, and weak hierarchy, in which an interaction effect can be selected if at least one of its corresponding linear effects has been selected. In [39], the authors proposed an approach known as SHIM to identify the important interaction effects. SHIM extends the classical Lasso [194] and enforces a strong hierarchy. An iterative algorithm was proposed based on Lasso, which may not scale to problems with high dimensional feature space. Radchenko *et. al* proposed the VANISH method to address the problem [158]. They developed a convex formulation with a refined penalty that can not only learn the sparse solution, but also treat the linear and interaction effects using different weights. This way, the main effect could have more influence on the prediction. In [20], a hierarchical lasso was proposed to search for interactions with large main effects instead of considering all possible interactions. The authors proposed an algorithm based on ADMM for strong hierarchy lasso and a generalized gradient descent for weak hierarchical lasso. More recently, Liu *et al.* [124] proposed an efficient algorithm for solving the non-convex weak hierarchical Lasso directly, based on the framework of general iterative shrinkage and thresholding (GIST) [67]. The authors proposed a closed form solution of proximal operator and further improved the efficiency of solving the subproblem of proximal operator from quadratic to linearithmic time complexity.

In many real work applications there are multiple related tasks. When those these tasks involve interaction effects, the tasks could be related via the high order feature interactions. In our paper, we propose to address the model complexity issue from interaction effects using a new perspective, by leveraging such relatedness.

3.1.3 Task relatedness in high order feature interactions

In this section, we present the framework of Multi-Task feature Interaction Learning (MTIL). For completeness, we give a self-contained introduction of our work. We will derive concrete learning algorithms under this framework in Section 3.1.4.

Linear and Interaction Effects. Consider the traditional linear models. For an input feature vector $\mathbf{x} \in \mathbb{R}^d$ and a scalar response y, we have assumed the following underlying linear generative model:

$$y = \sum_{i=1}^{d} x_i w_i + \epsilon,$$

where $\mathbf{w} \in \mathbb{R}^d$ is the weight vector for linear effects, and $\epsilon \sim \mathcal{N}(0, \sigma^2)$ is a Gaussian noise. A linear model $f(\mathbf{x}; \mathbf{w}) = \mathbf{x}^T\mathbf{w}$ can be a quite effective prediction function. However, if the underlying generative model includes effects from feature interactions, i.e.,

$$y = \sum_{i=1}^{d} x_i w_i + \sum_{i=1}^{d}\sum_{j=1}^{d} x_i x_j Q_{i,j} + \epsilon,$$

where $x_i x_j Q_{i,j}$ is the joint effect between the ith feature and the jth feature, and $Q_{i,j}$ is the weight for this joint effect. This type of feature interactions have been commonly found in many applications. If the training data follow this distribution then the linear model is not enough to capture the relationship between input features and output responses. One of the

approaches is to introduce non-linear feature interaction terms into the linear model. That is, we can denote it as a quadratic function:

$$f(\mathbf{x}; \mathbf{w}, \mathbf{Q}) = \mathbf{x}^T \mathbf{w} + \mathbf{x}^T \mathbf{Q} \mathbf{x}, \tag{3.1}$$

where $\mathbf{w} \in \mathbb{R}^d$ and $\mathbf{Q} \in \mathbb{R}^{d \times d}$ collectively represent the parameters for linear effects and interaction effects, respectively. We note that $\mathbf{Q}$ is typically symmetric because this representation includes two terms involving feature i and j: $x_i x_j (Q_{i,j} + Q_{j,i})$ and it also includes second-order feature transformations of the original features $x_i^2 Q_{i,i}$.

Discussions on Feature Interactions. In supervised learning, we seek a predictive function that maps an input vector $\mathbf{x} \in \mathbb{R}^d$ to a corresponding output $y \in \mathbb{R}$. Let $(\mathbf{X}, \mathbf{y}) = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), ...(\mathbf{x}_n, y_n)\}$ be a training dataset, in which each data point is drawn from certain *i.i.d.* distribution μ. The goal of learning is to find the best predictor $\hat{f} \in \mathcal{H}$ so that the predicted value $\hat{\mathbf{y}_i}$ for the input data $\mathbf{x}_i$ is as close as possible to the ground truth $\mathbf{y}_i$, $\forall (\mathbf{x}_i, y_i) \in (\mathbf{X}, \mathbf{y})$, given a loss function $L(.,.)$. We hope that the predictor f learned in this way is close to the optimal model that minimizes the expected loss according to the μ:

$$R(f) = \mathbb{E}_{(\mathbf{X},\mathbf{y}) \sim \mu} L(f(\mathbf{X}), \mathbf{y}). \tag{3.2}$$

Such predictor is given by the minimum of the empirical risk:

$$\hat{f} = \arg\min_{f \in \mathcal{H}} \sum_{i=1}^{n} L(f(\mathbf{x}_i), \mathbf{y}_i).$$

The error caused by learning the best predictor in the training dataset is called the estimation error. The error caused by using a restricted $\mathcal{H}$ is called the approximation error. For a

fixed data size, the smaller the hypothesis space $\mathcal{H}$, the larger the approximation error, and vice versa. The trade-off between approximation error and estimation error is controlled by selecting the size of $\mathcal{H}$. By including feature interactions we would enlarge the hypothesis space, and we may be able to dramatically minimize the approximation error compared to the traditional hypothesis space for linear models. On the other hand, we note that given a limited amount of data, a large hypothesis space may result in models with poor generalization performance. We will need to either increase our training data, or provide effective regularizations to narrow down the hypothesis space.

Multi-task Feature Interactions. We consider the setting that there are multiple learning tasks which are related not only in the original feature space, but also in terms of feature interactions. The propose framework simultaneously learns all related tasks and provides an effective regularization on the hypothesis space using relatedness on the interactions.

Let $\mathcal{D} = (\mathbf{X}_1, \mathbf{y}_1), \ldots, (\mathbf{X}_T, \mathbf{y}_T)$ be the training data for the T learning tasks, and the *i.i.d.* training samples for task t is drawn from $(\mu_t)^{m_t}$, where m_t is the number of data points available for task t. We collectively denote the distribution as $\mathcal{D} \sim \boldsymbol{\mu} = \prod_{t=1}^{T} (\mu_t)^{m_t}$. All tasks have a d-dimensional feature space (i.e., $\mathbf{x}_i \in \mathbb{R}^d$). The corresponding features are homogeneous and have the same semantic meaning. The total training data points are:

$$(\mathbf{X}_t, \mathbf{y}_t) = \{(\mathbf{x}_{1t}, y_{1t}), (\mathbf{x}_{2t}, y_{2t}), \ldots, (\mathbf{x}_{mt}, y_{mt})\}, t = 1, \ldots, T,$$

The goal of MTL is to learn T functions for the tasks such that $f_t(\mathbf{x}_{it}) = y_{it}$, based on the assumption that all task functions are related to some extent.

In order to consider interactions for each task, we use the quadratic predictive function in Eq. 3.1 for all tasks. We collectively represent the linear effects from all tasks as a matrix

$\mathbf{W} = [\mathbf{w}_1, \ldots, \mathbf{w}_T] \in \mathbb{R}^{d \times T}$, $\mathbf{w}_i \in \mathbb{R}^d$ and the interaction effects as a tensor $\mathcal{Q} \in \mathbb{R}^{d \times d \times T}$, in which the t-th frontal slice $\mathbf{Q}_t \in \mathbb{R}^{d \times d}$ represents the interaction effects for task t. We illustrate this interaction tensor in Figure 3.1(a).

Given specific loss functions $\hat{\ell}$ for samples from one task, (e.g., square loss for regression and logistic loss for classification, see Table 3.1), the loss function for each task is $\ell_t(f, \mathbf{w}, \mathbf{Q}; \mathbf{X}, \mathbf{y}) = \sum_{i=1}^{m_t} \hat{\ell}(f(\mathbf{x}_i; \mathbf{w}, \mathbf{Q}), y_i)$. Our multi-task feature interaction loss function is given by:

$$L(\mathbf{W}, \mathcal{Q}; f, \mathbf{X}, \mathbf{Y}) = \sum_{t=1}^{T} \ell_t(f, \mathbf{w}_t, \mathcal{Q}_t; \mathbf{X}_t, \mathbf{Y}_t). \tag{3.3}$$

Note that it is not necessary for all tasks to have the same loss function. In MTL, the learning of each task benefits from the knowledge from other tasks, which effectively reduces the hypothesis space for all tasks. In order to achieve knowledge transfer among tasks, we would like to impose shared representations via designing regularization terms on both $\mathbf{W}$ and $\mathcal{Q}$, which specify how tasks are related in the original feature space and features interactions, respectively.

The MTIL Framework. The proposed Multi-Task feature Interaction Learning (MTIL) framework is then given by the following learning objective:

$$\min_{\mathbf{W}, \mathcal{Q}} L(\mathbf{W}, \mathcal{Q}; f, \mathbf{X}, \mathbf{Y}) + \lambda_R R_F(\mathbf{W}) + \lambda_I R_I(\mathcal{Q}), \tag{3.4}$$

where $R_F(\mathbf{W})$ is the regularization providing task relatedness in the original feature space, $R_I(\mathcal{Q})$ is the regularization encoding our knowledge about how feature interactions are related among tasks, λ_R and λ_I are the corresponding regularization coefficients. For $\lambda_I \to \infty$, the

problem reduces to traditional MTL, when R_I is chosen properly. In this paper, we formulate two concrete approaches to capture the feature interaction patterns:

- **Shared Interaction Approach.** In many applications, even though we have a large number of feature interactions, only a few interactions may be related to the response [20, 39]. When learning with multiple tasks, different tasks may share exactly the same set of feature interactions, but with different effects. As such, we can design MTIL formulations that learns a set of common feature interactions, which could effectively reduce the hypothesis space. During the learning process the selected feature interactions for one task will be task's knowledge, contributing to the share representation: a set of indices of common interactions. An analogy in traditional MTL is the joint feature learning approach [142, 199], in which tasks share the same set of features. One way to achieve this approach is by using the structured sparsity to induce the same sparsity patterns on the interaction effects. An illustration of this approach is given in Figure 3.1(b).

- **Embedded Interaction Approach.** When the response from one task is related to complicated feature interactions, the patterns of such interactions may be captured by a low-dimensional space, resulting in a low-rank interaction matrix. When there are multiple related tasks, they could have a shared low-dimensional space, i.e., different interaction matrices may share the same set of rank-1 basis matrices, but have different weights associated with these basis matrices. When collectively represented by a tensor, we end up with a low-rank tensor. During the learning process, each task contributes their subspace information to facilitate learning of the share low-dimensional subspace, which in turn, improves the feature space. The analogy in traditional MTL is the

Table 3.1: Examples of common smooth loss functions.

Loss with Interaction	Loss function L_i	Gradient | Linear Eff. $\nabla_{\mathbf{W}} L_i$	Gradient | Interaction Eff. $\nabla_{\mathbf{Q}_t} L_i$
Logistic Loss*	$-[log(g(\mathbf{x}_i))y_{ti} + (1-y_{ti})(log(1-g(\mathbf{x}_i)))]$	$(g(\mathbf{x}_i) - y_{ti})\mathbf{x}_i$	$(g(\mathbf{x}_i) - y_{ti})\mathbf{x}_i\mathbf{x}_i^T$
Squared Loss	$\frac{1}{2}\|\|\mathbf{x}_i^T\mathbf{w}_t + \mathbf{x}_i^T\mathbf{Q}_t\mathbf{x}_i - y_{ti}\|\|_2^2$	$\mathbf{x}_i(\mathbf{x}_i^T\mathbf{w}_t + \mathbf{x}_i^T\mathbf{Q}_t\mathbf{x}_i - y_{ti})$	$\mathbf{x}_i(\mathbf{x}_i^T\mathbf{w}_t + \mathbf{x}_i^T\mathbf{Q}_t\mathbf{x}_i - y_{ti})\mathbf{x}_i^T$
Squared Hinge†	$h(y_{ti}(\mathbf{x}_i^T\mathbf{w}_t + \mathbf{x}_i^T\mathbf{Q}_t\mathbf{x}_i))$	$y_{ti}\mathbf{x}_i h'(\mathbf{x}_i^T\mathbf{w}_t + \mathbf{x}_i^T\mathbf{Q}_t\mathbf{x}_i)$	$y_{ti}\mathbf{x}_i\mathbf{x}_i^T h'(\mathbf{x}_i^T\mathbf{w}_t + \mathbf{x}_i^T\mathbf{Q}_t\mathbf{x}_i)$

*$g(\mathbf{x})$ is the sigmoid function defined as $g(\mathbf{x}_i) = 1/\left\{1 + exp(-(\mathbf{x}_i^T\mathbf{w}_t + \mathbf{x}_i^T\mathbf{Q}_t\mathbf{x}_i))\right\}$

†$h'(z) = \{-1 \quad \text{for } z \leq 0, \quad z-1 \text{ for } 0 < z < 1, \quad 0 \text{ for } z \geq 1\}$

low-rank based models [10, 89]. However, there are challenging questions such as: How to define a proper rank function for tensor? Are there tractable algorithms to induce low-rank structure in tensor? In the next section we will discuss these important questions and propose efficient algorithms. We illustrate this approach in Figure 3.1(c).

We note that even though we only provided two specific approaches in this paper, the proposed MTIL framework could offer broader class of formulations. The proposed framework allows many other possible ways to define task relatedness on feature interactions, leading to a brand-new research area of MTL.

3.1.4 Formulations and algorithms of the two MTIL approaches

In this section, we will study how the formulations and algorithms of the shared interaction approach and embedded interaction approach under the proposed MITL framework. We note that extension of multi-task learning to feature interactions is not trivial because of the involvement of tensors. We start with formulating the shared interaction approach by incorporating a group Lasso penalty to introduce structured sparsity on the tensor, which would select only a set of common feature interactions across different tasks that are relevant to the prediction. For the embedded interaction approach, we propose both a convex formulation and a non-convex formulation. While the convex formulation leads to efficient optimization algorithms and global solutions, the non-convex formulation provides reduced storage complexity for large-scale problems.

3.1.4.1 Preliminary

Here, we use the following basic definition of tensor:

Mode-n fiber is a vector defined by fixing every index but one. We may see it as the higher order analogue of matrix rows (mode-2 fibers) and columns (mode-1 fibers). For example, in a three-way tensor $\mathcal{Q} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, the mode-3 fiber is $\mathcal{Q}_{i,j,:} \in \mathbb{R}^{n_3}$.

Mode-n unfolding is the process of reordering the elements of an N-way tensor $\mathcal{Q} \in \mathbb{R}^{n_1 \times n_2 \times,..,\times n_N}$ into a matrix. The mode-k unfolding of tensor $\mathcal{Q}$ is denoted by $\mathcal{Q}_{(k)} \in \mathbb{R}^{n_k \times J_k}$, where $J_k = \prod_{i=1, i \neq k}^{N}$. The matrix is arranged by concatenating all mode-k fibers of the tensor.

Rank-n denotes the rank of tensor's mode-n unfolding. It's actually the dimension of the space spanned by the mode-n fibers of tensor. Specifically, $\text{rank}_n(\mathcal{Q}) = \text{rank}(\mathcal{Q}_{(n)})$. When $\mathcal{Q}$ is a matrix (i.e. 2-way tensor), this becomes the regular definition of rank, since $\text{rank}_1(\mathcal{Q}) = \text{rank}_2(\mathcal{Q}) = \text{rank}(\mathcal{Q})$.

3.1.4.2 Shared Interaction Approach

The goal of the shared interaction approach is to identify a set of common and relevant feature interactions across different tasks. The interaction tensor $\mathcal{Q}$ in our framework has provided a convenient representation to encode such information, and we are able to incorporating a group Lasso penalty [61] to induce a special type of structured sparsity on the tensor, coupling the same interactions for all tasks. Recall that the sparsity implies that only the significant interaction effects are captured in the model. For the purpose of shared interaction,

a *sparse tensor norm* is defined as:

$$||\mathcal{Q}||_{\text{GL-Sym}} \equiv \sum_{i=1}^{d} \sum_{j \geq i}^{d} \sqrt{\sum_{k=1}^{K} \left(Q_{i,j,k}^2 + Q_{j,i,k}^2 \right)}. \tag{3.5}$$

Note that this norm enforces a symmetric sparsity by over the tensor, so that the one group is defined to include coefficients of one interaction between feature i and feature j, from all tasks. Penalizing the tensor sparse norm leads to the following formulation:

$$\min_{\mathbf{W}, \mathcal{Q}} L(\mathbf{W}, \mathcal{Q}; f, \mathbf{X}, \mathbf{Y}) + \lambda_F R_F(\mathbf{W}) + \lambda_I ||\mathcal{Q}||_{\text{GL-Sym}}, \tag{3.6}$$

where the parameter λ_I control the sparsity of tensor $\mathcal{Q}$, a larger λ_I will end up with a more sparse $\mathcal{Q}$. The solution to formulation delivers a tensor such that the mode-3 fibers are either all zeros vectors or non zero vectors, i.e., interaction effects between 2 features x_i, x_j either exists on all tasks, or irrelevant for all tasks. Note that even the sparsity patterns is same for all tasks, their interactions may have different weights. It is easy to see that, this approach subsumes the traditional multi- task learning as a special case: when $\lambda_I \to \infty$ by setting regularization parameter on tensor $\mathcal{Q}$ to infinity, all the elements in of $\mathcal{Q}$ in the solution will be zeros, and the model only considers linear effects.

When the loss function L chosen is convex and continuously differentiable with Lipschitz continuous gradient [158], then we can use proximal based gradient methods, such as first order FISTA [16], SpaRSA [214] or second order Proximal Newton [108] to solve it efficiently. Because that the linear effects and interaction effects are decoupled in the predictive function, a major class of loss functions belong to this category, and we give a few examples of common loss functions in Table 3.1. Note that even when L is non-convex, a local optimal solution

can be efficiently obtained using the GIST framework [67]. The key to apply these algorithms is to efficiently compute the proximal operator that associates to the problem (refer to [150] for more details about proximal):

$$\min_{\mathbf{W},\mathcal{Q}} \frac{1}{2}(\|\mathbf{W}-\hat{\mathbf{W}}\|_F^2 + \|\mathcal{Q}-\hat{\mathcal{Q}}\|_F^2) + \rho_1 R_F(\mathbf{W}) + \rho_2 ||\mathcal{Q}||_{\text{GL-Sym}},$$

where $\hat{\mathbf{W}}$ and $\hat{\mathcal{Q}}$ are intermediate solutions at each step, ρ_1 and ρ_2 are regularization parameters augmented with step size. Note that we have extend the Forbenius norm from matrix to tensor. We see that the problem is decoupled for $\mathbf{W}$ and $\mathcal{Q}$. And the tensor proximal:

$$\min_{\mathcal{Q}} \frac{1}{2}\|\mathcal{Q}-\hat{\mathcal{Q}}\|_F^2 + \rho_2 ||\mathcal{Q}||_{\text{GL-Sym}},$$

can be solved in the same way as the group Lasso proximal operator [222]. Moreover, we find that when the gradient is symmetric, we don't need to enforce a symmetric tensor sparse norm, and we could simply use a simple alternative:

$$||\mathcal{Q}||_{GL} = \sum\nolimits_{i,j} \sqrt{\sum\nolimits_{k=1}^{K} Q_{i,j,k}^2},$$

and initialize the algorithm with a symmetric tensor as the starting point. The reason that symmetry holds can be explained by two parts. First, the gradient of $\mathcal{Q}$ is symmetric, therefore the gradient descent step won't change the symmetry of tensor $\mathcal{Q}$. Second, the proximal operator associated to sparse tensor norm won't change the symmetry of matrix. To see this, the proximal operation is performed by vectorizing the matrix into a vector and shrink each element of the vector with respect to a input vector, which is obtained by

the last gradient descent step. Since the input vector represents an symmetric matrix, the element and its symmetric element will always shrink to the same new value. Therefore, the symmetry of $\mathcal{Q}$ holds. The sparse tensor norm is equivalent to perform the l_1 projection of vectors where each element is the l_2 norm of mode-3 fiber in tensor $\mathcal{Q}$.

3.1.4.3 Embedded Interaction Approach

The share interaction approach has enforced a very restrictive form of how tasks are supposed to relate to each other. In many applications, the prediction may be a result of complicated feature interactions, instead only involves a few interactions. Even though the prediction may involve all feature interactions, it is usually a reasonable assumption that there are patterns among these interactions. Numerically, existence of patterns imply a low-dimensional subspace, which is reflected by a low-rank structure in the matrix. When there are multiple related learning tasks, one way for these tasks relate to others via a shared low-dimensional subspace, which gives us a low-rank tensor. As such, we may design a structured regularization to encourage the matrix $\mathcal{Q}$ to be a low-rank tensor. In this paper we describe one convex formulation that encourages low-rank structure by penalizing a tensor norm and one non-convex formulation that directly learns a low-rank representation.

Convex Formulation

One way to obtain a low-rank tensor is to augment our formulation with a rank penalty. One problem associates to tensor is that there is no consistent way to define the rank of a tensor. One way is to use the average rank of unfolding on different mode [62]:

$$\frac{1}{N}\sum_{n=1}^{N}\text{rank}_n(\mathcal{Q}) = \frac{1}{N}\sum_{n=1}^{N}\text{rank}(\mathcal{Q}_{(n)}),$$

where N is the total number of mode of the tensor ($N = 3$ when only pair-wise interactions), and $\mathcal{Q}_{(n)}$ is unfold on n mode. Since minimizing the rank function is proven to be NP-hard, we could penalize the trace norm instead, which is the convex envelope of the rank function. The trace norm is defined as the sum of singular values of the matrix variable [89]. We then obtain the following convex formulation:

$$\min_{\mathbf{W},\mathbf{Q}} L(\mathbf{W}, \mathcal{Q}; f, \mathbf{X}, \mathbf{Y}) + \lambda_R R_1(\mathbf{W}) + \frac{\lambda_I}{N} \sum_{n=1}^{3} ||\mathcal{Q}_{(n)}||_*, \tag{3.7}$$

where $\|.\|_*$ denotes the trace norm. However, this convex formulation penalizes every mode of tensor $\mathcal{Q}$ to be jointly low rank, which may be too restricted in practice, which may lead to suboptimal performance. Moreover, the practical way to solve the formulation in Eq. (3.7) is to use the alternating direction methods of multipliers (ADMM) [23], which introduces auxiliary variables and equality constraints, in order to decouple the three tensor trace norm terms. However, ADMM algorithm in practice is shown to have a slow convergence rate, and less preferred when composite proximal methods such as FISTA can be applied.

One alternative way to address these issues is to use the latent trace norm [195, 196], which is defined as following for a $N-$way tensor:

$$||\mathbf{Q}||_{\text{latent}} = \inf_{\mathcal{Q}^{(1)}+\mathcal{Q}^{(2)}+\ldots+\mathcal{Q}^{(N)}=\mathcal{Q}} \sum_{n=1}^{N} ||\mathcal{Q}^{(n)}_{(n)}||_*,$$

where $\mathcal{Q}^{(1)} \ldots \mathcal{Q}^{(N)}$ are a set of low-rank auxiliary tensors, which states that the original tensor can be decomposed into the sum of a set of tensors that are low-rank in different modes. Finally, we proposed to drop the equality constraint that each auxiliary tensor equal to the original one, but we directly use the mixture of tensors to represent the original tensor,

so the problem becomes a unconstrained optimization problem. The predictive function of task t with such mixture is given by:

$$f_{\text{mix}}(\mathbf{x}; \mathbf{w}_t, \{\mathcal{Q}^{(i)}\}_{i=1}^3) = \mathbf{x}^T \mathbf{w}_t + \mathbf{x}^T (\sum\nolimits_{i=1}^3 \mathcal{Q}_t^{(i)}) \mathbf{x},$$

where $\mathcal{Q}^{(j)} \in \mathbb{R}^{d \times d \times K}$, $\forall j = 1, 2, 3$ are the auxiliary tensors for replacing the original tensor $\mathcal{Q}$, matrix $\mathcal{Q}_{(j)}^{(j)} \in \mathbb{R}^{(n_1 n_2 n_3 / n_j) \times n_j}$ is the mode j unfolding of tensor $\mathcal{Q}^{(j)}$, $\mathcal{Q}_t^{(j)} \in \mathbb{R}^{d \times d}$ is the tth frontal slice of tensor $\mathcal{Q}^{(j)}$. Finally, our convex formulation under embedded interaction approach is given by:

$$\min_{\mathbf{W}, \{\mathcal{Q}^{(i)}\}_{i=1}^3} L(\mathbf{W}, \{\mathcal{Q}^{(i)}\}_{i=1}^3; f_{\text{mix}}, \mathbf{X}, \mathbf{Y}) \qquad + \lambda_F R_F(\mathbf{W}) + \lambda_I \sum_{j=1}^3 ||\mathcal{Q}_{(j)}^{(j)}||_*.$$

The convexity of this formulation holds since both the loss function and the penalty are convex. We note that this formulation can be solved in the same way as the formulation in Eq. (3.7), and the model is much more flexible to model the complicated interactions among the features, leveraging the advantages of such auxiliary tensors.

Non-Convex Formulation

Although using proximal gradient methods we are able to secure an optimal solution for the convex formulation, the time complexity and storage cost are unacceptable in practice as the dimension of data increase. To see this, we note that the proximal operator associated to a trace norm regularized objective requires singular projections [89], which requires cubic-complexity singular value decomposition. Recall in each iteration of the gradient methods could involve more than one computation of proximal operator [16], and thus the computation may be prohibitive when dimension grows larger. On the other hand, we have to maintain

3 dense tensors of size $d \times d \times T$ which means the storage cost is at $O(d^2)$, where T is the number of tasks and typically we have $T \ll d$. Also the mixture of three low-rank auxiliary tensors may lead to some difficulty when it comes to analyzing the predictive model itself.

To this end, we propose to use a tensor with a explicit low-rank structure. Consider the interaction effects matrix $\mathbf{Q} \in \mathbb{R}^{d \times d}$ for one task, we assume the low-rank decomposition $\mathbf{Q} = \mathbf{B}\tilde{\mathbf{Q}}\mathbf{B}^T$, where $\mathbf{B} \in \mathbb{R}^{d \times r}$ is a basis matrix, $\tilde{\mathbf{Q}} \in \mathbb{R}^{r \times r}$ is a small matrix, capturing the information of the original tensor under the set of bases (columns) in $\mathbf{B}$. To see this, we can expand $\mathbf{Q} = \sum_{i,j=1}^{r} \tilde{\mathbf{Q}}_{(i,j)} \mathbf{B}_i \mathbf{B}_j^T$, meaning the matrix $\mathbf{Q}$ is a result of interactions among bases in $\mathbf{B}$ and also spanned by the columns of $\mathbf{B}$. We thus can use a predictive function that explicitly considers this low-rank structure:

$$f_{\text{nvc}}(\mathbf{x}; \mathbf{w}, \mathbf{B}, \tilde{\mathbf{Q}}) = \mathbf{x}^T\mathbf{w} + \mathbf{x}^T\mathbf{B}\tilde{\mathbf{Q}}\mathbf{B}^T\mathbf{x}.$$

When there are multiple tasks, our assumption for embedded interaction approach is the shared basis, meaning $\mathbf{B}$ is restricted to be same as all other tasks. The multi-task loss function is thus given by:

$$L(\mathbf{W}, \{\mathbf{B}\}, \tilde{\mathcal{Q}}; f_{\text{nvc}}, \mathbf{X}, \mathbf{Y}) = \sum_{t=1}^{T} \ell_t(f_{\text{nvc}}, \mathbf{w}_t, \mathbf{B}, \tilde{\mathcal{Q}}_t; \mathbf{X}_t, \mathbf{Y}_t),$$

where $\tilde{\mathcal{Q}} \in \mathbb{R}^{r \times r \times T}$ collective denotes the set of matrices $\tilde{\mathcal{Q}}$ from all tasks. This loss function is not convex because of the multiplication of variables in $\mathbf{x}^T\mathbf{B}\tilde{\mathbf{Q}}\mathbf{B}^T\mathbf{x}$. This loss function leads to our final non-convex formulation for embedded:

$$\min_{\mathbf{W}, \{\mathbf{B}\}, \tilde{\mathcal{Q}}} L(\mathbf{W}, \{\mathbf{B}\}, \tilde{\mathcal{Q}}; f_{\text{nvc}}, \mathbf{X}, \mathbf{Y})$$

$$+ \lambda_F R_F(\mathbf{W}) + \lambda_I R_I(\{\mathbf{B}\}, \tilde{\mathcal{Q}}),$$

where the regularization $R_I(\{\mathbf{B}\}, \tilde{\mathcal{Q}})$ can be Forbenius norm or other structural information (e.g. ℓ_1 norm). The dimension r of $\mathbf{B}$ can be chosen according to the need of specific application demands, and can be selected by cross-validation. In general, we choose $r \ll d$. We note that the storage complexity for the feature interaction effects (e.g., tensor $\mathcal{Q}$) is reduce from $O(d^2K)$ to $O(dr + r^2K)$, which is dramatically smaller than the full tensor, especially in the high dimensional settings. We could use the family of block coordinate descent algorithms [198] to alternatively solve the variables $\mathbf{W}, \{\mathbf{B}\}$, and $\tilde{\mathcal{Q}}$, to get a local optimal solution.

3.1.5 Experiments

In this section, we perform experiments on both synthetic datasets and two real world datasets to evaluate the effectiveness of our proposed MTIL framework.

3.1.6 Synthetic Dataset

In order to justify the effectiveness of modeling the feature interactions and MTIL framework, we test our methods on synthetic datasets.

3.1.6.1 Effectiveness of modeling feature interactions

In this subsection, we test whether the interactions between features can be properly handled by adding the interaction term $\mathbf{Q}$. To do so, we create a single task synthetic dataset by assuming:

$$\mathbf{y} = \mathbf{X}\mathbf{w} + \text{diag}(\mathbf{X}\mathbf{Q}\mathbf{X}') + \boldsymbol{\epsilon}, \tag{3.8}$$

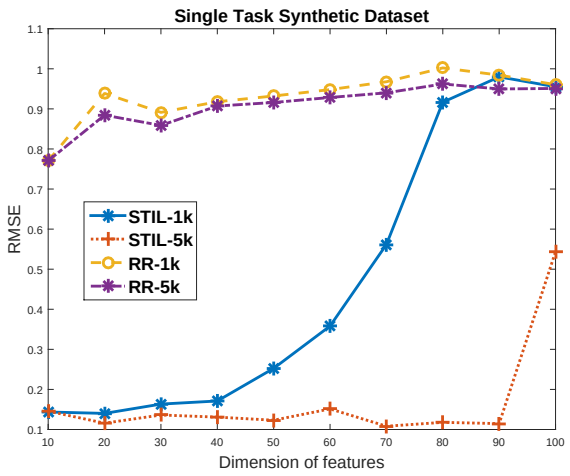


Figure 3.2: RMSE comparison between RR and STIL on two synthetic datasets with sample size of 1k and 5k, respectively.

where $\mathbf{X} \in \mathbb{R}^{n \times d}$ is the feature matrix, $\mathbf{y} \in \mathbb{R}^{n \times 1}$ is the responses, $\mathbf{w} \in \mathbb{R}^{d \times 1}$ is the weight vector, $\mathbf{Q} \in \mathbb{R}^{d \times d}$ is a symmetric, low-rank sparse matrix, which represents the feature interactions in the dataset, and $\boldsymbol{\epsilon} \sim \mathcal{N}(0, 0.01\mathbf{I}_n)$ is the additive noise term. We generate 20 synthetic datasets with different sizes (1000 or 1k and 5000 or 5k) and different feature dimensions (varying from 10 to 100, stepped by 10) by randomly selecting $\mathbf{X}$, $\mathbf{w}$, and $\mathbf{Q}$ and computing $\mathbf{y}$ according to Eq.(3.8).

We use single task feature interaction learning model (STIL) to evaluate the effectiveness of the interaction term $\mathbf{Q}$:

$$\min_{\mathbf{w},\mathbf{Q}} \sum_{i=1}^{n} \frac{1}{2}||\mathbf{x}_i^T\mathbf{w} + \mathbf{x}_i^T\mathbf{Q}\mathbf{x}_i - y_i||_2^2 + \frac{\lambda}{2}||\mathbf{w}||_2^2 + \mu||\mathbf{Q}||_{1,1},$$

where $\mathbf{w} \in \mathbb{R}^{d \times 1}$ is the weight vector, $\mathbf{Q} \in \mathbb{R}^{d \times d}$ is the feature interaction matrix, and $\|\mathbf{Q}\|_{1,1} = \sum_i \sum_j |\mathbf{Q}_{i,j}|$ denotes the $\ell_{1,1}$ norm.

We compared the Root Mean Square Error (RMSE) between the Ridge Regression(RR) and STIL on both of the synthetic datasets. As the results show in Figure 3.2, STIL outperforms RR on both of the datasets, which shows the effectiveness of modeling the

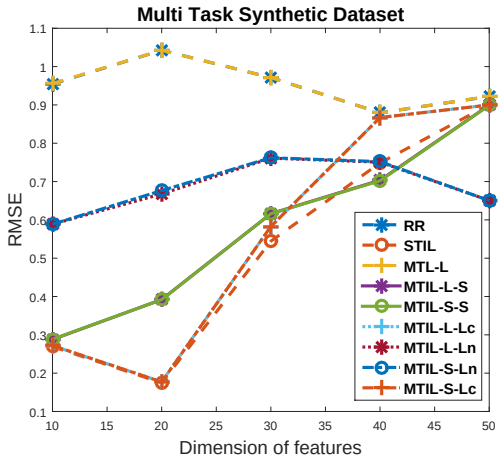


Figure 3.3: Synthetic dataset (Multi-task): Root Mean Square Error (RMSE) comparisons among all the methods. The Y-axis is RMSE, X-axis is dimension of features.

feature interaction in the data. Besides, STIL-5k (RR-5k) performs better than STIL-1k (RR-1k), which demonstrates that the learning models will capture the underlining models of the data better with larger training size. Also note that with the number of dimensions increases, STIL will gradually overfit the data, because of the dramatic increase of the interactions between features.

3.1.6.2 Effectiveness of MTIL

In order to test the effectiveness of MTIL, we generate a multi-task synthetic data by assuming:

$$\mathbf{y}_t = \mathbf{X}_t\mathbf{w}_t + \text{diag}(\mathbf{X}_t\mathbf{Q}_t\mathbf{X}_t^T),\ t = 1, 2, 3, .., T,$$

where $\mathbf{X}_t \in \mathbb{R}^{n\times d}$ is the feature matrix of task t, $\mathbf{y}_t \in \mathbb{R}^{n\times 1}$ is the responses of task t, $\mathbf{W} \in \mathbb{R}^{d\times T} = [\mathbf{w}_1, \mathbf{w}_2, \mathbf{w}_3, ..., \mathbf{w}_T]$ is the weights for tasks. As described in Section 3.1.4.3, we generate feature interaction matrix $\mathbf{Q}_t = \mathbf{B}\mathbf{q}_t\mathbf{B}^T$ and project it into a sparse, symmetric space.

In this experiment, we generate 5 datasets with different feature dimensions from 10 to

50, stepped by 10, by randomly selecting $\mathbf{X}_t$, $\mathbf{w}_t$, $\mathbf{B}$ and $\mathbf{q}_t$.

The predictive performance of the methods outlined below are examined on the synthetic multi-task datasets:

- Ridge Regression (RR): We choose this model as the baseline and make neither assumptions of feature interaction nor the relation among all the tasks.
- STIL: We perform STIL on each of the task independently.
- MTL-L: This approach refers to the traditional MTL method regularized by the trace norm of the weight matrix $\mathbf{W}$[10]. It does not make assumptions on feature interactions.
- MTIL-L-S: This approach, refers to multi-task feature interaction learning regularized by the trace norm of the weight matrix $\mathbf{W}$ and the tensor group lasso norm of tensor $\mathcal{Q}$ (see section 3.1.4.2).
- MTIL-S-S: This approach is similar to MTIL-L-S except that the regularization term on $\mathbf{W}$ is $\ell_{2,1}$ norm.
- MTIL-L-Lc: This approach refers to multi-task feature interaction learning regularized by the trace norm of the weight matrix $\mathbf{W}$ and latent trace norm of tensor $\mathcal{Q}$ (see section 3.1.4.3).
- MTIL-S-Lc: This approach is similar to MTIL-L-Lc except for that the regularization term on $\mathbf{W}$ is $\ell_{2,1}$ norm.
- MTIL-L-Ln: This approach refer to multi-task feature interaction learning regularized by the low rank norm of tensor $\mathcal{Q}$ and the trace norm of the weight matrix $\mathbf{W}$ (see section 3.1.4.3).

- MTIL-S-Ln: This approach is similar to MTIL-L-Ln except for that the regularization term on $\mathbf{W}$ is $\ell_{2,1}$ norm.

Figure 3.3 compares the RMSE of the above methods on the 5 synthetic datasets. We can see that MTIL-L-Ln and MTIL-S-Ln are not that sensitive to the change of feature dimensions, thanks to the low-rank assumption on the feature interaction. Also, RR and MTL-L share a similar performance, which is consistent with the fact that we did not assume any low-rank structure in this synthetic dataset. Note that although STIL performs almost the best on low dimensional data, its performance deteriorates rapidly compared with other MTIL methods, due to the incapability of learning the feature interactions across tasks.

3.1.7 School Dataset

This dataset contains the examination records of 15362 students with 28 features from 139 schools in years of 1985, 1986 and 1987, provided by the Inner London Education Authority(ILEA). In this dataset, each task is to predict exam scores for students in one out of the 139 schools. We perform 4 sets of experiments by varying the amount of training size, from 20% to 50% of the total sample size. We test the approaches summarized in section 3.1.6.2 and tune the parameters on λ_R in set $[10^{-1}, 10^0, ..., 10^9, 10^{10}]$. For MTIL-L-Ln and MTIL-S-Ln methods, the rank of matrix r for each task are tuned in $[2, 3, ..., 19, 20]$. For MTIL-L-S and MTIL-L-Lc, we tune the regularization parameters λ_I in $[10^{-1}, 10^0, ..., 10^9, 10^{10}]$.

The experimental results are shown in Table 3.2. First, for most of the methods, RMSE will decrease when the training size increases. This means that providing more data in the training set will help overcome the overfitting problem. Also, we found that the performance of embedded feature approaches (i.e. MTIL-L-Lc, MTIL-L-Ln, MTIL-S-Ln) are worse than the single task learning approach. The reason behind this is that embedded feature approaches

Table 3.2: Performance comparison MTIL and baselines on the School dataset

	Training 20%	Training 30%	Training 40%	Training 50%
RR	0.9149 ± 0.0031	0.9025 ± 0.0058	0.8885 ± 0.0067	0.8722 ± 0.0059
STIL	0.9149 ± 0.0031	0.9025 ± 0.0057	0.8885 ± 0.0067	0.8721 ± 0.0058
MTL-L	0.8998 ± 0.0044	0.8807 ± 0.0052	0.8657 ± 0.0032	0.8503 ± 0.0070
MTIL-L-S	**0.8623** ± 0.0048	**0.8506**± 0.0038	**0.8511**±0.0043	**0.8404** ± 0.0067
MTIL-S-S	0.8999 ± 0.0063	0.8907 ± 0.0049	0.8832 ± 0.0077	0.8686 ± 0.0046
MTIL-L-Lc	0.9252 ± 0.0090	0.8893 ± 0.0037	0.8859 ± 0.0037	0.8720 ± 0.0044
MTIL-S-Lc	0.9353 ± 0.0133	0.9139 ± 0.0053	0.8941 ± 0.0024	0.8761 ± 0.0062
MTIL-L-Ln	1.0084 ± 0.0180	0.9758 ± 0.0097	0.9328 ± 0.0267	0.9041 ± 0.0140
MTIL-S-Ln	1.0026 ± 0.0368	0.9585 ± 0.0059	0.9297 ± 0.0253	0.8965 ± 0.0066

do not have sparse constraints on the interaction term, which will severely overfit the data when there is not sufficient training samples. Additionally, the MTL-L and MTIL-L-S obtain better performance than single task learning, which indicates that the low-rank structure shared by tasks are effectively captured by the low-rank assumption in these two methods. Moreover, MTIL-L-S method outperforms all other methods, which empirically proves the effectiveness of learning the shared interactions with sparse constraints.

3.1.8 Modeling Alzheimer's Disease

The Alzheimer's Disease Neuroimaging Initiative (ADNI) database(adni.loni.ucla.edu), which was launched in 2003 as a 5-year public-private partnership, is aimed to test whether the positron emission tomography (PET), serial magnetic resonance imaging (MRI), other biological markers, and clinical and neuropsychological assessments can be combined to measure the progression of mild cognitive impairment (MCI) and early Alzheimer's disease (AD). We follow the procedure of preprocessing mentioned in [234] and obtain 648 subjects and 305 MRI features. The parameters are tuned in the same way as we described in 3.1.7.

The RMSE comparison result is shown in Table 3.3. First, we found that all of the MTLs outperform the single task learning approaches (RR and STIL), which demonstrates the

Table 3.3: Performance comparison MTIL and baselines on the ADNI dataset.

	RMSE ± standard deviation
RR	0.9418 ± 0.0023
STIL	0.9417 ± 0.0021
MTL-L	0.9031 ± 0.0007
MTIL-L-S	0.9030 ± 0.0007
MTIL-S-S	0.9162 ± 0.0017
MTIL-L-Lc	0.8941 ± 0.0050
MTIL-S-Lc	**0.8909** ± 0.0059
MTIL-L-Ln	0.8926 ± 0.0009
MTIL-S-Ln	0.9085 ± 0.0028

effectiveness of learning multiple tasks jointly by exploring the relatedness between tasks, as well as the existence of the underlying relatedness between tasks in the ADNI dataset. Second, the RMSE results of MTIL-L-S and MTL-L are comparable with each other, which indicates that the multiple tasks in this dataset do not share the same feature interaction structure. Finally, the result of MTIL-S-Lc method outperforms all other methods, which shows superiority of our feature interaction framework. Through a mixture of 3 low-rank tensor, we are able to learn the feature interaction pattern in this dataset.

3.1.9 Discussion

The proposed multi-task feature interaction learning framework has provided us a way to bridge related tasks using interaction effects. By employing different types of regularizations on the interaction effects tensor, the formulations under this framework have very different characteristics.

For the shared interaction approach: we utilize Group Lasso on the interaction tensor to control the model complexity. The proximal operator admits a closed form solution, and thus the overall computational cost is very low. We are able to obtain interpretable results from the model, showing what are important interactions that are relevant to the prediction tasks.

The main drawback is that we assume all tasks share the same set of interaction effects, which may not be the case for many data sets. One way to further improve the formulation is by extending the strong or weak heredity properties [20, 124] to the proposed MTIL framework.

For the embedded interaction approach: we can easily obtain the global optimal for the convex formulation. Though we are able to tune the regularization parameter on the trace norms to control the rank of the interaction tensor, it is usually very hard to decide the value unless cross-validation is used. A rank larger than necessary may lead to over-fitting when training samples are insufficient. On the other hand, the obtained mixture of 3 tensor is hard to interpret. The non-convex formulation provides a better model decomposition, from which we can see the combination of basis for different tasks and identify embedded bases that are shared among the set of tasks. The drawback of this formulation is that we may easily trapped in a bad local optimal unless we carefully choose the initial value (e.g., using the solution from the convex formulation).

In general, this framework can be generalized into many other possible relatedness on feature interactions by incorporating different regularization terms. Different approaches of this framework should be carefully chosen according to the application domain. In the future work we plan to study the statistical properties of the proposed model, which may lead to deeper understanding of these interaction models.

3.2 Multi-Task Relationship Learning

3.2.1 Introduction

Supervised learning has been a well studied area of machine learning and there are many efficient algorithms to learn from data and generate predictive models to infer labels for

unseen data points. As extensively studied in the statistical learning theory, the quantity and quality of the labeled training data is the key to high-performance models. Unfortunately, even in the big data era, obtaining labeled instances in many real world domains such as biology and healthcare still incurs substantial cost. For example, the National Institute of Aging funded over $60 million to Alzheimer's disease neuroimaging initiative to study the disease and data are collected from less than 1000 patients. The limited sample size largely restricted the study of disease progression with many possible biomarkers.

Interestingly, while machine learning demands a large set of training samples to learn simple concepts, the learning process of human beings allows us link a learning task with what we have learned before and thus we are able to learn complicated cognitive concepts with much less training samples. Motivated by this human learning, the multi-task learning (MTL) paradigm learns related machine learning tasks simultaneously and performs inductive knowledge transfer among the tasks to improve their the generalization performance. MTL has many successful applications in board fields such as data mining, computer vision, text mining, bioinformatics and healthcare analytics [101, 228, 68]. For example, capturing temporal relatedness among multiple learning tasks allows researchers to build high performance disease progression models for Alzheimer's disease by transfer knowledge among time points [234].

One approach to learning multiple tasks is based on the regularized MTL framework [55]. The regularized MTL is extensively studied because of its flexibility to incorporate various learning objectives such as least squares, logistic regression and hinge loss, and to extend them with different kinds of assumptions on how tasks are related. Examples of such task relatedness regularizations include shared sets of features via sparsity inducing norms [120], shared low-dimensional subspace via the nuclear norm [10], and clustering structures via spectral k-means [232]. The same framework can accommodate more complicated assumptions

such as dirty models [88] and robust models [37, 66]. Moreover, efficient implementations have been developed for regularized MTL, which can be easily extended to new regularization terms [233].

Many of the regularized MTL methods heavily depend on the prior knowledge of task relatedness. In [54, 96, 171], for example, the prior knowledge of task relatedness are assumed to be known and is then transfered to regularization terms to guide the learning. However, the relationship for all tasks may not always be available. To address this problem, the multi-task relationship learning (MTRL) approaches [227, 58, 225] are studied to *learn the task relationship* in the form of a *task covariance matrix* from the data, representing how similar are the two tasks. These methods have been shown to be more effective than others in some learning problems. However, recall that in MTL the training samples are typically insufficient, and thus we may not always be able to infer reliable task relationship from the training data. If misleading task covariance matrix is learned from insufficient and noisy data, the subsequent knowledge transfer guided by such covariance information will not be performed towards the right direction as we expected, and lead to suboptimal models.

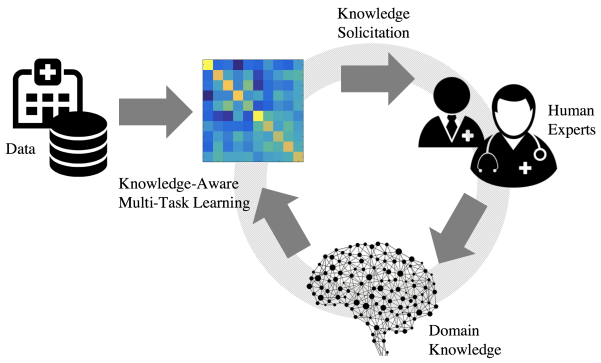


Figure 3.4: Overview of the proposed iMTRL framework, which involves human experts in the loop of multi-task learning. The framework consists of three phases: (1) *Knowledge-aware multi-task learning*: learning multi-task learning models from knowledge and data, (2) *Solicitation*: soliciting most informative knowledge from human experts using active learning based query strategy, (3) *Encoding*: encoding the domain knowledge to facilitate inductive transfer.

In many applications the human experts may have some domain knowledge about how some of tasks are related. For example, the physicians may indicate the predictive models of two disease models should be very similar due to the similarity in the their pathological pathways or dynamics in physiology. In those situations, soliciting and incorporating these domain knowledge in the learning could dramatically improve the generalization performance of learning models. Unfortunately, to the best of our knowledge, little research has been done on this area. We identified a few key questions in area: (1) What type of domain knowledge is suitable for guiding MTL? (2) How the solicited domain knowledge can be effectively incorporated into the MTL formulations; and (3) How the domain knowledge can be efficiently solicited?

To address the aforementioned challenges in MTL, this paper systematically investigated the above questions and propose a novel interactive multi-task relationship learning (iMTRL) framework. Specifically, in the iMTRL framework we propose to solicit the domain knowledge in the form of partial order between two pairs of tasks, which is equivalent to a pairwise relationship between two elements in the task covariance matrix. To effectively incorporate the partial order knowledge, we propose a knowledge aware MTRL (kMTRL) formulation, which learns a task covariance matrix constrained by the partial order relationships in the domain knowledge. We develop an efficient optimization algorithm for the proposed kMTRL. Moreover, since human labeling is very expensive even for weak supervision like tasks relationship, we propose an efficient query strategy for knowledge solicitation. We evaluate the proposed iMTRL framework on both synthetic and real datasets and demonstrate its efficiency and effectiveness.

Notation: We use lowercase letters to denote scalars, lowercase bold letters to denote vectors (e.g. $\mathbf{x}$), uppercase bold letters to denote matrices (e.g. $\mathbf{\Omega}$). We use $\mathbb{R}$ to denote the set of

real numbers and $\mathbb{R}_+(\mathbb{R}_{++})$ to denote the subset of non-negative (positive) ones. If $\mathbf{x} \in \mathbb{R}^d$, the p-norm of vector $\mathbf{x}$ is given by $\|\mathbf{x}\|_p = (\sum_{i=1}^d \|x_i\|^p)^{\frac{1}{p}}$. If $\mathbf{A} \in \mathbb{R}^{d\times K}$, we use $\mathbf{a}_j \in \mathbb{R}^d$ to denote the jth column of $\mathbf{A}$ and $\widetilde{\mathbf{a}}_i \in \mathbb{R}^T$ to denote the ith row of $\mathbf{A}$. For all $r, p > 1$, we define the $l_{p,q}$ norm of $\mathbf{A}$ as $\|\mathbf{A}\|_{p,q} = (\sum_{i=1}^d \|\widetilde{\mathbf{a}}_i\|_p^q)^{\frac{1}{p}}$. The set of K integers is denoted as $\mathbb{N}_K = [1, ..., K]$. We use $\mathbf{I}_d$ to denote a $d \times d$ identity matrix, and $\mathbf{1}_d$ to denote a d dimension vector with all elements are 1. Unless stated otherwise, all vectors are column vectors.

3.2.2 Related Work

Multi-task learning. MTL has been successfully applied to solve many challenging machine learning problems involving multiple related tasks. Recently the regularization based MTL approach has received a lot of attention because of its flexibility and efficient implementations. One major research direction in regularized MTL is to encode the relationship among tasks [54, 96, 58, 227, 171, 22]. The regularized MTL algorithms can be roughly classified into two types: the first involves assumptions about task relatedness, which are then "translated" into proper regularization terms in the regularization to infer a shared representation, that serves as the media of knowledge transfer. An example is the low-rank MTL [54, 96, 171], which seeks a shared low-dimensional subspace in task models, and the tasks are related through the shared subspace. One potential issue in such methods is that the prior knowledge may not always accurate and the assumption may not be suitable for all tasks. Later on some studies focus on infer the task relationship from the dataset [227, 58, 22], e.g, by learning a "covariance matrix" over tasks. Since the learned covariance matrix governing the knowledge transfer is also learned from data, these methods is heavily dependent on the quality and quantity of the training samples available. When an inaccurate task relationship is learned, it will lead to point the knowledge transfer in a wrong direction and lead to suboptimal models,

as will be shown in our empirical studies. To alleviate the problem of existing models, we propose an active learning framework which can interactively label the ground truth of task relationship into learning model and guide correct knowledge transfer.

Active Learning. There are two common categories of active learning: the pool based and the batch mode. The pool based active learning approaches select the most informative unlabeled instance iteratively, which is then labeled by user, with the goal of learning a better model with less efforts [173]. The selection process is often referred as a *query*. However, such sequential query selection strategy is inefficient in many cases, i.e. adding one labeled data point at a time is typically insufficient to substantially improve the performance of model, and thus the training procedure is very slow. In contrast, the batch-mode active learning approaches select a set of most informative query instances simultaneously. To the best of our knowledge, all previous active learning focus on how to select a group of most informative instances or training samples. Here, we instead propose a novel query strategy to query another type of supervision: task relationship. This supervision is intuitive but comes with a significant challenge, i.e., most previous active learning strategies cannot be directly applied.

In our study the task supervision is represented by partial orders which lead to pairwise constraints. There are a few previous studies on the effectiveness of the pairwise constraints [215, 70] under active learning framework. In [70], a clustering algorithm named Active-PCCA was proposed to consider whether two data points should be assigned to the same cluster or not, by which it biases the categorization towards the one expected. The most informative pairwise constraints are selected using the data points on the frontier of those least well-defined clusters. In [215], the authors studied a semi-supervised clustering algorithm with a query strategy to choose pairwise constraints by selecting the most informative instance, as well as data points in its neighborhoods. The pairwise constraints are in the form of Must-link

and Cannot-link, which restrict two data points should be in the same class or not. However, those methods are developed for clustering algorithms. How to select pairwise constraints on task relationship that are suitable for the MTL framework remains to be an open problem. In this work, we study query strategies for task relationship supervision, including one novel strategy based on the inconsistency of learning model.

Interactive Machine Learning. Interactive machine learning (IML) is a systematic way to include human in the learning loop, observing the results of learning and providing feedback to improve the generalization performance of learning model [6]. It has provided a natural way to integrate background knowledge into the learning procedure [7, 9, 210, 8]. For example, the system called "perception-based classication" (PBC) [9] has been pioneered to offer an interactive way to construct decision. The PBC is able to construct a smaller decision tree but the accuracy achieved doesn't has significant improve compared to other decision tree methods such as C4.5. The decision construction has been further extend in [210]. They also found out that users can build good models only when the visualization are apparent in two dimension. Manual classifier construction is not successful for large data set involving high dimension interaction. In [7], an end-user IML system (ReGroup) are proposed to be able to help people create customized groups in social networks. In [8], the authors developed an IML system named as (CueT) to learn the triaging decision about network alarm in a highly dynamic environment. In this paper, iMTRL is proposed to combine the domain knowledge in terms of task relationship to build learning models. Our work is exploring a completely novel problem compared to the previous studies in interactive machine learning.

3.2.3 Interactive Multi-Task Relationship Learning

In this section, we first review the strengths and potential issues of the multi-task relationship learning in Subsection 3.2.3.1, which motivate the overarching framework of the proposed interactive multi-task relationship learning (iMTRL) in Subsection 3.2.3.2. Subsection 3.2.3.3 presents the knowledge-aware MTRL (kMTRL) formulation and algorithm. Subsection 3.2.3.5 introduces the novel batch mode knowledge query strategy based on active learning.

3.2.3.1 Revisit the Multi-task Relationship Learning

Before discussing the iMTRL framework, we revisit the multi-task relationship learning (MTRL) [227], one popular MTL model that learns not only the prediction models but also task relationship. The MTRL framework has a well founded Bayesian background. Assume we have K related learning tasks, and in each task we are given a data matrix and their corresponding responses. Let d be the number of features. For the task k, we are given m samples and their corresponding responses, collectively denoted by $\mathbf{X}^k = [(\mathbf{x}_1^k)^T; (\mathbf{x}_2^k)^T; ...; (\mathbf{x}_m^k)^T] \in \mathbb{R}^{m \times d}$ and $\mathbf{y}^k \in \mathbb{R}^m$. We assume that the responses come from a linear combination of features with a Gaussian noise, so that for sample j from task i, we have $y_j^i = \mathbf{w}_i^T \mathbf{x}_j^i + b_i + \epsilon_i$, where distribution of the noise is given by $\epsilon_i \sim \mathcal{N}(0, \epsilon_i^2)$. The goal of the learning is to estimate the task parameters $\mathbf{W} = [\mathbf{w}_1, ..., \mathbf{w}_K]$ and bias term $\mathbf{b} = [b_1, ..., b_K]$ for all K tasks from data.

Based on the assumption we can write the likelihood of y_j^i given $\mathbf{x}_j^i, \mathbf{w}_i, b_i$, and ϵ_i is given by:

$$p(y_j^i | \mathbf{x}_j^i, \mathbf{w}_i, b_i, \epsilon_i) \sim \mathcal{N}(\mathbf{w}_i^T \mathbf{x}_j^i + b_i, \epsilon_i^2),$$

where $\mathcal{N}(\mathbf{m}, \Sigma)$ represents the multivariate distribution with mean $\mathbf{m}$ and covariance matrix

Σ [21]. The prior on $\mathbf{W} = (\mathbf{w}_1, ..., \mathbf{w}_K)$ is given by:

$$p(\mathbf{W}|\epsilon_i) \sim \left(\prod_{i=1}^{K} \mathcal{N}(\mathbf{w}_i|0_d, \sigma_i^2 \mathbf{I}_d) \right) q(\mathbf{W}),$$

where $\mathbf{I}_d \in \mathbb{R}^{d \times d}$ is the identity matrix. The first term is the extension of ridge prior to the multi-task learning setting, which controls the model complexity of each task $\mathbf{w}_i$. The second term refers to the task relationship, in which MTRL tries to learn the covariance of $\mathbf{W}$ using a matrix-variate normal distribution for $q(\mathbf{W})$

$$q(\mathbf{W}) = \mathcal{MN}_{d \times K}(\mathbf{W}|0_{d \times K}, \mathbf{I}_d \otimes \mathbf{\Omega}),$$

where $\mathcal{MN}_{d \times K}(\mathbf{M}, \mathbf{A} \otimes \mathbf{B})$ denotes matrix-variate normal distribution with mean $\mathbf{M} \in \mathbb{R}^{d \times K}$, row covariance matrix $\mathbf{A} \in \mathbb{R}^{d \times d}$ and column covariance matrix $\mathbf{B} \in \mathbb{R}^{K \times K}$. According to the Bayes's theorem, the posterior distribution for $\mathbf{W}$ is proportional to the product of the prior distribution and the likelihood function [21]:

$$p(\mathbf{W}|\mathbf{X}, \mathbf{y}, \mathbf{b}, \epsilon, \sigma, \mathbf{\Omega}) \propto p(\mathbf{y}|\mathbf{X}, \mathbf{W}, \mathbf{b}, \epsilon) p(\mathbf{W}|\mathbf{\Omega}, \sigma), \tag{3.9}$$

where $\mathbf{X}$ collectively denotes the data matrix for K tasks and $\mathbf{y} = [\mathbf{y}^1, ..., \mathbf{y}^k]$ denotes labels for all data points.

By taking negative logarithm of Eq. (3.9), the maximum a posteriori estimation of $\mathbf{W}$ and maximum likelihood estimation of $\mathbf{\Omega}$ is given by:

$$\min_{\mathbf{W}, \Omega} \sum_{k=1}^{K} \frac{1}{\epsilon_k^2} \|\mathbf{y} - \mathbf{X}^k \mathbf{w}_k - b_k \mathbf{1}_{n_k}\|_F^2 + \frac{1}{\sigma_k^2} \mathrm{tr}(\mathbf{W}\mathbf{W}^T) + \mathrm{tr}(\mathbf{W}\mathbf{\Omega}^{-1}\mathbf{W}^T) + d \ln(\mathbf{\Omega}). \tag{3.10}$$

In the above formulation, the last term $d\ln(\mathbf{\Omega})$ controls the complexity of $\mathbf{\Omega}$ and is a concave function. In order to obtain a convex objective function, the MTRL proposed to use $\mathrm{tr}(\mathbf{\Omega}) = 1$ instead to control the complexity and project $\mathbf{\Omega}$ to be a positive semi-definite matrix. As such, the objective function of MTRL is derived as follows:

$$\min_{\mathbf{W},\Omega} \sum_{k=1}^{K} \frac{1}{n_k} \|\mathbf{y}^k - \mathbf{X}^k \mathbf{w}_k - b_k \mathbf{1}_{n_k}\|_F^2 + \frac{\lambda_1}{2} \mathrm{tr}(\mathbf{W}\mathbf{W}^T) \\ + \frac{\lambda_2}{2} \mathrm{tr}(\mathbf{W}\mathbf{\Omega}^{-1}\mathbf{W}^T). \text{ s.t. } \mathbf{\Omega} \succeq 0, \mathrm{tr}(\mathbf{\Omega}) = 1 \tag{3.11}$$

An alternating algorithm is proposed in [227] to solve this formulation. The algorithm iteratively solves two steps: first it optimizes Eq (3.11) with respect to $\mathbf{W}$ and $\mathbf{b}$ when $\mathbf{\Omega}$ is fixed; it then optimizes the objective function with respective to $\mathbf{\Omega}$, which admits a closed-form solution:

$$\mathbf{\Omega} = (\mathbf{W}^T\mathbf{W})^{1/2} / \mathrm{tr}((\mathbf{W}^T\mathbf{W})^{1/2}). \tag{3.12}$$

We note that there is a feedback loop in the learning of MTRL as illustrated above. The MTRL achieves knowledge transfer among task models via the task relation matrix $\mathbf{\Omega}$, and the task models will be used to estimate $\mathbf{\Omega}$. If the $\mathbf{\Omega}$ can be learned correctly or can closely represent the true tasks relationship, it will benefit learning on the tasks parameters $\mathbf{W}$ by guiding the knowledge transfer in a good direction. In turn, the better tasks parameters will help the algorithm to identify a more accurate estimation of $\mathbf{\Omega}$. The positive feedback loop is the key to help building a good MTRL model. On the contrary, the training procedure will be biased to wrong direction once we keep getting misleading feedbacks in the loop. To be more specific, once data is either low-quality or insufficient-quantity, the $\mathbf{\Omega}$ will indicate an inaccurate direction to transfer the knowledge among tasks, which leads to a negative

feedback in the loop. This will end up learning a model with poor generalization performance, examples of which will be elaborated in the empirical studies.

Another remark is that in Eq. (3.11), due to the relaxation, the solution of $\mathbf{\Omega}$ is no longer the extract solution from the maximum likelihood estimation of column covariance matrix derived from Eq. (3.10). The advantages of the objective function in Eq. (3.11) compared to Eq. (3.10) have been discussed in details in [227]. We would like to further point out that the learned $\mathbf{\Omega}$ is actually a better representation of tasks relationship than the column covariance matrix. Recall that the covariance suggests the extent that elements in two vectors move to the same direction. Suppose we have tasks parameters $\mathbf{W} \in \mathbb{R}^{d \times K}$, the unbiased sample covariance can be computed by $\mathbf{C} = \mathbf{W}_c^T \mathbf{W}_c/(d-1)$, where $\mathbf{W}_c = \mathbf{W} - \mathbf{1}_d^T \mathbf{1}_d \mathbf{W}/d$ is the centralized tasks models. This measure is only meaningful when there are enough number of dimension d and the variance contains in tasks parameters. If $\mathbf{W} = [1, -2; 1, -2]$, the covariance matrix will return an all-zero matrix which will not indicate a correct relationship. Instead, an accurate estimation can be inferred by using Eq. (3.12). We can obtain a correlation matrix $\mathbf{Corr} = [1 - 1; -1, 1]$ from $\mathbf{\Omega}$.

The above discussions lead to two important conclusions: (1) The $\mathbf{\Omega}$ can indicate a genuine task relationship. (2) Maintaining an accurate $\mathbf{\Omega}$ is the key in this learning procedure.

3.2.3.2 The iMTRL Framework

In MTL scenarios, the quality and quantity of training data usually impose significant challenges to the learning algorithms. The task covariance matrix $\mathbf{\Omega}$ inferred from the data may not always give an accurate description of the true task relationship, which in turn would prevent effective knowledge transfer. Fortunately, in many real-world applications, human experts possess indispensable domain knowledge about relatedness among some tasks.

For example, when building models predicting different regions of the brain from clinical features, neuroscientist and medical researcher can reveal important relationship among the regions. As such, solicit feedback from human experts on task relationship and encode them as supervision is especially attractive. To achieve this goal we need to answer the following problems:

1. What type of knowledge representation can be efficiently solicited from human experts, and also can be used to effectively guide the learning algorithms?

2. How to design MTL algorithm that combines the domain knowledge and data-driven insights?

3. How to effectively solicit knowledge, reducing the workload of the human experts by supplying only the most important knowledge that affects the learning system?

In this paper we propose a framework of interactive multi-task Machine learning (iMTRL), which provides an integrated solution to address the above challenging questions. The framework is illustrated in Fig 3.4. The iMTRL is an iterative learning procedure that involves human experts in the loop. In each iteration, the learning procedure involves the following:

1. *Encoding.* The domain knowledge of task relationship is represented as partial orders, and can be encoded in the learning as pairwise constraints.

2. *Knowledge-Aware Multi-Task Learning.* We propose a novel MTL algorithm that infers models and task relationship from data and conform the solicited knowledge.

3. *Active Learning based Knowledge Query.* To maximize the usefulness of solicited knowledge, we propose a knowledge query strategy based on active learning.

It is natural and intuitive to use partial orders as the knowledge presentation for task relationship. Query a question that whether the task i and j are more related than task i and k is much easier than asking to which extent the task i and j are related to each other. For example, ith task and jth tasks has positive relationship while the ith task and kth task has negative relationship, then this relationship is represented by a partial order $\mathbf{\Omega}_{i,j} \geq \mathbf{\Omega}_{i,k}$. The focus of this paper is the algorithm development for iMTRL and we make a few assumptions to alleviate common issues in using this presentation and simply our discussions:

Assumption 1. *The domain knowledge acquired from human expert is accurate. The expert may choose not to label if he/she is not confident.*

Assumption 2. *The acquired partial orders are compatible, i.e. when* $\mathbf{\Omega}_{i,j} > \mathbf{\Omega}_{i,k}$ *and* $\mathbf{\Omega}_{i,k} > \mathbf{\Omega}_{k,p}$ *are established, the* $\mathbf{\Omega}_{i,j} < \mathbf{\Omega}_{k,p}$ *cannot be included.*

If this situation happens, we can discard the less important constraints and make the remain constraints be compatible. The importance of constraints can be measured by the Inconsistency which we will introduced in Definition 2.

3.2.3.3 A knowledge-aware extension of MTRL

Assume in the current iteration of iMTRL, our domain knowledge is stored in a set $\mathcal{T}$ defined by:

$$\mathcal{T} = \{\mathbf{\Omega} : \mathbf{\Omega}_{i_1,j_1} \geq \mathbf{\Omega}_{i_2,j_2} \ \forall (i_1, j_1, i_2, j_2) \in S\}, \tag{3.13}$$

where each pairwise constraint has specified a preferred half-space that an ideal solution $\mathbf{\Omega}$ should belong to, and the set S contains the indexes of tasks selected by our query strategy. The partial order information is more important than the magnitude of $\mathbf{\Omega}$. The reason is that if we multiply each element in $\mathbf{\Omega}$ with a scalar a, it's equal to solve the Eq. (3.15) replacing

λ_2 with $a\lambda_2$ [51]. Hence, the magnitude of elements in $\mathbf{\Omega}$ can be adjusted simultaneously without changing the results. But the order of pairs in $\mathbf{\Omega}$ is a more important structure to encode. These algorithmic advantages reinforced our choice of using pairwise constraints to represent domain knowledge.

We note that the constraints in Eq. (3.13) would lead to a trivial solution that $\mathbf{\Omega}_{i_1,j_1} = \mathbf{\Omega}_{i_2,j_2}\ \forall(i_1,j_1,i_2,j_2) \in S$, which is apparently not the effect we seek. To overcome this problem, we add a positive parameter c so that we can assure the elements in $\mathbf{\Omega}$ preserve the true pair wise order. Hence, the convex set $\mathcal{T}$ is changed to:

$$\mathcal{T} = \{\mathbf{\Omega} : \mathbf{\Omega}_{i_1,j_1} \geq \mathbf{\Omega}_{i_2,j_2} + c, \quad \forall(i_1,j_1,i_2,j_2) \in S\}. \tag{3.14}$$

The proposed knowledge-aware multi-task relationship (kMTRL) learning extends the MTRL by enforcing a feasible space for $\mathbf{\Omega}$ specified by $\mathcal{T}$. To this end, the kMTRL formulation is given by the following optimization problem:

$$\begin{aligned} \min_{\mathbf{W},\mathbf{b},\mathbf{\Omega}} \mathcal{F}(\mathbf{W},\mathbf{b},\mathbf{\Omega}) = &\sum_{k=1}^{K} \frac{1}{n_k} \|\mathbf{y}^k - \mathbf{X}^k\mathbf{w}_k - b_k\mathbf{1}_{n_k}\|_F^2 \\ &+ \frac{\lambda_1}{2}\mathrm{tr}(\mathbf{W}\mathbf{W}^T) + \frac{\lambda_2}{2}\mathrm{tr}(\mathbf{W}\mathbf{\Omega}^{-1}\mathbf{W}^T) \\ \text{s.t.} \quad &\mathbf{\Omega} \succeq 0,\ \mathrm{tr}(\mathbf{\Omega}) = 1,\ \mathbf{\Omega} \in \mathcal{T} \end{aligned} \tag{3.15}$$

We note that even though the problem of kMTRL is considered to be more challenging to solve than MTRL because of additional constraints introduced in $\mathcal{T}$, the solution space of kMTRL is much smaller because each constraint cuts the solution space in half, and the optimization algorithms may converge faster in this case.

3.2.3.4 Efficient Optimization for kMTRL

The proposed kMTRL is a convex optimization problem, and we propose to solve it using an alternating algorithm:

Step 1: We first optimize the objective function with respect to $\mathbf{W}$ and $\mathbf{b}$ given a fixed $\mathbf{\Omega}$. This step can either be solved using the linear system [227] or off-the-shelf solvers such as CVX [69] and FISTA [16]. Different solvers can be applied depending on the nature of the data: first order solvers such as FISTA is more scalable when there are many samples, while solving linear system can be more efficient as feature dimension is high.

Step 2: Given $\mathbf{W}$ and $\mathbf{b}$, the objective function with respect to $\mathbf{\Omega}$ is given by an analytical solution using Eq. (3.12).

Step 3: The $\mathbf{\Omega}$ is projected to the convex set:

$$\mathbf{T} = \{\mathbf{\Omega} | \mathbf{\Omega} \in \mathcal{T}, \mathbf{\Omega} \succeq 0, \mathrm{tr}(\mathbf{\Omega}) = 1\}$$

by solving the Euclidean projection problem below:

$$\min_{\mathbf{\Omega}} \|\mathbf{\Omega} - \hat{\mathbf{\Omega}}\|_F^2, \quad s.t.\ \mathbf{\Omega} \in \mathbf{T}$$

where the $\hat{\mathbf{\Omega}}$ is the analytical solution we obtained from the Eq. (3.12). This objective function can be solved efficiently using a successive projection algorithm [76] that iteratively projects the solution to each constraint in the set.

The KKT analysis [35] of the above optimization problem leads to the property summarized in Theorem 1, and leads to Algorithm 3.2. To simplify the discussion, we requires the true pair orders are in the form of $\mathbf{\Omega}_{i1,j1} \geq \mathbf{\Omega}_{i2,j2}$.

Theorem 1. *Suppose that* $\mathcal{T} = \{\mathbf{\Omega} : \mathbf{\Omega}_{i1,j1} \geq \mathbf{\Omega}_{i2,j2} + c\}$*, then, for any* $\mathbf{\Omega} \in \mathbb{R}^{K \times K}$*, the projection of* $\mathbf{\Omega}$ *to the convex set* $\mathcal{T}$ *is given by:*

$$\text{Proj}(\mathbf{\Omega}) = \mathbf{\Omega} \text{ if } \mathbf{\Omega} \in \mathcal{T},$$

otherwise

$$\text{Proj}(\mathbf{\Omega}) = \mathbf{\Omega}^* = \begin{cases} \mathbf{\Omega}^*_{i1,j1} = \frac{1}{2}(\mathbf{\Omega}_{i1,j1} + \mathbf{\Omega}_{i2,j2} + c) \\ \mathbf{\Omega}^*_{i2,j2} = \frac{1}{2}(\mathbf{\Omega}_{i1,j1} + \mathbf{\Omega}_{i2,j2} - c) \\ \mathbf{\Omega}^*_{p,q} = \mathbf{\Omega}_{p,q}, \ \forall (p,q) \neq (i1,j1) \text{ and } (i2,j2) \end{cases}$$

In practice, the term $\mathbf{W}^T\mathbf{W}$ is not guaranteed to be a full rank matrix. In fact, in a typical MTL setting $\mathbf{W}$ is a low rank matrix and thus the $\mathbf{\Omega}$ calculated by Eq. (3.12) is also a rank deficiency matrix. Moreover, recall that the operation that projects $\mathbf{\Omega}$ to a convex set has a very high chance lead to a singular matrix. The numerical problems during the inversion of the singular matrix $\mathbf{\Omega}$ will lead to a meaningless inverse of task relation matrix and corrupt the training procedure. Therefore, we propose to solve a perturbed version of our original objective function Eq. (3.15) as follows:

$$\begin{aligned} \min_{\mathbf{W},\mathbf{b},\mathbf{\Omega}} \mathcal{F}(\mathbf{W},\mathbf{b},\mathbf{\Omega}) &= \sum_{k=1}^{K} \frac{1}{n_k} \|\mathbf{y}^k - \mathbf{X}^k\mathbf{w}_k - b_k\mathbf{1}_{n_k}\|_F^2 \\ &+ \frac{\lambda_1}{2}\text{tr}(\mathbf{W}\mathbf{W}^T) + \frac{\lambda_2}{2}\text{tr}(\mathbf{\Omega}^{-1}(\mathbf{W}\mathbf{W}^T + \epsilon\mathbf{I})), \\ \text{s.t.} \qquad & \mathbf{\Omega} \succeq 0, \ \text{tr}(\mathbf{\Omega}) = 1, \ \mathbf{\Omega} \in \mathcal{T} \end{aligned} \tag{3.16}$$

where $\mathcal{T}$ follows the definition in Eq. (3.14). As a result, the analytical solution of $\mathbf{\Omega}$ in **Step**

Algorithm 3.1: knowledge-aware Multi-Task Relationship Learning (kMTRL)

Require: Training data $\{\mathbf{X}^k, \mathbf{y}^k\}_k^K$, constraints set S, regularization parameters λ_1, λ_2, a positive number c. Randomly initialize $\mathbf{W}^0$. $\mathbf{\Omega}^0 = \mathbf{I}/d$.

while $\mathbf{W}$ and $\mathbf{\Omega}$ are not converge **do**
  Compute $\{\mathbf{W}, \mathbf{b}\} = \arg\min_{\mathbf{W},\mathbf{b}} \mathcal{F}(\mathbf{W}, \mathbf{b}, \mathbf{\Omega})$
  Compute $\mathbf{\Omega}$ using Eq. (3.12)
  $\mathbf{\Omega}$ = Proj($\mathbf{\Omega}$, S, n, c)
end while
return $\mathbf{W}$, $\mathbf{b}$, $\mathbf{\Omega}$

Algorithm 3.2: Projection algorithm

Require: Task correlation matrix $\mathbf{\Omega}$, constraints set S, max iteration n, a positive number c.

for $i = 1, ..., n$ **do**
  while $\forall(i_1, j_1, i_2, j_2) \in S$ **do**
    if $\mathbf{\Omega}_{i_1,j_1} < \mathbf{\Omega}_{i_2,j_2}$ **then**
      $\mathbf{\Omega}_{i_1,j_1} = \frac{1}{2}(\mathbf{\Omega}_{i_1,j_1} + \mathbf{\Omega}_{i_2,j_2} + c)$
      $\mathbf{\Omega}_{i_1,j_1} = \frac{1}{2}(\mathbf{\Omega}_{i_1,j_1} + \mathbf{\Omega}_{i_2,j_2} - c)$
    end if
  end while
  Dynamic update $c = c \times 0.9$
  Project $\mathbf{\Omega}$ to be a positive semi-definite matrix
  if All constraints are satisfied **then**
    break
  end if
end for
return $\mathbf{\Omega}$

2. is thus replaced by the following:

$$\mathbf{\Omega} = (\mathbf{W}^T\mathbf{W} + \epsilon\mathbf{I})^{1/2}/\mathrm{tr}((\mathbf{W}^T\mathbf{W} + \epsilon\mathbf{I})^{1/2}. \tag{3.17}$$

The algorithm to solve the objective function Eq. (3.16) is presented in Algorithm 3.1. This algorithm can be interpreted as alternately performing supervised and unsupervised steps. In the supervised step we learn the task specific parameters ($\mathbf{W}$ and $\mathbf{b}$). In unsupervised step we get the task relationship matrix from the task parameters. Finally, the last supervised step we encode prior knowledge to the task relationship matrix $\mathbf{\Omega}$. We repeat the steps iteratively until converge.

Algorithm 3.3: Query Strategy of Pairwise Constraints

Require: The task correlation matrix $\mathbf{\Omega}$, the model parameter matrix $\mathbf{W}$ for all tasks, the number of pairwise constraints n selected to be query

Compute $\hat{\mathbf{\Omega}} = (\mathbf{W}^T\mathbf{W})^{1/2}/\mathrm{tr}((\mathbf{W}^T\mathbf{W})^{1/2})$
while $\forall(i_1, j_1, i_2, j_2)$ **do**
    Compute $\mathbf{\Omega}_{(i_1,j_1,i_2,j_2)}$ and $\hat{\mathbf{\Omega}}_{(i_1,j_1,i_2,j_2)}$
end while
while $\forall(i_1, j_1, i_2, j_2)$ **do**
    Compute $\mathrm{Inc}_{(i_1,j_1,i_2,j_2)}$
end while
Select n pairs with highest scores into the set $\mathcal{T}$
return $\mathcal{T}$

Algorithm 3.4: iMTRL framework

Require: Training sets $\{\mathbf{X}^k, \mathbf{y}^k\}_k^K$, number of selected queries $\mathbf{q}$, regularization parameters λ_1, λ_2, positive number c, $\mathcal{T}^0 = \emptyset$

for $i = 1, ..., n$ **do**
    $(\mathbf{\Omega}^i, \mathbf{W}^i, \mathbf{b}^i) = \mathrm{kMTRL}(\{\mathbf{X}^k, \mathbf{y}^k\}_k^K, \mathcal{T}^{i-1}, \lambda_1, \lambda_2, c)$
    $\mathcal{T}^i = \mathrm{query}(\mathbf{W}^i, \mathbf{\Omega}^i, \mathbf{q}_i)$
    $\mathcal{T}^i = \mathcal{T}^i \cup \mathcal{T}^{i-1}$
end for
$\mathbf{\Omega} = \mathbf{\Omega}^i$, $\mathbf{W} = \mathbf{W}^i$, $\mathbf{b} = \mathbf{b}^i$
return $\mathbf{\Omega}$, $\mathbf{W}$, $\mathbf{b}$

3.2.3.5 Batch Mode Pairwise Constraints Active learning

There are too many possible pairs for human experts to label them all, and thus the efficiency of iMTRL framework heavily relies on the quality of the pairs selected by the system. In this subsection, we discuss the important question of how to efficiently solicit the domain knowledge. Specifically, we would like to select the pairs that are most informative to the learning process. We propose an efficient heuristic query strategy as elaborated as follows.

We first design a score function for pairwise constraints based on the *inconsistency* in the model. To explain the inconsistency, we denote the analytical solution calculated by $\mathbf{W}$ as $\hat{\mathbf{\Omega}} = (\mathbf{W}^T\mathbf{W})^{1/2}/\mathrm{tr}((\mathbf{W}^T\mathbf{W})^{1/2})$ and the difference between elements $\mathbf{\Omega}_{i_1,j_1}$ and $\mathbf{\Omega}_{i_2,j_2}$ in the learned $\mathbf{\Omega}$ as $\mathbf{\Omega}_{(i_1,j_1,i_2,j_2)} = \mathbf{\Omega}_{i_1,j_1} - \mathbf{\Omega}_{i_2,j_2}$. Then inconsistency in the model is defined as follows:

Definition 2. *Inconsistency is defined as:*

$$Inc_{(i_1,j_1,i_2,j_2)} = sign(i_1, j_1, i_2, j_2)|\mathbf{\Omega}_{(i_1,j_1,i_2,j_2)} - \hat{\mathbf{\Omega}}_{(i_1,j_1,i_2,j_2)}|,$$

where $sign(i_1, j_1, i_2, j_2) = \dfrac{\mathbf{\Omega}_{(i_1,j_1,i_2,j_2)}\hat{\mathbf{\Omega}}_{(i_1,j_1,i_2,j_2)}}{|\mathbf{\Omega}_{(i_1,j_1,i_2,j_2)}\hat{\mathbf{\Omega}}_{(i_1,j_1,i_2,j_2)}|}$.

The $\text{Inc}_{(i_1,j_1,i_2,j_2)}$ represents two types of inconsistency:

Negative inconsistency: Given that the pairwise orders of two relationship matrices ($\mathbf{\Omega}$ and $\hat{\mathbf{\Omega}}$) are not consistent, i.e. $\mathbf{\Omega}_{i_1,j_1} > \mathbf{\Omega}_{i_2,j_2}$, but $\hat{\mathbf{\Omega}}_{i_1,j_1} < \hat{\mathbf{\Omega}}_{i_2,j_2}$ or vice versa, the $\text{Inc}_{(i_1,j_1,i_2,j_2)}$ is always negative. The smaller the $\text{Inc}_{(i_1,j_1,i_2,j_2)}$ is, the higher is the heuristic score.

Positive inconsistency: Given that the pairwise orders of two relationship matrices are consistent, then the inconsistency comes from $\|\mathbf{\Omega}_{(i_1,j_1,i_2,j_2)} - \hat{\mathbf{\Omega}}_{(i_1,j_1,i_2,j_2)}\|$. The larger the $\text{Inc}_{(i_1,j_1,i_2,j_2)}$ is, the higher is the heuristic score .

Note that the disorder of two pairs are more important that the difference of two pairs, and all pairs with negative inconsistency has the priority to be selected over those with positive inconsistency. At the first iteration, before adding any pairwise constraints into the training procedure, the learned $\mathbf{\Omega}$ is very close to the analytical solution calculated from $\mathbf{W}$, i.e. $\mathbf{\Omega}_{(i_1,j_1,i_2,j_2)} = \hat{\mathbf{\Omega}}_{(i_1,j_1,i_2,j_2)}$, except for the disturb of numerical term $\epsilon\mathbf{I}$. Therefore, the inconsistency is caused by some numerical issues in the first round. Therefore at the first training iteration, there is no negative inconsistency. As the number of constraints added into the model, the inconsistency will appear and the query strategy will become more effective in this situation. The Algorithm 3.3 describes the query strategy.

Finally, we summarize all procedures of iMTRL in Algorithm 3.4. The line 1 means there are n iterations learning procedures need to be conducted. The line 2 corresponds to the

knowledge-aware MTL step in our iMTRL framework. The line 3 is to solicit the domain knowledge and line 4 is to answer the query and encoding the knowledge into the model.

3.2.4 Experiments

3.2.4.1 Importance of High-Quality Task Relationship

In this subsection, we conduct experiments to show that encoding an accurate task relationship will significantly enhance the performance of MTRL. The effectiveness of MTRL has already been demonstrated in [227], in which the authors showed that MTRL can infer an accurate task relationship from a relatively clean dataset with sufficient training samples. Here we use a toy example to show that MTRL would infer a misleading relationship when noise presents and there are insufficient training samples. The toy dataset is generated as follows. There are three tasks with data sampled from $y = 3x + 10$, $y = -2x + 5$ and $y = 10x + 1$, respectively. For each tasks we generate 5 samples from a uniformly distribution in $[0, 10]$. The function outputs for three tasks are corrupted by a Gaussian noise with zero mean and standard variance equal to 30, 10 and 10, respectively. According to the generative regression functions, we expect that the correlation between the first task and third task is close to 1 and for the rest of pairs is close to -1. We use the linear kernel of MTRL with $\lambda_1 = 0.01$ and $\lambda_2 = 0.05$. The learned $\mathbf{\Omega}$ gives a correlation matrix as follows:

$$\begin{bmatrix} 1 & 0.9999 & -0.9999 \\ 0.9999 & 1 & -1 \\ -0.9999 & -1 & 1 \end{bmatrix}$$

From the above matrix we see that the learned relationship for task 1 is opposite to the supposed relationship, because of the highly noised data. This will leads to suboptimal solution

for $\mathbf{W} = [-3.7283, -2.6605, 3.0105]$, as compared to the ground truth $\mathbf{W} = [3, -2, 10]$. On the other hand, if we encode the true tasks relationship by fixing the $\mathbf{\Omega}$ to be the ground truth during the learning process, with the exactly same parameters setting as above. We can then learn a model $\mathbf{W} = [0.6850, -0.3878, 2.5840]$ that is closer to the ground truth in terms of l_2 norm and keeps the correct tasks relationship. This procedure is denoted as truth-encoded multi-task relationship learning (eMTRL) in this subsection.

This observation motivates us to further explore the effectiveness of eMTRL. We created synthetic dataset by generating $K = 10$ tasks parameters $\mathbf{w}_i$ and b_i from a uniform distribution between 0 and 1. Each task contains 25 samples drawn from a Gaussian distribution with zero means and the variance equals to 10. The function response is also corrupted by a Gaussian noise with zero mean and has a variance of 5. We split this synthetic dataset to training, validation and testing set. Out of the 25 samples for each tasks, 20% are for training, 30% for validation and 50% for testing. We fix the number of samples and the number of tasks, vary the number of features from 20 to 100. The parameters λ_1 and λ_2 have been tuned in $[1 \times 10^{-3}, 1 \times 10^{-2}, 1 \times 10^{-1}]$ and $[0, 1 \times 10^{-3}, 1 \times 10^{-2}, 1 \times 10^{-1}, 1, 10, 1 \times 10^{2}, 1 \times 10^{3}]$, respectively.

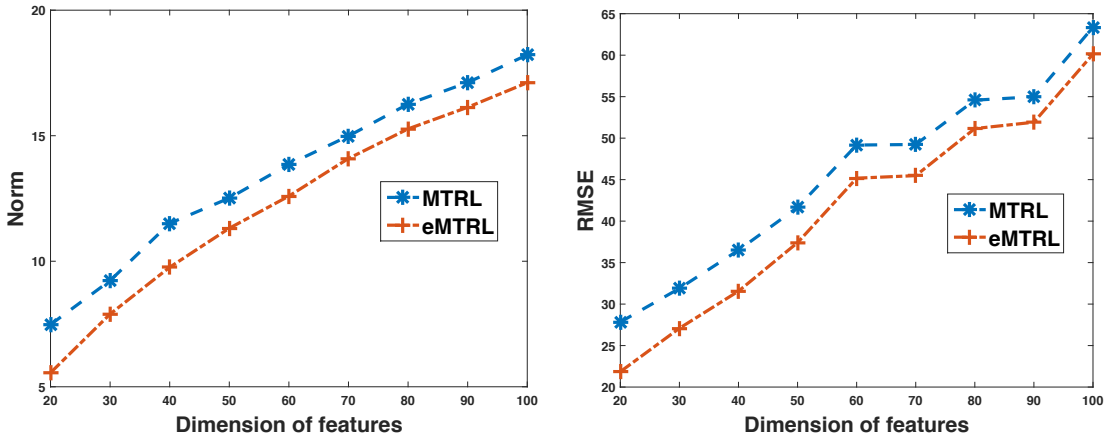
The performance has been evaluated using Root Mean Square Error (RMSE) and Frobenius norm between learned task model and the ground truth task model. The results shown in Figure 3.5 indicate that encoding the knowledge about task relationship will significantly benefit the prediction. Even though eMTRL is not a practical model because we can never know the true task relationship, the experimental results confirm that there is a huge potential to improve predictive performance if we can take advantage of domain knowledge. The experimental results in next section will show how to efficiently solicit and incorporate the domain knowledge about tasks relationship into the learning.

Table 3.4: The average RMSE of query and random strategy on testing dataset over 5 random splitting of training and validation samples.

· number of constraints	0	5	10	15	20	25	30	35	40
Query Strategy	1.1387	1.1267	1.1224	1.1117	1.1125	1.1101	1.1102	1.1137	1.1168
Random Selection	1.1387	1.1255	1.1390	1.1284	1.1165	1.1285	1.1379	1.1382	1.1364

Table 3.5: The RMSE comparison of kMTRL and baselines.

School	RR	MTL-L	MTL-l21	MTRL	kMTRL-20	kMTRL-40	kMTRL-60	kMTRL-80
5%	1.1737±0.0041	1.1799± 0.0047	1.176± 0.0043	1.0615 ± 0.0167	1.0584 ± 0.0128	1.0553 ± 0.0155	**1.0551** ± 0.0158	1.0551 ± 0.0159
10%	1.1428±0.0306	1.1485 ± 0.0293	1.1477 ± 0.0282	0.9872 ± 0.0057	0.9823 ± 0.0030	0.9805 ± 0.0014	**0.9803** ± 0.0018	0.9803 ± 0.0018
15%	1.0665±0.0395	1.0699 ± 0.0405	1.0700 ± 0.0399	0.9491 ± 0.0060	0.9334 ± 0.0057	**0.9321** ± 0.0081	0.9322 ± 0.0083	0.9323 ± 0.0082
20%	0.9756±0.0157	0.9774 ± 0.0153	0.9776 ± 0.0149	0.9047 ± 0.0031	0.8966 ± 0.0123	0.8906 ± 0.0123	0.8844 ± 0.0022	**0.8843** ± 0.0019
MMSE	RR	MTL-L	MTL-l21	MTRL	kMTRL-5	kMTRL-10	kMTRL-15	kMTRL-20
2%	0.9503± 0.1467	0.9319±0.1497	0.9314±0.1693	0.9106 ± 0.0976	0.9113 ± 0.0982	**0.9058** ± 0.0926	0.9058 ± 0.0926	0.9058 ± 0.0926



(a) (b)

Figure 3.5: Performance of MTRL and eMTRL as the number of features changing, in terms of (a) Frobenius norm and (b) RMSE. MTRL [227] learns both task models and task relationship at the same time, while eMTRL here learns the task models while the task relationship $\mathbf{\Omega}$ is fixed to ground truth, i.e. encoding the correct domain knowledge about the task relationship.

3.2.4.2 Effectiveness of Query Strategy

In this subsection, we conduct the experiments to show that encoding the domain knowledge in the form of partial order is useful. We follow the same synthetic data set with 20 feature dimension generated above. The same setting of splitting training, testing and validation dataset, and 5 fold random split validation are applied. The parameters λ_1 and λ_2 have been tuned in $[1 \times 10^{-3}, 1 \times 10^{-2}, 1 \times 10^{-1}]$ and $[0, 1 \times 10^{-3}, 1 \times 10^{-2}, 1 \times 10^{-1}, 1, 10, 1 \times 10^{2}, 1 \times 10^{3}]$, respectively. After the learning algorithm converges, we compare the the pairwise constraints

are chosen by the proposed query strategy and the randomly selected strategy. The results of two strategies are reported in Table 3.4. We see the trend that both of the proposed query strategy and the random selection reach better generalization performance as the number of incorporated pairwise constraints increases. To be more specific, the results in first column is worse than all the results using query strategy and most of the results using random selection. This show that solicit the domain knowledge in terms of pairwise constraints is effective. On the other hand, when comparing the results of the proposed query strategy and random selection, we see that our query strategy selects important pairwise constraints, leading to a better model than the random query. When the number of pairwise constraints is larger than 5, the proposed query strategy works consistently better than random selection.

3.2.4.3 Interactive Scheme for Query Strategy

To further analysis our query strategy, we also explore different interactive schemes in our query strategy. There are multiple ways to query a certain amount of partial orders. We can either query many times and each time with less labeling efforts, or vice versa. We use *kMTRL-a-b* to denote a total b constraints and each time we query a constraints (the human expert needs to interact with the system b/a times). The different interactive scheme will highly impact the user experience. For example, kMTRL-10-100 needs to query experts 10 times and experts need to label 10 constraints at each time. Also, it takes 10 training iterations which is much more expensive than other schemes. In contrast, kMTRL-100-100 only needs to query experts once, which is the most efficient scheme. However, this scheme cannot benefit from the iterative process of iMTRL. The pairwise constraints added in previous iterations will affect the model and won't be selected again. This will reveal other important constraints. Taking a one iteration scheme cannot utilize this information. The results are summarized

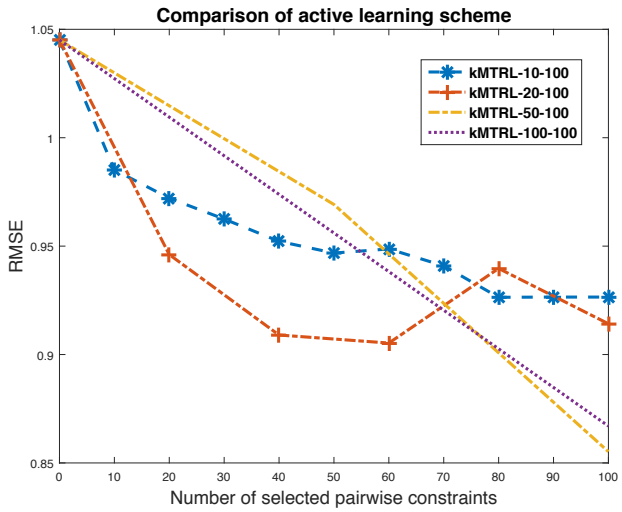


Figure 3.6: The averaged RMSE of kMTRL using different setting of query strategy. The kMTRL-10-100 means selecting 10 pairwise constraints at the end of each iteration, start from zero, add 10 pairwise constraints at a time, until 100 constraints. For all 4 schemes, kMTRL with zero constraints is equivalent to MTRL. Results are the average over 5 fold random splitting.

in Figure 3.6. We see that kMTRL-50-100 achieves the best performance. Therefore, the best scheme indicate that our query strategy is mostly effective when we balance the two parameters, and thus it does not require intensively interaction with experts and meanwhile utilizes the previous information effectively[1].

3.2.4.4 Performance on Real Datasets

The school dataset is a widely used benchmark dataset for multi-task regression problem. It contains 15372 students with 28 features from 139 secondary schools in the year of 1985, 1986 and 1987, provided by the Inner London Education Authority(ILEA). The task is to predict the score for students in 139 schools. The experimental settings are explained as follows. We first split the dataset into training, validation and testing datasets. The percentage of testing samples varies from 10% to 25% of all samples each tasks in original dataset. Taking the 10% testing dataset as an example, we perform 3-fold random split on the rest 90% data.

[1]Code is publicly available at `https://github.com/illidanlab/iMTL`

Each fold has 20% samples for training and 70% for testing. The same random splitting are applied to the three datasets.

Another real dataset we used here is Alzheimer's Disease Neuroimaging Initiative (ADNI) database[2]. The experimental setup is same as described in the paper [235]. The goal is to predict the successive cognition status of patients based on the measurements at the screening or the baseline visit. We use 2% samples for training, 10% for testing and the rest for validation. We also perform 3-fold random split on this dataset. The predictive performance of the competing methods listed below are reported on the real datasets:

- RR: This approach refers to ridge regression.
- MTL-L: This approach refers to the low-rank multi-task learning with trace norm regularization [10].
- MTL-L21: This approach refers to multi-task joint feature learning using $l_{2,1}$ norm that selects a subset of features shared by all tasks [121].
- MTRL: This approach refers to the multi-task relationship learning as we described in Section 3.2.3 [227].
- kMTRL-N: This approach refers to the proposed kMTRL method with N pairwise encoded into the model.

We tune the regularization parameters on $\mathbf{W}$ in $[1 \times 10^{-3}, 1 \times 10^{-2}, 1 \times 10^{-1}]$ for RR, MTL-L and MTL-L21. The regularization parameters λ_1 and λ_2 in Eq.(3.16) are tuned in $[1 \times 10^{-3}, 1 \times 10^{-2}, 1 \times 10^{-1}]$ and $[0, 1 \times 10^{-3}, 1 \times 10^{-2}, 1 \times 10^{-1}, 1, 10, 1 \times 10^{2}, 1 \times 10^{3}]$ respectively. The best parameters are selected based on the performance on the validation

[2]Data is publicly available at `http://adni.loni.usc.edu/`

Table 3.6: The name of the brain regions in Figure 3.8, where (C) denotes cortical parcellation and (W) denotes white matter parcellation.

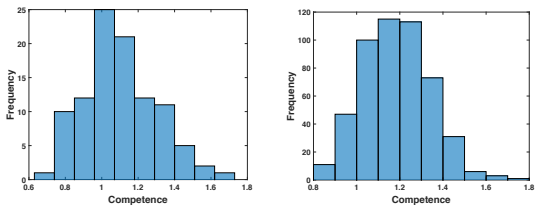
#	Intra-region	Inter-region (Row)	Inter-region (Column)
1	(C) Right Caudal Middle Frontal	(W) Right Putamen	(C) Right Inferior Temporal
2	(C) Right Pericalcarine	(W) Left Cerebral Cortex	(C) Left Rostral Middle Frontal
3	(W) Corpus Callosum Mid Anterior	(W) Right Ventral Diencephalon	(C) Right Pars Triangularis
4	(W) Right Cerebellum Cortex	(C) Right Caudal Anterior Cingulate	(C) Right Precentral
5	(W) Corpus Callosum Central	(C) Left Temporal Pole	(C) Right Medial Orbitofrontal
6	(C) Left Bank ssts	(C) Right Postcentral	(C) Left Pars Triangularis
7	(C) Right Pars Opercularis	(C) Right Precentral	(C) Right Superior Parietal
8	(C) Left Isthmus Cingulate	(W) Right Cerebral Cortex	(C) Right Inferior Parietal
9	(C) Left Supramarginal	(C) Left Isthmus Cingulate	(C) Left Pars Orbitalis
10	(C) Right Inferior Temporal	(C) Left Superior Frontal	(W) Corpus Callosum Central

set. The performance of learned models are measured by RMSE on the testing dataset. The experimental results are shown in Table 3.5, from which we see that kMTRL achieves the best results. In this experiment, we adopt the scheme kMTRL-20-80 for school dataset and kMTRL-5-20 for MMSE dataset as described in previous subsection.

3.2.5 Case Study: Brain Atrophy and Alzheimer's Disease

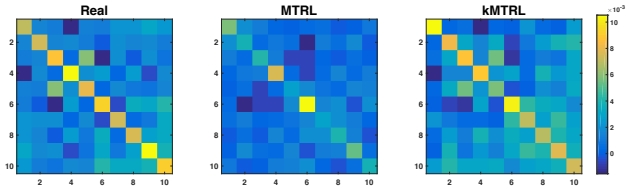
In this section we apply the proposed iMTRL framework to study the brain atrophy patterns and how the changes in the brain is associated to different clinical dementia scores and symptoms that are related to Alzheimer's disease (AD). It is estimated that there are currently 5 million Americans have AD, and AD has become one of the leading causes of death in the United States. Since AD is characterized by structural atrophy in the brain, there is a pressing demand of understanding how the brain atrophy is related to the progression of the disease.

In this work we study how the structural features of brain regions can be related to 51 cognitive markers such as, Alzheimer's Disease Assessment Scale (ADAS), clinical dementia rating (CDR), Global Deterioration Scale (GDS), Hachinski, Neuropsychological Battery, WMS-R Logic, and other neuropsychological assessment scores. We are interested in predicting the volume of brain areas extracted from the structural magnetic resonance imaging (MRI).

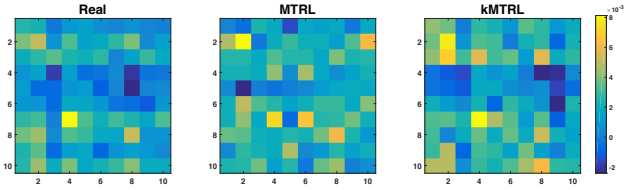


(a) (b)

Figure 3.7: The distribution of competence on (a) intra-region covariance and (b) inter-region covariance. kMTRL performs better than MTRL when competence>1. Higher competence indicates better performance achieved by kMTRL as compared to MTRL. We see in a majority of regions the kMTRL outperforms the MTRL.



(a) Intra-region covariance



(b) Inter-region covariance

Figure 3.8: Comparison of sub-matrices of covariance among (left) task covariance using 90% all data points that is considered as "ground truth", (middle) the covariance matrix learned via MTRL on 20% data and (right) the covariance matrix learned via kMTRL on 20% data with 0.8% pair-wise constraints queried by the proposed query scheme.

We use the ADNI cohort consisting 648 subjects whose baseline MRI images passed quality control. We used the FreeSurfer tool to extract the 99 brain volumes from regions of interest (ROIs) of the baseline MRI images. Considering the prediction of the volume of each ROI as a learning task, we thus have a collection of 99 learning tasks, with each task having 648 samples and 51 features. Since the brain regions are related during the aging process and Alzheimer's progression, the MTL approach can be used to improve the performance by considering such relatedness among brain regions.

We adopt the same experimental setting as in the previous experiments, where we compare the MTRL with the proposed kMTRL by querying and adding pair-wise expert knowledge

and inspecting the effectiveness of the queried task relationship supervision. We show the differences among the (1) task covariance using 90% all data points that is considered as "ground truth", (2) the covariance matrix learned via MTRL on 10% data and (3) the covariance matrix learned via kMTRL on 10% data with 0.8% pair-wise constraints queried by the proposed query scheme. Since the complete 99×99 covariance matrices are hard to visualize, we choose investigate two types of subregions of the covariance matrices: (a) a random intra region of the covariance of the size 10×10 (row regions and column regions are the same) and (b) a random inter region of the covariance of the size 10×10 (row regions and column regions are different). We define the *competence* metric to quantify how the quality of the sub-covariance:

$$\|\Omega_{\text{MTRL}} - \Omega_{\text{real}}\|_F / \|\Omega_{\text{kMTRL}} - \Omega_{\text{real}}\|_F, \tag{3.18}$$

where the kMTRL performs better than MTRL when competence > 1, and the higher the better. We repeatedly choose random sub-covariances and the distribution of the competence is shown in the Figure 3.7, indicating that in a majority of cases knowledge can improve relationship estimation.

We visualize two sub-covariance matrices in Figure 3.8, whose regions are shown in Table 3.6. In Figure 3.8(a), we see that the covariances from both the ground truth and the kMTRL discourage the positive knowledge transfer from *Right Cerebellum Cortex*, which agrees with the pathological characteristics of AD [182], where cerebellum does not correlate with the progression of AD. Also the positive correlation between *Corpus Callosum Mid Anterior* and *Corpus Callosum Central* is identified in both the ground truth and the kMTRL, and ignored by MTRL. The significant reduced corpus callosum size was previously reported

in AD studies [192], and the progression patterns of the two regions can be similar because of the physical distance between the two regions. Figure 3.8(b), we see that the unsubstantiated strong correlation between *Right Precentral* and *Left Pars Triangularis* as found in MTRL has been largely suppressed by the domain knowledge. However, since we only specified partial order relationship, there are chances the proposed kMTRL algorithm may "over-utilize" the supervision, as we notice that some unsubstantiated positive correlations involving *Right Ventral Diencephalon* are introduced to the covariance. We plan to further elaborate the findings and clinical insights of AD and dementia in the journal extension of this paper.

Chapter 4

Data-Driven Collaborative Learning

In this chapter, we discuss data-driven collaboration in reinforcement learning. More specifically, we first propose a collaborative deep reinforcement learning framework that can address the knowledge transfer among heterogeneous tasks. Under this framework, we propose deep knowledge distillation to adaptively align the domain of different tasks with the utilization of deep alignment network. Secondly, we further construct heterogeneous learning agents in the same task to improve its sample-efficiency. The central idea is to disentangle exploration and exploitation agents and then conduct data-driven transfer through imitation learning, which leads to an off-policy learning framework largely facilitates the learning efficiency. The off-policy learning framework uses generalized policy iteration for exploration and exploits the stableness of supervised learning for deriving policy, which accomplishes the unbiasedness, variance reduction, off-policy learning, and sample efficiency at the same time.

4.1 Collaborative Deep Reinforcement Learning

4.1.1 Introduction

On the other hand, the study of human learning has largely advanced the design of machine learning and data mining algorithms, especially in reinforcement learning and transfer learning. The recent success of deep reinforcement learning (DRL) has attracted increasing attention

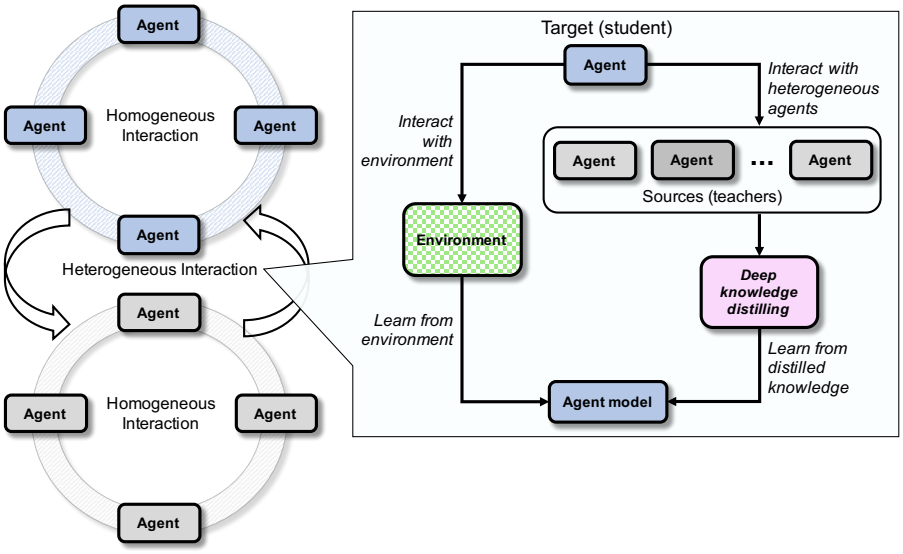


Figure 4.1: Illustration of Collaborative Deep Reinforcement Learning Framework.

from the community, as DRL can discover very competitive strategies by having learning agents interacting with a given environment and using rewards from the environment as the supervision (e.g., [132, 86, 107, 174]). Even though most of current research on DRL has focused on learning from games, it possesses great transformative power to impact many industries with data mining and machine learning techniques such as clinical decision support [193], marketing [3], finance [2], visual navigation [236], and autonomous driving [32]. Although there are many existing efforts towards effective algorithms for DRL [131, 137], the computational cost still imposes significant challenges as training DRL for even a simple game such as PONG [24] remains very expensive. The underlying reasons for the obstacle of efficient training mainly lie in two aspects: First, the supervision (rewards) from the environment is very sparse and implicit during training. It may take an agent hundreds or even thousands actions to get a single reward, and which actions that actually lead to this reward are ambiguous. Besides the insufficient supervision, training deep neural network itself takes lots of computational resources.

Due to the aforementioned difficulties, performing knowledge transfer from other related tasks or well-trained deep models to facilitate training has drawn lots of attention in the community [159, 191, 151, 86, 166]. Existing transfer learning can be categorized into two classes according to the means that knowledge is transferred: *data transfer* [82, 151, 166] and *model transfer* [53, 227, 229, 151]. Model transfer methods implement knowledge transfer from introducing inductive bias during the learning, and has been extensively studied in both transfer learning/multi-task learning (MTL) community and deep learning community. For example, in the regularized MTL models such as [55, 233], tasks with the same feature space are related through some structured regularization. Another example is the multi-task deep neural network, where different tasks share parts of the network structures [229]. One obvious disadvantage of model transfer is the lack of flexibility: usually the feasibility of inductive transfer has largely restricted the model structure of learning task, which makes it not practical in DRL because for different tasks the optimal model structures may be radically different. On the other hand, the recently developed data transfer (also known as knowledge distillation or mimic learning) [82, 166, 151] embeds the source model knowledge into data points. Then they are used as knowledge bridge to train target models, which can have different structures as compared to the source model [82, 25]. Because of the structural flexibility, the data transfer is especially suitable to deal with structure variant models.

There are two situations that transfer learning methods are essential in DRL:

Certificated heterogeneous transfer. Training a DRL agent is computational expensive. If we have a well-trained model, it will be beneficial to assist the learning of other tasks by transferring knowledge from this model. Therefore we consider following research question: Given one *certificated* task (i.e. the model is well-designed, extensively trained and performs very well), how can we maximize the information that can be used in the training of other

related tasks? Some model transfer approaches directly use the weights from the trained model to initialize the new task [151], which can only be done when the model structures are the same. Thus, this strict requirement has largely limited its general applicability on DRL. On the other hand, the initialization may not work well if the tasks are significantly different from each other in nature [151]. This challenge could be partially solved by generating an intermediate dataset (logits) from the existing model to help learning the new task. However, new problems would arise when we are transferring knowledge between *heterogeneous tasks*. Not only the action spaces are different in dimension, the intrinsic action probability distributions and semantic meanings of two tasks could differ a lot. Specifically, one action in PONG may refer to move the paddle upwards while the same action index in RIVERRAID [24] would correspond to fire. Therefore, the distilled dataset generated from the trained source task cannot be directly used to train the heterogeneous target task. In this scenario, the first key challenge we identified in this work is that how to conduct data transfer among heterogeneous tasks so that we can maximally utilize the information from a certificated model while still maintain the flexibility of model design for new tasks. During the transfer, the transferred knowledge from other tasks may contradict to the knowledge that agents learned from its environment. One recently work [159] use an attention network selective eliminate transfer if the contradiction presents, which is not suitable in this setting since we are given a certificated task to transfer. Hence, the second challenge is how to resolve the conflict and perform a meaningful transfer.

Lack of expertise. A more general desired but also more challenging scenario is that DRL agents are trained for multiple heterogeneous tasks without any pre-trained models available. One feasible way to conduct transfer under this scenario is that agents of multiple tasks share part of their network parameters [229, 166]. However, an inevitable drawback is, multiple

models lose their task-specific designs since the shared part needs to be the same. Another solution is to learn a domain invariant feature space shared by all tasks [4]. However, some task-specific information is often lost while converting the original state to a new feature subspace. In this case, an intriguing questions is that: can we design a framework that fully utilizes the original environment information and meanwhile leverages the knowledge transferred from other tasks?

This paper investigates the aforementioned problems systematically and proposes a novel Collaborative Deep Reinforcement Learning (CDRL) framework (illustrated in Figure 4.1) to resolve them. Our major contribution is threefold:

- First, in order to transfer knowledge among heterogeneous tasks while remaining the task-specific design of model structure, a novel deep knowledge distillation is proposed to address the heterogeneity among tasks, with the utilization of deep alignment network designed for the domain adaptation.

- Second, in order to incorporate the transferred knowledge from heterogeneous tasks into the online training of current learning agents, similar to human collaborative learning, an efficient collaborative asynchronously advantage actor-critic learning (cA3C) algorithm is developed under the CDRL framework. In cA3C, the target agents are able to learn from environments and its peers simultaneously, which also ensure the information from original environment is sufficiently utilized. Further, the knowledge conflict among different tasks is resolved by adding an extra distillation layer to the policy network under CDRL framework, as well.

- Last but not least we present extensive empirical studies on OpenAI gym to evaluate the proposed CDRL framework and demonstrate its effectiveness by achieving more

than 10% performance improvement compared to the current state-of-the-art.

Notations: In this paper, we use teacher network/source task denotes the network/task contained the knowledge to be transferred to others. Similarly, the student network/target task is referred to those tasks utilizing the knowledge transferred from others to facilitate its own training. The expert network denotes the network that has already reached a relative high averaged reward in its own environment. In DRL, an agent is represented by a policy network and a value network that share a set of parameters. Homogeneous agents denotes agents that perform and learn under independent copies of same environment. Heterogeneous agents refer to those agents that are trained in different environments.

4.1.2 Related Work

Multi-agent learning. One closely related area to our work is multi-agent reinforcement learning. A multi-agent system includes a set of agents interacting in one environment. Meanwhile they could potentially interact with each other [28, 103, 73, 190]. In collaborative multi-agent reinforcement learning, agents work together to maximize a shared reward measurement [103, 73]. There is a clear distinction between the proposed CDRL framework and multi-agent reinforcement learning. In CDRL, each agent interacts with its own environment copy and the goal is to maximize the reward of the target agents. The formal definition of the proposed framework is given in Section 4.1.5.

Transfer learning. Another relevant research topic is domain adaption in the field of transfer learning [149, 183, 200]. The authors in [183] proposed a two-stage domain adaptation framework that considers the differences among marginal probability distributions of domains, as well as conditional probability distributions of tasks. The method first re-weights the data

from the source domain using Maximum Mean Discrepancy and then re-weights the predictive function in the source domain to reduce the difference on conditional probabilities. In [200], the marginal distributions of the source and the target domain are aligned by training a network, which maps inputs into a domain invariant representation. Also, knowledge distillation was directly utilized to align the source and target class distribution. One clear limitation here is that the source domain and the target domain are required to have the same dimensionality (i.e. number of classes) with same semantics meanings, which is not the case in our deep knowledge distillation.

In [4], an invariant feature space is learned to transfer skills between two agents. However, projecting the state into a feature space would lose information contained in the original state. There is a trade-off between learning the common feature space and preserving the maximum information from the original state. In our work, we use data generated by intermediate outputs in the knowledge transfer instead of a shared space. Our approach thus retains complete information from the environment and ensures high quality transfer. The recently proposed A2T approach [159] can avoid negative transfer among different tasks. However, it is possible that some negative transfer cases may because of the inappropriate design of transfer algorithms. In our work, we show that we can perform successful transfer among tasks that seemingly cause negative transfer.

Knowledge transfer in deep learning. Since the training of each agent in an environment can be considered as a learning task, and the knowledge transfer among multiple tasks belongs to the study of multi-task learning. The multi-task deep neural network (MTDNN) [229] transfers knowledge among tasks by sharing parameters of several low-level layers. Since the low-level layers can be considered to perform representation learning, the MTDNN is learning a shared representation for inputs, which is then used by high-level layers in

the network. Different learning tasks are related to each other via this shared feature representation. In the proposed CDRL, we do not use the share representation due to the inevitable information loss when we project the inputs into a shared representation. We instead perform explicitly knowledge transfer among tasks by distilling knowledge that are independent of model structures. In [82], the authors proposed to compress cumbersome models (teachers) to more simple models (students), where the simple models are trained by a dataset (knowledge) distilled from the teachers. However, this approach cannot handle the transfer among heterogeneous tasks, which is one key challenge we addressed in this paper.

Knowledge transfer in deep reinforcement learning. Knowledge transfer is also studied in deep reinforcement learning. [131] proposed multi-threaded asynchronous variants of several most advanced deep reinforcement learning methods including Sarsa, Q-learning, Q-learning and advantage actor-critic. Among all those methods, asynchronous advantage actor-critic (A3C) achieves the best performance. Instead of using experience replay as in previous work, A3C stabilizes the training procedure by training different agents in parallel using different exploration strategies. This was shown to converge much faster than previous methods and use less computational resources. We show in Section 4.1.5 that the A3C is subsumed to the proposed CDRL as a special case. In [151], a single multi-task policy network is trained by utilizing a set of expert Deep Q-Network (DQN) of source games. At this stage, the goal is to obtain a policy network that can play source games as close to experts as possible. The second step is to transfer the knowledge from source tasks to a new but related target task. The knowledge is transferred by using the DQN in last step as the initialization of the DQN for the new task. As such, the training time of the new task can be significantly reduced. Different from their approach, the proposed transfer strategy is not to directly mimic experts' actions or initialize by a pre-trained model. In [166], knowledge distillation

was adopted to train a multi-task model that outperforms single task models of some tasks. The experts for all tasks are firstly acquired by single task learning. The intermediate outputs from each expert are then distilled to a similar multi-task network with an extra controller layer to coordinate different action sets. One clear limitation is that major components of the model are exactly the same for different tasks, which may lead to degraded performance on some tasks. In our work, transfer can happen even when there are no experts available. Also, our method allow each task to have their own model structures. Furthermore, even the model structures are the same for multiple tasks, the tasks are not trained to improve the performance of other tasks (i.e. it does not mimic experts from other tasks directly). Therefore our model can focus on maximizing its own reward, instead of being distracted by others.

4.1.3 Background

4.1.3.1 Reinforcement Learning

In this work, we consider the standard reinforcement learning setting where each agent interacts with it's own environment over a number of discrete time steps. Given the current state $s_t \in \mathcal{S}$ at step t, agent g_i selects an action $a_t \in \mathcal{A}$ according to its policy $\pi(a_t|s_t)$, and receives a reward r_{t+1} from the environment. The goal of the agent is to choose an action a_t at step t that maximize the sum of future rewards $\{r_t\}$ in a decaying manner: $R_t = \sum_{i=0}^{\infty} \gamma^i r_{t+i}$, where scalar $\gamma \in (0, 1]$ is a discount rate. Based on the policy π of this agent, we can further define a state value function $V(s_t) = E[R_t|s = s_t]$, which estimates the expected discounted return starting from state s_t, taking actions following policy π until the game ends. The goal in reinforcement learning algorithm is to maximize the expected return.

Since we are mainly discussing one specific agent's design and behavior throughout the paper, we leave out the notation of the agent index for conciseness.

4.1.3.2 Asynchronous Advantage actor-critic algorithm (A3C)

The asynchronous advantage actor-critic (A3C) algorithm [131] launches multiple agents in parallel and asynchronously updates a global shared target policy network $\pi(a|s, \theta_p)$ as well as a value network $V(s, \theta_v)$. parametrized by θ_p and θ_v, respectively. Each agent interacts with the environment, independently. At each step t the agent takes an action based on the probability distribution generated by policy network. After playing a n-step rollout or reaching the terminal state, the rewards are used to compute the advantage with the output of value function. The updates of policy network is conducted by applying the gradient:

$$\nabla_{\theta_p} \log \pi(a_t|s_t; \theta_p) A(s_t, a_t; \theta_v),$$

where the advantage function $A(s_t, a_t; \theta_v)$ is given by:

$$\sum_{i=0}^{T-t-1} \gamma^i r_{t+i} + \gamma^{T-t} V(s_T; \theta_v) - V(s_t; \theta_v).$$

Term T represents the step number for the last step of this rollout, it is either the max number of rollout steps or the number of steps from t to the terminal state. The update of value network is to minimize the squared difference between the environment rewards and value function outputs, i.e.,

$$\min_{\theta_v} (\sum_{i=0}^{T-t-1} \gamma^i r_{t+i} + \gamma^{T-t} V(s_T; \theta_v) - V(s_t; \theta_v))^2.$$

The policy network and the value network share the same layers except for the last output layer. An entropy regularization of policy π is added to improve exploration, as well.

4.1.3.3 Knowledge distillation

Knowledge distillation [82] is a transfer learning approach that distills the knowledge from a teacher network to a student network using a temperature parameterized "soft targets" (i.e. a probability distribution over a set of classes). It has been shown that it can accelerate the training with less data since the gradient from "soft targets" contains much more information than the gradient obtained from "hard targets" (e.g. 0, 1 supervision).

To be more specific, logits vector $\mathbf{z} \in \mathcal{R}^d$ for d actions can be converted to a probability distribution $\mathbf{h} \in (0,1)^d$ by a softmax function, raised with temperature τ:

$$\mathbf{h}(i) = \text{softmax}(\mathbf{z}/\tau)_i = \frac{exp(\mathbf{z}(i)/\tau)}{\sum_j exp(\mathbf{z}(j)/\tau)}, \tag{4.1}$$

where $\mathbf{h}(i)$ and $\mathbf{z}(i)$ denotes the i-th entry of $\mathbf{h}$ and $\mathbf{z}$, respectively.

Then the knowledge distillation can be completed by optimize the following Kullback-Leibler divergence (KL) with temperature τ [166, 82].

$$L_{KL}(D, \theta_p^\beta) = \sum_{t=1} \text{softmax}(\mathbf{z}_t^\alpha/\tau) \ln \frac{\text{softmax}(\mathbf{z}_t^\alpha/\tau)}{\text{softmax}(\mathbf{z}_t^\beta)} \tag{4.2}$$

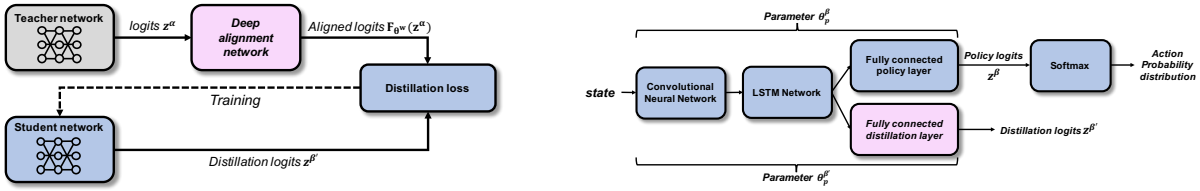
where $\mathbf{z}_t^\alpha$ is the logits vector from teacher network (notation α represents teacher) at step t, while $\mathbf{z}_t^\beta$ is the logits vector from student network (notation β represents student) of this step. θ_p^β denotes the parameters of the student policy network. D is a set of logits from teacher network.

4.1.4 Collaborative deep reinforcement learning framework

In this section, we introduce the proposed collaborative deep reinforcement learning (CDRL) framework. Under this framework, a collaborative Asynchronous Advantage Actor-Critic (cA3C) algorithm is proposed to confirm the effectiveness of the collaborative approach. Before we introduce our method in details, one underlying assumption we used is as follows:

Assumption 3. *If there is a universe that contains all the tasks* $E = \{e_1, e_2, ..., e_\infty\}$ *and* k_i *represents the corresponding knowledge to master each task* e_i*, then* $\forall i, j, k_i \cap k_j \neq \emptyset$*.*

This is a formal description of our common sense that any pair of tasks are not absolutely isolated from each other, which has been implicitly used as a fundamental assumption by most prior transfer learning studies [151, 166, 55, 37, 235].Therefore, we focus on mining the shared knowledge across multiple tasks instead of providing strategy selecting tasks that share knowledge as much as possible, which remains to be unsolved and may lead to our future work. The goal here is to utilize the existing knowledge as well as possible. For example, we may only have a well-trained expert on playing Pong game, and we want to utilize its expertise to help us perform better on other games. This is one of the situations that can be solved by our collaborative deep reinforcement learning framework.



(a) Distillation procedure (b) Student network structure.

Figure 4.2: Deep knowledge distillation. In (a), the teacher's output logits $\mathbf{z}^\alpha$ is mapped through a deep alignment network and the aligned logits $\mathcal{F}_\theta\omega(\mathbf{z}^\alpha)$ is used as the supervision to train the student. In (b), the extra fully connected layer for distillation is added for learning knowledge from teacher. For simplicity's sake, time step t is omitted here.

4.1.5 Collaborative deep reinforcement learning

In deep reinforcement learning, since the training of agents are computational expensive, the well-trained agents should be further utilized as source agents (agents where we transferred knowledge from) to facilitate the training of target agents (agents that are provided with the extra knowledge from source). In order to incorporate this type of collaboration to the training of DRL agents, we formally define the collaborative deep reinforcement learning (CDRL) framework as follows:

Definition 3. *Given m independent environments $\{\varepsilon_1, \varepsilon_2, ..., \varepsilon_m\}$ of m tasks $\{e_1, e_2, ..., e_m\}$, the corresponding m agents $\{g_1, g_2, ..., g_m\}$ are collaboratively trained in parallel to maximize the rewards (master each task) with respect to target agents.*

- *Environments. There is no restriction on the environments: The m environments can be totally different or with some duplications.*

- *In parallel. Each environment ε_i only interacts with the one corresponding agent g_i, i.e., the action a_t^j from agent g_j at step t has no influence on the state s_{t+1}^i in $\varepsilon_i, \forall i \neq j$.*

- *Collaboratively. The training procedure of agent g_i consists of interacting with environment ε_i and interacting with other agents as well. The agent g_i is not necessary to be at same level as "collaborative" defined in cognitive science [50]. E.g., g_1 can be an expert for task e_1 (environment ε_1) while he is helping agent g_2 which is a student agent in task e_2.*

- *Target agents. The goal of CDRL can be set as maximizing the rewards that agent g_i obtains in environment ε_i with the help of interacting with other agents, similar to inductive transfer learning where g_i is the target agent for target task and others*

are source tasks. The knowledge is transfered from source to target g_i by interaction. When we set the goal to maximize the rewards of multiple agents jointly, it is similar to multi-task learning where all tasks are source tasks and target tasks at the same time.

Notice that our definition is very different from the previously defined collaborative multiagent Markow Decision Process (collaborative multiagent MDP) [103, 73] where a set of agents select a global joint action to maximize the sum of their individual rewards and the environment is transitted to a new state based on that joint action. First, MDP is not a requirement in CDRL framework. Second, in CDRL, each agent has its own copy of environment and maximizes its own cumulative rewards. The goal of collaboration is to improve the performance of collaborative agents, compared with isolated ones, which is different from maximizing the sum of global rewards in collaborative multiagent MDP. Third, CDRL focuses on how agents collaborate among heterogeneous environments, instead of how joint action affects the rewards. In CDRL, different agents are acting in parallel, the actions taken by other agents won't directly influence current agent's rewards. While in collaborative multiagent MDP, the agents must coordinate their action choices since the rewards will be directly affected by the action choices of other agents.

Furthermore, CDRL includes different types of interaction, which makes this a general framework. For example, the current state-of-the-art is A3C [131] can be categorized as one homogeneous CDRL method with advantage actor-critic interaction. Specifically, multiple agents in A3C are trained in parallel with the same environment. All agents first synchronize parameters from a global network, and then update the global network with their individual gradients. This procedure can be seen as each agent maintains its own model (a different version of global network) and interacts with other agents by sending and receiving gradients.

In this paper, we propose a novel interaction method named deep knowledge distillation

under the CDRL framework. It is worth noting that the interaction in A3C only deals with the homogeneous tasks, i.e. all agents have the same environment and the same model structure so that their gradients can be accumulated and interacted. By deep knowledge distillation, the interaction can be conducted among heterogeneous tasks.

4.1.6 Deep knowledge distillation

As we introduced before, knowledge distillation [82] is trying to train a student network that can behave similarly to the teacher network by utilizing the logits from the teacher as supervision. However, transferring the knowledge among heterogeneous tasks faces several difficulties. First, the action spaces of different tasks may have different dimensions. Second, even if the dimensionality of action space is same among tasks, the action probability distributions for different tasks could vary a lot, as we illustrated in Figure 4.5 (a) and (b). Thus, the action patterns represented by the logits of different policy networks are usually different from task to task. If we directly force a student network to mimic the action pattern of a teacher network for a different task, it could be trained in a wrong direction, and finally ends up with worse performance than isolated training. In fact, this suspect has been empirically verified in our experiments.

Based on the above observation, we propose deep knowledge distillation to transfer knowledge between heterogeneous tasks. As illustrated in Figure 4.2 (a), the approach for deep knowledge distillation is straightforward. We use a deep alignment network to map the logits of the teacher network from a heterogeneous source task e^{α} (environment ε^{α}), then the logits is used as our supervision to update the student network of target task e^{β} (environment ε^{β}). This procedure is performed by minimizing following objective function over student

policy network parameters $\theta_p^{\beta\prime}$:

$$L_{KL}(D, \theta_p^{\beta\prime}, \tau) = \sum_t l_{KL}(\mathcal{F}_{\theta\omega}(\mathbf{z}_t^{\alpha}), \mathbf{z}_t^{\beta\prime}, \tau), \tag{4.3}$$

where

$$l_{KL}(\mathcal{F}_{\theta\omega}(\mathbf{z}_t^{\alpha}), \mathbf{z}_t^{\beta\prime}, \tau) = \text{softmax}(\mathcal{F}_{\theta\omega}(\mathbf{z}_t^{\alpha})/\tau) \ln \frac{\text{softmax}(\mathcal{F}_{\theta\omega}(\mathbf{z}_t^{\alpha})/\tau)}{\text{softmax}(\mathbf{z}_t^{\beta\prime})}.$$

Here θ^{ω} denotes the parameters of the deep alignment network, which transfers the logits $\mathbf{z}_t^{\alpha}$ from the teacher policy network for knowledge distillation by function $\mathcal{F}_{\theta\omega}(\mathbf{z}_t^{\alpha})$ at step t. As we show in Figure 4.2 (b), θ_p^{β} is the student policy network parameters (including parameters of CNN, LSTM and policy layer) for task e^{β}, while $\theta_p^{\beta\prime}$ denotes student network parameters of CNN, LSTM and distillation layer. It is clear that the distillation logits $\mathbf{z}_t^{\beta\prime}$ from the student network does not determine the action probability distribution directly, which is established by the policy logits $\mathbf{z}_t^{\beta}$, as illustrated in Figure 4.2 (b). We add another fully connected distillation layer to deal with the mismatch of action space dimensionality and the contradiction of the transferred knowledge from source domain and the learned knowledge from target domain. The input to both of the teacher and the student network is the state of environment ε^{β} of target task e^{β}. It means that we want to transfer the expertise from an expert of task e^{α} towards the current state. Symbol D is a set of logits from the teacher network in one batch and τ is the temperature same as described in Eq (4.1). In a trivial case that the teacher network and the student network are trained for same task (e^{α} equals e^{β}), then the deep alignment network $\mathcal{F}_{\theta\omega}$ would reduce to an identity mapping, and the problem is also reduced to a single task policy distillation, which has been proved to be

effective in [166]. Before we can apply the deep knowledge distillation, we need to first train a good deep alignment network. In this work, we provide two types of training protocols for different situations:

Offline training: This protocol first trains two teacher networks in both environment ε^{α} and ε^{β}. Then we use the logits of both two teacher networks to train a deep alignment network $\mathcal{F}_{\theta\omega}$. After acquiring a pre-trained $\mathcal{F}_{\theta\omega}$, we train a student network of task e^{β} from scratch, in the meanwhile the teacher network of task e^{α} and $\mathcal{F}_{\theta\omega}$ are used for deep knowledge distillation.

Online training: Suppose we only have a teacher network of task e^{α}, and we want to use the knowledge from task e^{α} to train the student network for task e^{β} to get higher performance from scratch. The pipeline of this method is that, we firstly train the student network by interacting with the environment ε^{β} for a certain amount of steps T_1, and then start to train the alignment network $\mathcal{F}_{\theta\omega}$, using the logits from the teacher network and the student network. Afterwards, at step T_2, we start performing deep knowledge distillation. Obviously T_2 is larger than T_1, and the value of them are task-specific, which is decided empirically in this work.

The offline training could be useful if we have already had a reasonably good model for task e^{β}, while we want to further improve the performance using the knowledge from task e^{α}. The online training method is used when we need to learn the student network from scratch. Both types of training protocol can be extended to multiple heterogeneous tasks.

4.1.7 Collaborative Asynchronous Advantage Actor-Critic

In this section, we introduce the proposed collaborative asynchronous advantage actor-critic (cA3C) algorithm. As we described in section 4.1.5, the agents are running in parallel. Each agent goes through the same training procedure as described in Algorithm 4.1. As it shows, the training of agent g_1 can be separated into two parts: The first part is to interact with the environment, get the reward and compute the gradients to minimize the value loss and policy loss based on Generalized Advantage Estimation (GAE) [169]. The second part is to interact with source agent g_2 so that the logits distilled from agent g_2 can be transferred by the deep alignment network and used as supervision to bias the training of agent g_1.

To be more concrete, the pseudo code in Algorithm 4.1 is an envolved version of A3C based on online training of deep knowledge distillation. At T-th iteration, the agent interacts with the environment for t_{max} steps or until the terminal state is reached (Line 6 to Line 15). Then the updating of value network and policy network is conducted by GAE. This variation of A3C is firstly implemented in OpenAI universe starter agent [147]. Since the main asynchronous framework is the same as A3C, we still use the A3C to denote this algorithm although the updating is the not the same as advantage actor-critic algorithm used in original A3C paper [131].

The online training of deep knowledge distillation is mainly completed from Line 25 to Line 32 in Algorithm 4.1. The training of the deep alignment network starts from T_1 steps (Line 25 - 28). After T_1 steps, the student network is able to generate a representative action probability distribution, and we have suitable supervision to train the deep alignment network as well, parameterized by θ^{ω}. After T_2 steps, θ^{ω} will gradually converge to a local optimal,

and we start the deep knowledge distillation. As illustrated in Figure 4.2 (b), we use symbol $\theta_p^{\beta'}$ to represent the parameters of CNN, LSTM and the fully connected distillation layer, since we don't want the logits from heterogeneous directly affect the action pattern of target task. To simplify the discussion, the above algorithm is described based on interacting with a single agent from a heterogeneous task. In algorithm 4.1, logits $\mathbf{z}_t^{\alpha}$ can be acquired from multiple teacher networks of different tasks, each task will train its own deep alignment network θ^{ω} and distill the aligned logits to the student network.

As we described in previous section 4.1.5, there are two types of interactions in this algorithm: 1). GAE interaction uses the gradients shared by all homogeneous agents. 2) Distillation interaction is the deep knowledge distillation from teacher network. The GAE interaction is performed only among homogeneous tasks. By synchronizing the parameters from a global student network in Algorithm 4.1 (line 3), the current agent receives the GAE updates from all the other agents who interactes with the same environment. In line 21 and 22, the current agent sends his gradients to the global student network, which will be synchronized with other homogeneous agents. The distillation interaction is then conducted in line 31, where we have the aligned logits $\mathcal{F}_{\theta^{\omega}}(\mathbf{z}_t^{\alpha})$ and the distillation logits $\mathbf{z}_t^{\beta'}$ to compute the gradients for minimizing the distillation loss. The gradients of distillation are also sent to the global student network. The role of global student network can be regarded as a parameter server that helps sending interactions among the homogeneous agents. From a different angle, each homogeneous agent maintains an instinct version of global student network. Therefore, both two types of interactions affect all homogeneous agents, which means that the distillation interactions from agent g_2 and agent g_1 would affect all homogeneous agents of agent g_1.

Algorithm 4.1: Online cA3C

Require: Global shared parameter vectors Θ_p and Θ_v and global shared counter $T = 0$; Agent-specific parameter vectors Θ'_p and Θ'_v, GAE [169] parameters γ and λ. Time step to start training deep alignment network and deep knowledge distillation T_1, T_2.

while $T < T_{max}$ **do**
  Reset gradients: $d\theta_p = 0$ and $d\theta_v = 0$
  Synchronize agent-specific parameters $\theta'_p = \theta_p$ and $\theta'_v = \theta_v$
  $t_{start} = t$, Get state s_t
  Receive reward r_t and new state s_{t+1}
  repeat
    Perform a_t according to policy
    Receive reward r_t and new state s_{t+1}
    Compute value of state $v_t = V(s_t; \theta'_v)$
    if $T \geq T_1$ **then**
      Compute the logits $\mathbf{z}_t^{\alpha}$ from teacher network.
      Compute the policy logits $\mathbf{z}_t^{\beta}$ and distillation logits $\mathbf{z}_t^{\beta'}$ from student network.
    end if
    $t = t + 1, T = T + 1$
  until terminal s_t or $t - t_{start} >= t_{max}$

$$R = v_t = \begin{cases} 0 & \text{for terminal } s_t \\ V(s_t, \theta'_v) & \text{for non-terminal } s_t \end{cases}$$

  for $i \in \{t-1, ..., t_{start}\}$ **do**
    $\delta_i = r_i + \gamma v_{i+1} - v_i$
    $A = \delta_i + (\gamma\lambda)A$
    $R = r_i + \gamma R$
    $d\theta_p \leftarrow d\theta_p + \nabla \log \pi(a_i|s_i; \theta')A$
    $d\theta_v \leftarrow d\theta_v + \partial(R - v_i)^2/\partial\theta'_v$
  end for
  Perform asynchronous update of θ_p using $d\theta_p$ and of θ_v using $d\theta_v$.
  if $T \geq T_1$ **then**
    // Training deep alignment network.
    $\min_{\theta^{\omega}} \sum_t l_{KL}(\mathbf{z}_t^{\beta}, \mathbf{z}_t^{\alpha}, \tau)$, l_{KL} is defined in Eq (4.3).
  end if
  if $T \geq T_2$ **then**
    // online deep knowledge distillation.
    $\min_{\theta_p^{\beta'}} \sum_t l_{KL}(\mathcal{F}_{\theta^{\omega}}(\mathbf{z}_t^{\alpha}), \mathbf{z}_t^{\beta'})$
  end if
end while

4.1.8 Experiments

4.1.8.1 Training and Evaluation

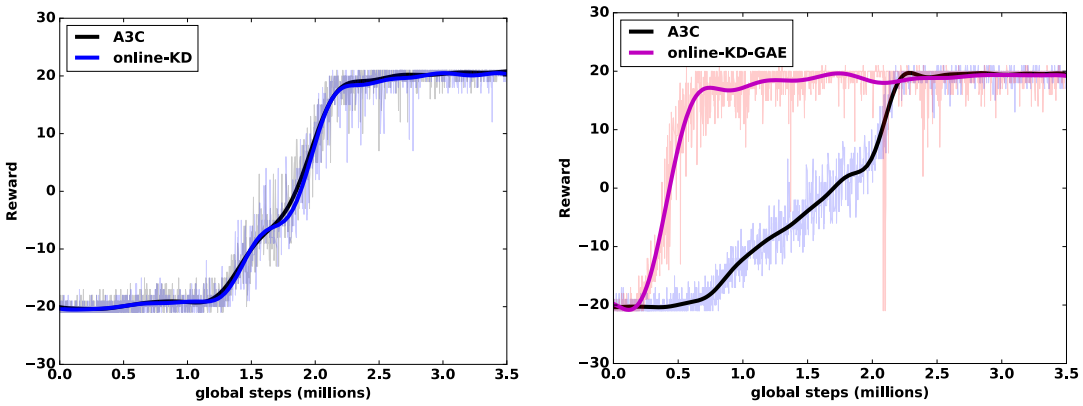
In this work, training and evaluation are conducted in OpenAI Gym [24], a toolkit that includes a collection of benchmark problems such as classic Atari games using Arcade Learning Environment (ALE) [18], classic control games, etc. Same as the standard RL setting, an agent is stimulated in an environment, taking an action and receiving rewards and observations at each time step. The training of the agent is divided into episodes, and the goal is to maximize the expectation of the total reward per episode or to reach higher performance using as few episodes as possible.

4.1.8.2 Certificated Homogeneous transfer

In this subsection, we verify the effectiveness of knowledge distillation as a type of interaction in collaborative deep reinforcement learning for homogeneous tasks. This is also to verify the effectiveness of the simplest case for deep knowledge distillation. Although the effectiveness of policy distillation in deep reinforcement learning has been verified in [166] based on DQN, there is no prior studies on asynchronous online distillation. Therefore, our first experiment is to demonstrate that the knowledge distilled from a certificated task can be used to train a decent student network for a homogeneous task. Otherwise, the even more challenging task of transferring among heterogeneous sources may not work. We note that in this case, the Assumption 3 is fully satisfied given $k_1 = k_2$, where k_1 and k_2 are the knowledge needed to master task e_1 and e_2, respectively. In this experiment, we conduct experiments in a gym environment named PONG. It is a classic Atari game that an agent controls a paddle to bounce a ball pass another player agent. The maximum reward that each episode can reach is 21.

First, we train a teacher network that learns from its own environment by asynchronously performing GAE updates. We then train a student network using only online knowledge distillation from the teacher network. For fair comparisons, we use 8 agents for all environments in the experiments. Specifically, both the student and the teacher are training in PONG with 8 agents. The 8 agents of the teacher network are trained using the A3C algorithm (equivalent to CDRL with GAE updates in one task). The 8 agents of student network are trained using normal policy distillation, which uses the logits generated from the teacher network as supervision to train the policy network directly. From the results in Figure 4.3 (a) we see that the student network can achieve a very competitive performance that is is almost same as the state-of-arts, using online knowledge distillation from a homogeneous task. It also suggests that the teacher doesn't necessarily need to be an expert, before it can guide the training of a student in the homogeneous case. Before 2 million steps, the teacher itself is still learning from the environment, while the knowledge distilled from teacher can already be used to train a reasonable student network. Moreover, we see that the hybrid of two types of interactions in CDRL has a positive effect on the training, instead of causing performance deterioration.

In the second experiment, the student network is learning from both the online knowledge distillation and the GAE updates from the environment. We find that the convergence is much faster than the state-of-art, as shown in Figure 4.3 (b). In this experiment, the knowledge is distilled from the teacher to student in the first one million steps and the distillation is stopped after that. We note that in homogeneous CDRL, knowledge distillation is used directly with policy logits other than distillation logits. The knowledge transfer setting in this experiment is not a practical one because we already have a well-trained model of PONG, but it shows that when knowledge is correctly transferred, the combination of online knowledge



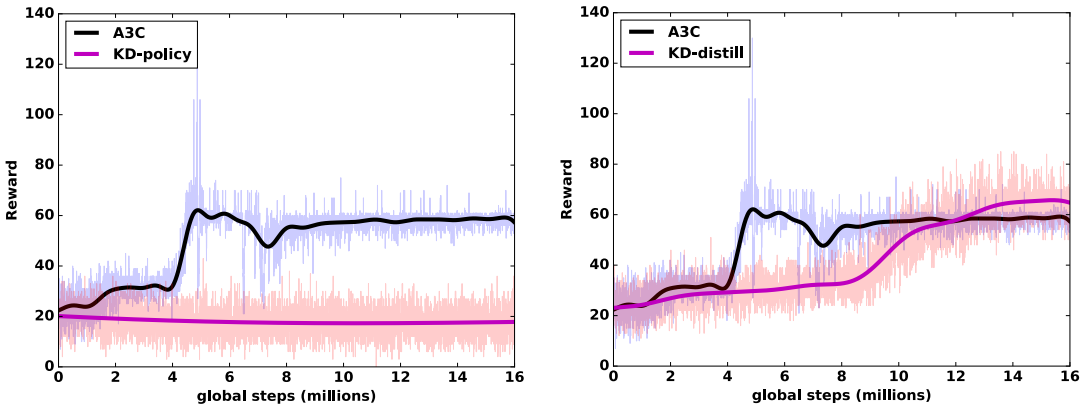
(a) online KD only (b) online KD with GAE

Figure 4.3: Performance of online homogeneous knowledge distillation.

distillation and the GAE updates is an effective training procedure.

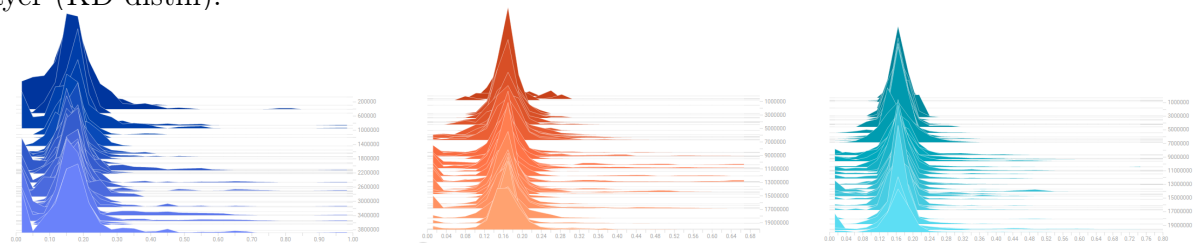
4.1.8.3 Certificated Heterogeneous Transfer

In this subsection, we design experiments to illustrate the effectiveness of CDRL in certificated heterogeneous transfer, with the proposed deep knowledge distillation. Given a certificated task PONG, we want to utilize the existing expertise and apply it to facilitate the training of a new task BOWLING. In the following experiments, we do not tune any model-specific parameters such as number of layers, size of filter or network structure for BOWLING. We first directly perform transfer learning from PONG to BOWLING by knowledge distillation. Since the two tasks has different action patterns and action probability distributions, directly knowledge distillation with a policy layer is not successful, as shown in Figure 4.4 (a). In fact, the knowledge distilled from PONG contradicts to the knowledge learned from BOWLING, which leads to the much worse performance than the baseline. We show in Figure 4.5 (a) and (b) that the action distributions of PONG and BOWLING are very different. To resolve this, we distill the knowledge through an extra distillation layer as illustrated in Figure 4.2 (b). As such, the knowledge distilled from the certificated heterogeneous task can be successfully transferred to the student network with improved performance after the learning is complete.



(a) KD with policy layer (b) KD with distillation layer

Figure 4.4: Performance of online knowledge distillation from a heterogeneous task. (a) distillation from a PONG expert using the policy layer to train a BOWLING student (KD-policy). (b) distillation from a PONG expert to a BOWLING student using an extra distillation layer (KD-distill).

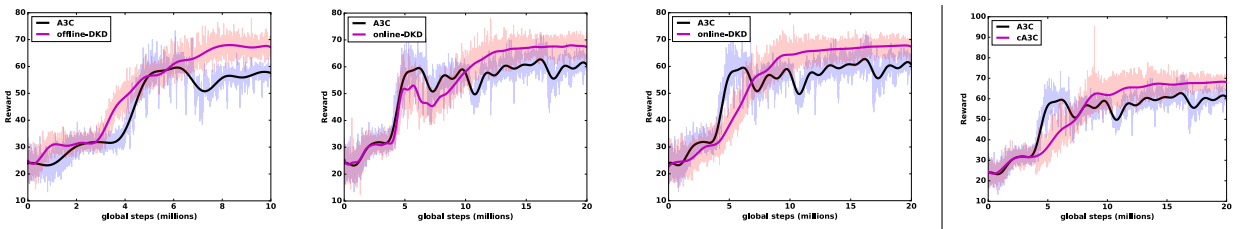


(a) PONG (b) BOWLING (c) aligned PONG

Figure 4.5: The action probability distributions of a PONG expert, a BOWLING expert and an aligned PONG expert.

However, this leads to a much slower convergence than the baseline as shown in Figure 4.4 (b), because that it takes time to learn a good distillation layer to align the knowledge distilled from PONG to the current learning task. An interesting question is that, is it possible to have both improved performance and faster convergence?

Deep knowledge distillation – Offline training. To handle the heterogeneity between PONG and BOWLING, we first verify the effectiveness of deep knowledge distillation with an offline training procedure. The offline training is split into two stages. In the first stage, we train a deep alignment network with four fully connected layers using the Relu activation function. The training data are logits generated from an expert PONG network and BOWLING



(a) Offline (b) Online Strategy 1 (c) Online Strategy 2 (d) Collaborative

Figure 4.6: Performance of **offline**, **online** deep knowledge distillation, and collaborative learning.

network. The rewards of the networks at convergence are 20 and 60 respectively. In stage 2, with the PONG teacher network and trained deep alignment network, we train a BOWLING student network from scratch. The student network is trained with both GAE interactions with its environment, and the distillation interactions from the teacher network and the deep alignment network. The results in Figure 4.6 (a) show that deep knowledge distillation can transfer knowledge from PONG to BOWLING both efficiently and effectively.

Deep knowledge distillation – Online training. A more practical setting of CDRL is the online training, where we simultaneously train deep alignment network and conduct the online deep knowledge distillation. We use two online training strategies: 1) The training of deep alignment network starts after 4 million steps, when the student BOWLING network can perform reasonably well, and the knowledge distillation starts after 6 million steps. 2) The training of deep alignment network starts after 0.1 million steps, and the knowledge distillation starts after 1 million steps. Results are shown in Figure 4.6 (b) and (c) respectively. The results show that both strategies reach higher performance than the baseline. Moreover, the results suggest that we do not have to wait until the student network reaches a reasonable performance before we start to train the deep alignment network. This is because the deep alignment network is train to align two distributions of PONG and BOWLING, instead of transferring the actual knowledge. Recall that the action probability distribution of PONG

and BOWLING are quite different as shown in Figure 4.5 (a) and (b). After we projecting the logits of PONG using the deep alignment network, the distribution is very similar to BOWLING, as shown in Figure 4.5 (c).

4.1.8.4 Collaborative Deep Reinforcement Learning

In previous experiments, we assume that there is a well-trained PONG expert, and we transfer knowledge from the PONG expert to the BOWLING student via deep knowledge distillation. A more challenging settings that both of BOWLING and PONG are trained from scratch. In this experiment, we we show that the CDRL framework can still be effective in this setting. In this experiment, we train a BOWLING network and a PONG network from scratch using the proposed cA3C algorithm. The PONG agents are trained with GAE interactions only, and the target BOWLING receive supervision from both GAE interactions and distilled knowledge from PONG via a deep alignment network. We start to train the deep alignment network after 3 million steps, and perform deep knowledge distillation after 4 million steps, where the PONG agents are still updating from the environment. We note that in this setting, the teacher network is constantly being updated, as knowledge is distilled from the teacher until 15 million steps. Results in Figure 4.6 (d) show that the proposed cA3C is able to converge to a higher performance than the current state-of-art. The reward of last one hundred episodes of A3C is 61.48 ± 1.48, while cA3C achieves 68.35 ± 1.32, with a significant reward improvement of 11.2%.

4.2 Ranking Policy Gradient

4.2.1 Introduction

To utilize the collaborative strategy for improving the sample-efficiency in single agent reinforcement learning, we disentangle the exploration and exploitation into two separate agents and conduct data-driven collaboration through imitation learning, which leads to a more sample-efficient off-policy learning framework. We first approach the sample-efficient reinforcement learning from a ranking perspective. Instead of estimating the optimal action value function, we concentrate on learning optimal rank of actions. The rank of actions depends on the *relative action values*. As long as the relative action values preserve the same rank of actions as the optimal action values (Q-values), we choose the same optimal action. To learn optimal relative action values, we propose the *ranking policy gradient (RPG)* that optimizes the actions' rank with respect to the long-term reward by learning the pairwise relationship among actions.

Ranking Policy Gradient (RPG) that directly optimizes relative action values to maximize the return is a policy gradient method. The track of off-policy actor-critic methods [46, 72, 208] have made substantial progress on improving the sample-efficiency of policy gradient. However, the fundamental difficulty of learning stability associated with the bias-variance trade-off remains [136]. In this work, we first exploit the equivalence between RL optimizing the lower bound of return and supervised learning that imitates a specific optimal policy. Build upon this theoretical foundation, we propose a general off-policy learning framework that equips the generalized policy iteration [187, Chap. 4] with an external step of supervised learning. The proposed off-policy learning not only enjoys the property of optimality preserving (unbiasedness), but also largely reduces the variance of policy gradient because of its

independence of the horizon and reward scale. Furthermore, this learning paradigm leads to a sample complexity analysis of large-scale MDP, in a non-tabular setting without the linear dependence on the state space. Based on our sample-complexity analysis, we define the exploration efficiency that quantitatively evaluates different exploration methods. Besides, we empirically show that there is a trade-off between optimality and sample-efficiency, which is well aligned with our theoretical indication. Last but not least, we demonstrate that the proposed approach, consolidating the RPG with off-policy learning, significantly outperforms the state-of-the-art [80, 17, 42, 132].

4.2.2 Related works

Sample Efficiency. The sample efficient reinforcement learning can be roughly divided into two categories. The first category includes variants of Q-learning [132, 167, 203, 80]. The main advantage of Q-learning methods is the use of off-policy learning, which is essential towards sample efficiency. The representative DQN [132] introduced deep neural network in Q-learning, which further inspried a track of successful DQN variants such as Double DQN [203], Dueling networks [209], prioritized experience replay [167], and RAINBOW [80]. The second category is the actor-critic approaches. Most of recent works [46, 208, 71] in this category leveraged importance sampling by re-weighting the samples to correct the estimation bias and reduce variance. The main advantage is in the wall-clock times due to the distributed framework, firstly presented in [131], instead of the sample-efficiency. As of the time of writing, the variants of DQN [80, 42, 17, 167, 203] are among the algorithms of most sample efficiency, which are adopted as our baselines for comparison.

RL as Supervised Learning. Many efforts have focused on developing the connections between RL and supervised learning, such as Expectation-Maximization algorithms [45, 152,

102, 1], Entropy-Regularized RL [145, 74], and Interactive Imitation Learning (IIL) [44, 188, 163, 165, 184, 81, 148]. EM-based approaches apply the probabilistic framework to formulate the RL problem maximizing a lower bound of the return as a re-weighted regression problem, while it requires on-policy estimation on the expectation step. Entropy-Regularized RL optimizing entropy augmented objectives can lead to off-policy learning without the usage of importance sampling while it converges to soft optimality [74].

Of the three tracks in prior works, the IIL is most closely related to our work. The IIL works firstly pointed out the connection between imitation learning and reinforcement learning [163, 188, 165] and explore the idea of facilitating reinforcement learning by imitating experts. However, most of imitation learning algorithms assume the access to the expert policy or demonstrations. The off-policy learning framework proposed in this thesis can be interpreted as an online imitation learning approach that constructs expert demonstrations during the exploration without soliciting experts, and conducts supervised learning to maximize return at the same time. In short, our approach is different from prior arts in terms of at least one of the following aspects: objectives, oracle assumptions, the optimality of learned policy, and on-policy requirement. More concretely, the proposed method is able to learn optimal policy in terms of long-term reward, without access to the oracle (such as expert policy or expert demonstration) and it can be trained both empirically and theoretically in an off-policy fashion. A more detailed discussion of the related work on reducing RL to supervised learning is provided in Appendix A.

PAC Analysis of RL. Most existing studies on sample complexity analysis [95, 180, 97, 179, 105, 91, 90, 223] are established on the value function estimation. The proposed approach leverages the probably approximately correct framework [202] in a different way such that it does not rely on the value function. Such independence directly leads to a practically

sample-efficient algorithm for large-scale MDP, as we demonstrated in the experiments.

4.2.3 Notations and Problem Setting

Here, we consider a finite horizon T, discrete time Markov Decision Process (MDP) with a finite discrete state space $\mathcal{S}$ and for each state $s \in \mathcal{S}$, the action space $\mathcal{A}_s$ is finite. The environment dynamics is denoted as $\mathbf{P} = \{p(s'|s,a), \forall s, s' \in \mathcal{S}, a \in \mathcal{A}_s\}$. We note that the dimension of action space can vary given different states. We use $m = \max_s \|\mathcal{A}_s\|$ to denote the maximal action dimension among all possible states. Our goal is to maximize the expected sum of positive rewards, or return $J(\theta) = \mathbf{E}_{\tau,\pi_\theta}[\sum_{t=1}^{T} r(s_t, a_t)]$, where $0 < r(s,a) < \infty, \forall s, a$. In this case, the optimal deterministic Markovian policy always exists [156][Proposition 4.4.3]. The upper bound of trajectory reward ($r(\tau)$) is denoted as $R_{\max} = \max_\tau r(\tau)$. A comprehensive list of notations is elaborated in Table 4.1.

4.2.4 Ranking Policy Gradient

Value function estimation is widely used in advanced RL algorithms [132, 131, 170, 71, 80, 42] to facilitate the learning process. In practice, the on-policy requirement of value function estimations in actor-critic methods has largely increased the difficulty of sample-efficient learning [46, 71]. With the advantage of off-policy learning, the DQN [132] variants are currently among the most sample-efficient algorithms [80, 42, 17]. For complicated tasks, the value function can align with the relative relationship of action's return, but the absolute values are hardly accurate [132, 85].

The above observations motivate us to look at the decision phase of RL from a different prospect: Given a state, the decision making is to perform a *relative comparison* over available

Table 4.1: Notations for Section 4.2.

Notations	Definition
λ_{ij}	The discrepancy of the relative action value of action i and action j. $\lambda_{ij} = \lambda_i - \lambda_j$, where $\lambda_i = \lambda(s, a_i)$. Notice that the value here is not the estimation of return, it represents which action will have relatively higher return if followed.
$Q^\pi(s, a)$	The action value function or equivalently the estimation of return taking action a at state s, following policy π.
p_{ij}	$p_{ij} = P(\lambda_i > \lambda_j)$ denotes the probability that i-th action is to be ranked higher than j-th action. Notice that p_{ij} is controlled by θ through λ_i, λ_j
τ	A trajectory $\tau = \{s(\tau, t), a(\tau, t)\}_{t=1}^T$ collected from the environment. It is worth noting that this trajectory is not associated with any policy. It only represents a series of state-action pairs. We also use the abbreviation $s_t = s(\tau, t)$, $a_t = a(\tau, t)$.
$r(\tau)$	The trajectory reward $r(\tau) = \sum_{t=1}^T r(s_t, a_t)$ is the sum of reward along one trajectory.
$R_{\max}$	$R_{\max}$ is the maximal possible trajectory reward, i.e., $R_{\max} = \max_\tau r(\tau)$. Since we focus on MDPs with finite horizon and immediate reward, therefore the trajectory reward is bounded.
$\sum_\tau$	The summation over all possible trajectories τ.
$p(\tau)$	The probability of a specific trajectory is collected from the environment given policy π_θ. $p_\theta(\tau) = p(s_0)\Pi_{t=1}^T \pi_\theta(a_t\|s_t)p(s_{t+1}\|s_t, a_t)$
$\mathcal{T}$	The set of all possible near-optimal trajectories. $\|\mathcal{T}\|$ denotes the number of near-optimal trajectories in $\mathcal{T}$.
n	The number of training samples or equivalently state action pairs sampled from uniformly (near)-optimal policy.
m	The number of discrete actions.

actions and then choose the best action, which can lead to relatively higher return than others. Therefore, an alternative solution is to learn the optimal rank of the actions, instead of deriving policy from the action values. In this section, we show how to optimize the rank of actions to maximize the return, and thus avoid the necessity of accurate estimation for optimal action value function. To learn the rank of actions, we focus on learning *relative*

action value (λ-values), defined as follows:

Definition 4 (Relative action value (λ-values))**.** *For a state s, the relative action values of m actions ($\lambda(s, a_k), k = 1, ..., m$) is a list of scores that denotes the rank of actions. If $\lambda(s, a_i) > \lambda(s, a_j)$, then action a_i is ranked higher than action a_j.*

The optimal relative action values should preserve the same optimal action as the optimal action values:

$$\arg\max_a \lambda(s, a) = \arg\max_a Q^{\pi *}(s, a)$$

where $Q^{\pi *}(s, a_i)$ and $\lambda(s, a_i)$ represent the optimal action value and the relative action value of action a_i, respectively. We omit the model parameter θ in $\lambda_\theta(s, a_i)$ for concise presentation.

Remark 1. *The λ-values are different from the advantage function $A^\pi(s, a) = Q^\pi(s, a) - V^\pi(s)$. The advantage functions quantitatively show the difference of return taking different actions following the current policy π. The λ-values only determine the relative order of actions and its magnitudes are not the estimations of returns.*

To learn the λ-values, we can construct a probabilistic model of λ-values such that the best action has the highest probability to be selected than others. Inspired by learning to rank [26], we consider the pairwise relationship among all actions, by modeling the probability (denoted as p_{ij}) of an action a_i to be ranked higher than any action a_j as follows:

$$p_{ij} = \frac{\exp(\lambda(s, a_i) - \lambda(s, a_j))}{1 + \exp(\lambda(s, a_i) - \lambda(s, a_j))}, \tag{4.4}$$

where $p_{ij} = 0.5$ means the relative action value of a_i is same as that of the action a_j, $p_{ij} > 0.5$ indicates that the action a_i is ranked higher than a_j. Given the independent Assumption 4, we can represent the probability of selecting one action as the multiplication of a set of

pairwise probabilities in Eq (4.4). Formally, we define the pairwise ranking policy in Eq (4.5). Please refer to Section A in the Appendix for the discussions on feasibility of Assumption 4.

Definition 5. *The pairwise ranking policy is defined as:*

$$\pi(a = a_i|s) = \Pi_{j=1,j\neq i}^{m}\ p_{ij}, \tag{4.5}$$

where the p_{ij} is defined in Eq (4.4). The probability depends on the relative action values $q = [\lambda_1, ..., \lambda_m]$. The highest relative action value leads to the highest probability to be selected.

Assumption 4. *For a state s, the set of events $E = \{e_{ij}|\forall i \neq j\}$ are conditionally independent, where e_{ij} denotes the event that action a_i is ranked higher than action a_j. The independence of the events is conditioned on a MDP and a stationary policy.*

Our ultimate goal is to maximize the long-term reward through optimizing the pairwise ranking policy or equivalently optimizing pairwise relationship among the action pairs. Ideally, we would like the pairwise ranking policy selects the best action with the highest probability and the highest λ-value. To achieve this goal, we resort to the policy gradient method. Formally, we propose the ranking policy gradient method (RPG), as shown in Theorem 2.

Theorem 2 (Ranking Policy Gradient Theorem)**.** *For any MDP, the gradient of the expected long-term reward $J(\theta) = \sum_\tau p_\theta(\tau)r(\tau)$ w.r.t. the parameter θ of a pairwise ranking policy (Def 5) can be approximated by:*

$$\nabla_\theta J(\theta) \approx \mathbf{E}_{\tau\sim\pi_\theta}\left[\sum\nolimits_{t=1}^{T} \nabla_\theta \left(\sum\nolimits_{j=1,j\neq i}^{m} (\lambda_i - \lambda_j)/2\right) r(\tau)\right], \tag{4.6}$$

and the deterministic pairwise ranking policy π_θ is: $a = \arg\max_i \lambda_i,\ i = 1, \ldots, m$, where

λ_i denotes the relative action value of action a_i ($\lambda_\theta(s_t, a_t)$, $a_i = a_t$), s_t and a_t denotes the t-th state-action pair in trajectory τ, $\lambda_j, \forall j \neq i$ denote the relative action values of all other actions that were not taken given state s_t in trajectory τ, i.e., $\lambda_\theta(s_t, a_j)$, $\forall a_j \neq a_t$.

The proof of Theorem 2 is provided in Appendix A. Theorem 2 states that optimizing the discrepancy between the action values of the best action and all other actions, is optimizing the pairwise relationships that maximize the return. One limitation of RPG is that it is not convenient for the tasks where only optimal stochastic policies exist since the pairwise ranking policy takes extra efforts to construct a probability distribution [see Appendix A]. In order to learn the stochastic policy, we introduce Listwise Policy Gradient (LPG) that optimizes the probability of ranking a specific action on the top of a set of actions, with respect to the return. In the context of RL, this top one probability is the probability of action a_i to be chosen, which is equal to the sum of probability all possible permutations that map action a_i at the top. This probability is computationally prohibitive since we need to consider the probability of $m!$ permutations. Inspired by listwise learning to rank approach [31], the top one probability can be modeled by the softmax function (see Theorem 3). Therefore, LPG is equivalent to the REINFORCE [212] algorithm with a softmax layer. LPG provides another interpretation of REINFORCE algorithm from the perspective of learning the optimal ranking and enables the learning of both deterministic policy and stochastic policy (see Theorem 4).

Theorem 3 ([31], Theorem 6)**.** *Given the action values $q = [\lambda_1, ..., \lambda_m]$, the probability of action i to be chosen (i.e. to be ranked on the top of the list) is:*

$$\pi(a_t = a_i | s_t) = \frac{\phi(\lambda_i)}{\sum_{j=1}^{m} \phi(\lambda_j)}, \tag{4.7}$$

where $\phi(*)$ is any increasing, strictly positive function. A common choice of ϕ is the

exponential function.

Theorem 4 (Listwise Policy Gradient Theorem)**.** *For any MDP, the gradient of the long-term reward* $J(\theta) = \sum_\tau p_\theta(\tau) r(\tau)$ *w.r.t. the parameter* θ *of listwise ranking policy takes the following form:*

$$\nabla_\theta J(\theta) = \mathbf{E}_{\tau \sim \pi_\theta} \left[\sum_{t=1}^{T} \nabla_\theta \left(\log \frac{e^{\lambda_i}}{\sum_{j=1}^{m} e^{\lambda_j}} \right) r(\tau) \right], \tag{4.8}$$

where the listwise ranking policy π_θ *parameterized by* θ *is given by Eq (4.9) for tasks with deterministic optimal policies:*

$$a = \arg\max_i \lambda_i, \quad i = 1, \ldots, m \tag{4.9}$$

or Eq (4.10) for stochastic optimal policies:

$$a \sim \pi(*|s), \quad i = 1, \ldots, m \tag{4.10}$$

where the policy takes the form as in Eq (4.11)

$$\pi(a = a_i | s_t) = \frac{e^{\lambda_i}}{\sum_{j=1}^{m} e^{\lambda_j}} \tag{4.11}$$

is the probability that action i *being ranked highest, given the current state and all the relative action values* $\lambda_1 \ldots \lambda_m$*.*

The proof of Theorem 4 exactly follows the direct policy differentiation [153, 212] by replacing the policy to the form of the Softmax function. The action probability $\pi(a_i|s), \forall i = 1, ..., m$ forms a probability distribution over the set of discrete actions [31, Lemma 7].

Theorem 4 states that the vanilla policy gradient [212] parameterized by Softmax layer is optimizing the probability of each action to be ranked highest, with respect to the long-term reward. Furthermore, it enables learning both of the deterministic policy and stochastic policy.

To this end, seeking sample-efficiency motivates us to learn the relative relationship (RPG (Theorem 2) and LPG (Theorem 4)) of actions, instead of deriving policy based on action value estimations. However, both of the RPG and LPG belong to policy gradient methods, which suffers from large variance and the on-policy learning requirement [187]. Therefore, the intuitive implementations of RPG or LPG are still far from sample-efficient. In the next section, we will describe a general off-policy learning framework empowered by supervised learning, which provides an alternative way to accelerate learning, preserve optimality, and reduce variance.

4.2.5 Off-policy Learning as Supervised Learning

In this section, we discuss the connections and discrepancies between RL and supervised learning, and our results lead to a sample-efficient off-policy learning paradigm for RL. The main result in this section is Theorem 5, which casts the problem of maximizing the lower bound of return into a supervised learning problem, given one relatively mild Assumption 5 and practical assumptions 4,6. It can be shown that these assumptions are valid in a range of common RL tasks, as discussed in Lemma 6 in Appendix A. The central idea is to collect only the near-optimal trajectories when the learning agent interacts with the environment, and imitate the near-optimal policy by maximizing the log likelihood of the state-action pairs from these near-optimal trajectories. With the road map in mind, we then begin to introduce our approach as follows.

In a discrete action MDP with finite states and horizon, given the near-optimal policy π_*, the stationary state distribution is given by: $p_{\pi_*}(s) = \sum_\tau p(s|\tau)p_{\pi_*}(\tau)$, where $p(s|\tau)$ is the probability of a certain state given a specific trajectory τ and is not associated with any policies, and only $p_{\pi_*}(\tau)$ is related to the policy parameters. The stationary distribution of state-action pairs is thus: $p_{\pi_*}(s,a) = p_{\pi_*}(s)\pi_*(a|s)$. In this section, we consider the MDP that each initial state will lead to at least one (near)-optimal trajectory. For a more general case, please refer to the discussion in Appendix A. In order to connect supervised learning (i.e., imitating a near-optimal policy) with RL and enable sample-efficient off-policy learning, we first introduce the trajectory reward shaping (TRS), defined as follows:

Definition 6 (Trajectory Reward Shaping, TRS)**.** *Given a fixed trajectory τ, its trajectory reward is shaped as follows:*

$$w(\tau) = \begin{cases} 1, & \text{if } r(\tau) \geq c \\ 0, & o.w. \end{cases}$$

where $c = R_{\max} - \epsilon$ is a problem-dependent near-optimal trajectory reward threshold that indicates the least reward of near-optimal trajectory, $\epsilon \geq 0$ and $\epsilon \ll R_{\max}$. We denote the set of all possible near-optimal trajectories as $\mathcal{T} = \{\tau | w(\tau) = 1\}$, i.e., $w(\tau) = 1, \forall \tau \in \mathcal{T}$.

Remark 2. *The threshold c indicates a trade-off between the sample-efficiency and the optimality. The higher the threshold, the less frequently it will hit the near-optimal trajectories during exploration, which means it has higher sample complexity, while the final performance is better (see Figure 4.10).*

Remark 3. *The trajectory reward can be reshaped to any positive functions that are not related to policy parameter θ. For example, if we set $w(\tau) = r(\tau)$, the conclusions in this section still hold (see Eq (A.6) in Appendix A). For the sake of simplicity, we set $w(\tau) = 1$.*

Different from the reward shaping work [139], where shaping happens at each step on $r(s_t, a_t)$, the proposed approach directly shapes the trajectory reward $r(\tau)$, which facilitates the smooth transform from RL to SL. After shaping the trajectory reward, we can transfer the goal of RL from maximizing the return to maximize the long-term performance (Def 7).

Definition 7 (Long-term Performance)**.** *The long-term performance is defined by the expected shaped trajectory reward:*

$$\sum\nolimits_{\tau} p_\theta(\tau) w(\tau). \tag{4.12}$$

According to Def 6, the expectation over all trajectories is the equal to that over the near-optimal trajectories in $\mathcal{T}$, i.e., $\sum_\tau p_\theta(\tau)w(\tau) = \sum_{\tau\in\mathcal{T}} p_\theta(\tau)w(\tau)$.

The optimality is preserved after trajectory reward shaping ($\epsilon = 0, c = R_{\max}$) since the optimal policy π_* maximizing long-term performance is also an optimal policy for the original MDP, i.e., $\sum_\tau p_{\pi_*}(\tau)r(\tau) = \sum_{\tau\in\mathcal{T}} p_{\pi_*}(\tau)r(\tau) = R_{\max}$, where $\pi_* = \arg\max_{\pi_\theta} \sum_\tau p_{\pi_\theta}(\tau)w(\tau)$ and $p_{\pi_*}(\tau) = 0, \forall\tau \notin \mathcal{T}$ (see Lemma 4 in Appendix A). Similarly, when $\epsilon > 0$, the optimal policy after trajectory reward shaping is a near-optimal policy for original MDP. Note that most policy gradient methods use the softmax function, in which we have $\exists\tau \notin \mathcal{T}, p_{\pi_\theta}(\tau) > 0$ (see Lemma 5 in Appendix A). Therefore when softmax is used to model a policy, it will not converge to an exact optimal policy. On the other hand, ideally, the discrepancy of the performance between them can be arbitrarily small based on the universal approximation [83] with general conditions on the activation function and Theorem 1 in [188].

Essentially, we use TRS to filter out near-optimal trajectories and then we maximize the probabilities of near-optimal trajectories to maximize the long-term performance. This procedure can be approximated by maximizing the log-likelihood of near-optimal state-action

pairs, which is a supervised learning problem. Before we state our main results, we first introduce the definition of uniformly near-optimal policy (Def 8) and a prerequisite (Asm. 5) specifying the applicability of the results.

Definition 8 (Uniformly Near-Optimal Policy, UNOP)**.** *The Uniformly Near-Optimal Policy π_* is the policy whose probability distribution over near-optimal trajectories ($\mathcal{T}$) is a uniform distribution. i.e. $p_{\pi_*}(\tau) = \frac{1}{|\mathcal{T}|}, \forall \tau \in \mathcal{T}$, where $|\mathcal{T}|$ is the number of near-optimal trajectories. When we set $c = R_{\max}$, it is an optimal policy in terms of both maximizing return and long-term performance. In the case of $c = R_{\max}$, the corresponding uniform policy is an optimal policy, we denote this type of optimal policy as uniformly optimal policy (UOP).*

Assumption 5 (Existence of Uniformly Near-Optimal Policy)**.** *We assume the existence of Uniformly Near-Optimal Policy (Def. 8).*

Based on Lemma 6 in Appendix A, Assumption 5 is satisfied for certain MDPs that have deterministic dynamics. Other than Assumption 5, all other assumptions in this work (Assumptions 4,6) can almost always be satisfied in practice, based on empirical observations. With these relatively mild assumptions, we present the following long-term performance theorem, which shows the close connection between supervised learning and RL.

Theorem 5 (Long-term Performance Theorem)**.** *Maximizing the lower bound of expected long-term performance in Eq (4.12) is maximizing the log-likelihood of state-action pairs sampled from a uniformly (near)-optimal policy π_*, which is a supervised learning problem:*

$$\arg\max_{\theta} \sum\nolimits_{s\in\mathcal{S}} \sum\nolimits_{a\in\mathcal{A}_s} p_{\pi_*}(s,a) \log \pi_\theta(a|s) \tag{4.13}$$

The optimal policy of maximizing the lower bound is also the optimal policy of maximizing

the long-term performance and the return.

Remark 4. *It is worth noting that Theorem 5 does not require a uniformly near-optimal policy π_* to be deterministic. The only requirement is the existence of a uniformly near-optimal policy.*

Remark 5. *Maximizing the lower bound of long-term performance is maximizing the lower bound of long-term reward since we can set $w(\tau) = r(\tau)$ and $\sum_\tau p_\theta(\tau) r(\tau) \geq \sum_{\mathcal{T}} p_\theta(\tau) w(\tau)$. An optimal policy that maximizes this lower bound is also an optimal policy maximizing the long-term performance when $c = R_{\max}$, thus maximizing the return.*

The proof of Theorem 5 can be found in Appendix A. Theorem 5 indicates that we break the dependency between current policy π_θ and the environment dynamics, which means off-policy learning is able to be conducted by the above supervised learning approach. Furthermore, we point out that there is a potential discrepancy between imitating UNOP by maximizing log likelihood (even when the optimal policy's samples are given) and the reinforcement learning since we are maximizing a lower bound of expected long-term performance (or equivalently the return over the near-optimal trajectories only) instead of return over all trajectories. In practice, the state-action pairs from an optimal policy is hard to construct while the uniform characteristic of UNOP can alleviate this issue (see Sec 4.2.6). Towards sample-efficient RL, we apply Theorem 5 to RPG, which reduces the ranking policy gradient to a classification problem by Corollary 1.

Corollary 1 (Ranking performance policy gradient)**.** *The lower bound of expected long-term performance (defined in Eq (4.12)) using pairwise ranking policy (Eq (4.5)) can be*

approximately optimized by the following loss:

$$\min_{\theta} \sum\nolimits_{s,a_i} p_{\pi*}(s,a_i) \left(\sum\nolimits_{j=1,j\neq i}^{m} \max(0, 1 + \lambda(s,a_j) - \lambda(s,a_i)) \right). \tag{4.14}$$

Corollary 2 (Listwise performance policy gradient)**.** *Optimizing the lower bound of expected long-term performance by the listwise ranking policy (Eq (4.11)) is equivalent to:*

$$\max_{\theta} \sum\nolimits_{s} p_{\pi*}(s) \sum\nolimits_{i=1}^{m} \pi_*(a_i|s) \log \frac{e^{\lambda_i}}{\sum_{j=1}^{m} e^{\lambda_j}} \tag{4.15}$$

The proof of this Corollary is a direct application of theorem 5 by replacing policy with the softmax function.

The proof of Corollary 1 can be found in Appendix A. Similarly, we can reduce LPG to a classification problem (see Corollary 2). One advantage of casting RL to SL is variance reduction. With the proposed off-policy supervised learning, we can reduce the upper bound of the policy gradient variance, as shown in the Corollary 3. Before introducing the variance reduction results, we first make the common assumptions on the MDP regularity (Assumption 6) similar to [43, 46, A1]. Furthermore, the Assumption 6 is guaranteed for bounded continuously differentiable policy such as softmax function.

Assumption 6. *we assume the existence of maximum norm of log gradient over all possible state-action pairs, i.e.*

$$C = \max_{s,a} \|\nabla_\theta \log \pi_\theta(a|s)\|_\infty$$

Corollary 3 (Policy gradient variance reduction)**.** *Given a stationary policy, the upper bound of the variance of each dimension of policy gradient is* $\mathcal{O}(T^2C^2R_{\max}^2)$*. The upper bound of gradient variance of maximizing the lower bound of long-term performance Eq (4.13) is* $\mathcal{O}(C^2)$*, where* C *is the maximum norm of log gradient based on Assumption 6. The supervised learning has reduced the upper bound of gradient variance by an order of* $\mathcal{O}(T^2R_{\max}^2)$ *as compared to the regular policy gradient, considering* $R_{\max} \geq 1, T \geq 1$*, which is a very common situation in practice.*

The proof of Corollary 3 can be found in Appendix A. This corollary shows that the variance of regular policy gradient is upper-bounded by the square of time horizon and the maximum trajectory reward. It is aligned with our intuition and empirical observation: the longer the horizon the harder the learning. Also, the common reward shaping tricks such as truncating the reward to $[-1, 1]$ [34] can help the learning since it reduces variance by decreasing $R_{\max}$. With supervised learning, we concentrate the difficulty of long-time horizon into the exploration phase, which is an inevitable issue for all RL algorithms, and we drop the dependence on T and $R_{\max}$ for policy variance. Thus, it is more stable and efficient to train the policy using supervised learning. One potential limitation of this method is that the trajectory reward threshold c is task-specific, which is crucial to the final performance and sample-efficiency. In many applications such as Dialogue system [111], recommender system [130], etc., we design the reward function to guide the learning process, in which c is naturally known. For the cases that we have no prior knowledge on the reward function of MDP, we treat c as a tuning parameter to balance the optimality and efficiency, as we empirically verified in Figure 4.10. The major theoretical uncertainty on general tasks is the existence of a uniformly near-optimal policy, which is negligible to the empirical performance. The rigorous theoretical analysis of this problem is beyond the scope of this work.

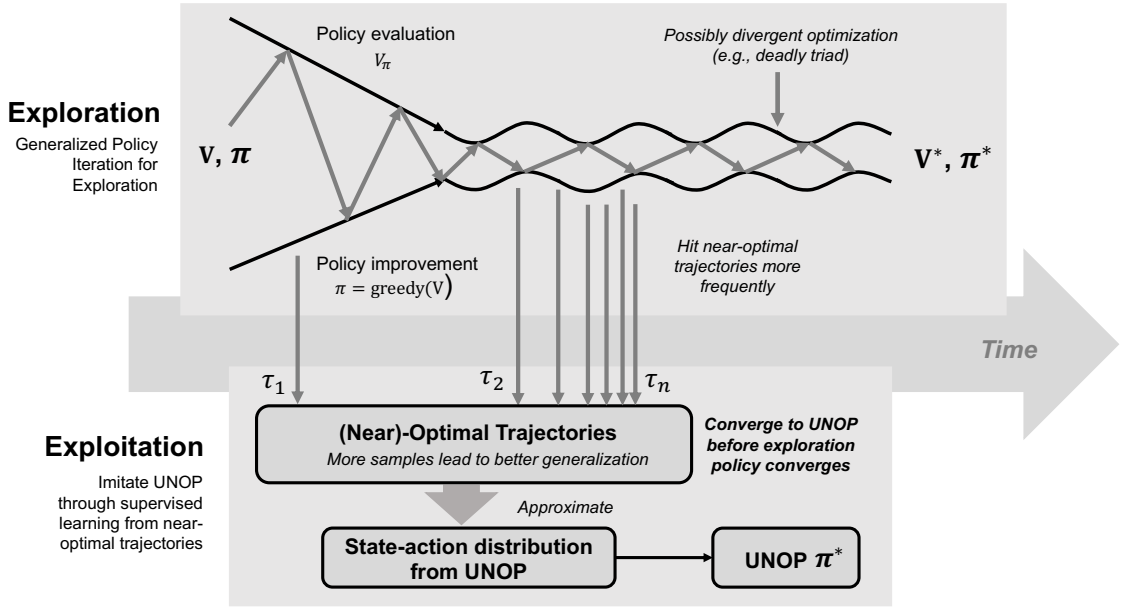


Figure 4.7: Off-policy learning framework.

4.2.6 An algorithmic framework for off-policy learning

Based on the discussions in Section 4.2.5, we exploit the advantage of reducing RL into supervised learning via a proposed two-stages off-policy learning framework. As we illustrated in Figure 4.7, the proposed framework contains the following two stages:

Generalized Policy Iteration for Exploration. The goal of the exploration stage is to collect different near-optimal trajectories as frequently as possible. Under the off-policy framework, the exploration agent and the learning agent can be separated. Therefore, any existing RL algorithm can be used during the exploration. The principle of this framework is using the most advanced RL agents as an exploration strategy in order to collect more near-optimal trajectories and leave the policy learning to the supervision stage.

Supervision. In this stage, we imitate the uniformly near-optimal policy, UNOP (Def 8). Although we have no access to the UNOP, we can approximate the state-action distribution from UNOP by collecting the near-optimal trajectories only. The near-optimal samples

are constructed online and we are not given any expert demonstration or expert policy beforehand. This step provides a sample-efficient approach to conduct exploitation, which enjoys the superiority of stability (Figure 4.9), variance reduction (Corollary 3), and optimality preserving (Theorem 5).

The two-stage algorithmic framework can be directly incorporated in RPG and LPG to improve sample efficiency. The implementation of RPG is given in Algorithm 4.2, and LPG follows the same procedure except for the difference in the loss function. The main requirement of Alg. 4.2 is on the exploration efficiency and the MDP structure. During the exploration stage, a sufficient amount of the different near-optimal trajectories need to be collected for constructing a representative supervised learning training dataset. Theoretically, this requirement always holds [see Appendix Section A, Lemma 7], while the number of episodes explored could be prohibitively large, which makes this algorithm sample-inefficient. This could be a practical concern of the proposed algorithm. However, according to our extensive empirical observations, we notice that long before the value function based state-of-the-art converges to near-optimal performance, enough amount of near-optimal trajectories are already explored.

Therefore, we point out that instead of estimating optimal action value functions and then choosing action greedily, using value function to facilitate the exploration and imitating UNOP is a more sample-efficient approach. As illustrated in Figure 4.7, value based methods with off-policy learning, bootstrapping, and function approximation could lead to a divergent optimization [187, Chap. 11]. In contrast to resolving the instability, we circumvent this issue via constructing a stationary target using the samples from (near)-optimal trajectories, and perform imitation learning. This two-stage approach can avoid the extensive exploration of the suboptimal state-action space and reduce the substantial number of samples needed

for estimating optimal action values. In the MDP where we have a high probability of hitting the near-optimal trajectories (such as PONG), the supervision stage can further facilitate the exploration. It should be emphasized that our work focuses on improving the sample-efficiency through more effective exploitation, rather than developing novel exploration method.

Algorithm 4.2: Off-Policy Learning for Ranking Policy Gradient (RPG)

Require: The near-optimal trajectory reward threshold c, the number of maximal training episodes N_{max}. Maximum number of time steps in each episode T, and batch size b.

while episode $< N_{\max}$ **do**
  repeat
    Retrieve state s_t and sample action a_t by the specified exploration agent (random, ϵ-greedy, or any RL algorithms).
    Collect the experience $e_t = (s_t, a_t, r_t, s_{t+1})$ and store to the replay buffer.
    $t = t + 1$
    if t % update step $== 0$ **then**
      Sample a batch of experience $\{e_j\}_{j=1}^{b}$ from the near-optimal replay buffer.
      Update π_θ based on the hinge loss Eq (4.14) for RPG.
      Update the exploration agent using samples from the regular replay buffer (In simple MDPs such as PONG where near-optimal trajectories are encountered frequently, near-optimal replay buffer can be used to update the exploration agent).
    end if
  until terminal s_t or $t - t_{\text{start}} >= T$
  if return $\sum_{t=1}^{T} r_t \geq c$ **then**
    Take the near-optimal trajectory $e_t, t = 1, ..., T$ in the latest episode from the regular replay buffer, and insert the trajectory into the near-optimal replay buffer.
  end if
  if t % evaluation step $== 0$ **then**
    Evaluate the RPG agent by greedily choosing the action. If the best performance is reached, then stop training.
  end if
end while

4.2.7 Sample Complexity and Generalization Performance

In this section, we present a theoretical analysis on the sample complexity of RPG with off-policy learning framework in Section 4.2.6. The analysis leverages the results from the Probably Approximately Correct (PAC) framework, and provides an alternative approach

to quantify sample complexity of RL from the perspective of the connection between RL and SL (see Theorem 5), which is significantly different from the existing approaches that use value function estimations [95, 180, 97, 179, 105, 91, 90, 223]. We show that the sample complexity of RPG (Theorem 6) depends on the properties of MDP such as horizon, action space, dynamics, and the generalization performance of supervised learning. It is worth mentioning that the sample complexity of RPG has no linear dependence on the state-space, which makes it suitable for large-scale MDPs. Moreover, we also provide a formal quantitative definition (Def 9) on the exploration efficiency of RL.

Corresponding to the two-stage framework in Section 4.2.6, the sample complexity of RPG also splits into two problems:

- **Learning efficiency:** How many state-action pairs from the uniformly optimal policy do we need to collect, in order to achieve good generalization performance in RL?
- **Exploration efficiency:** For a certain type of MDPs, what is the probability of collecting n training samples (state-action pairs from the uniformly near-optimal policy) in the first k episodes in the worst case? This question leads to a quantitative evaluation metric of different exploration methods.

The first stage is resolved by Theorem 6, which connects the lower bound of the generalization performance of RL to the supervised learning generalization performance. Then we discuss the exploration efficiency of the worst case performance for a binary tree MDP in Lemma 2. Jointly, we show how to link the two stages to give a general theorem that studies how many samples we need to collect in order to achieve certain performance in RL.

In this section, we restrict our discussion on the MDPs with a fixed action space and assume the existence of deterministic optimal policy. The policy $\pi = \hat{h} = \arg\min_{h \in \mathcal{H}} \hat{\epsilon}(h)$

corresponds to the empirical risk minimizer (ERM) in the learning theory literature, which is the policy we obtained through learning on the training samples. $\mathcal{H}$ denotes the hypothesis class from where we are selecting the policy. Given a hypothesis (policy) h, the empirical risk is given by $\hat{\epsilon}(h) = \sum_{i=1}^{n} \frac{1}{n} \mathbf{1}\{h(s_i) \neq a_i\}$. Without loss of generosity, we can normalize the reward function to set the upper bound of trajectory reward equals to one (*i.e.*, $R_{\max} = 1$), similar to the assumption in [90]. It is worth noting that the training samples are generated *i.i.d.* from an unknown distribution, which is perhaps the most important assumption in the statistical learning theory. *i.i.d.* is satisfied in this case since the state action pairs (training samples) are collected by filtering the samples during the learning stage, and we can manually manipulate the samples to follow the distribution of UOP (Def 8) by only storing the unique near-optimal trajectories.

4.2.8 Supervision stage: Learning efficiency

To simplify the presentation, we restrict our discussion on the finite hypothesis class (i.e. $|\mathcal{H}| < \infty$) since this dependence is not germane to our discussion. However, we note that the theoretical framework in this section is not limited to the finite hypothesis class. For example, we can simply use the VC dimension [204] or the Rademacher complexity [15] to generalize our discussion to the infinite hypothesis class, such as neural networks. For completeness, we first revisit the sample complexity result from the PAC learning in the context of RL.

Lemma 1 (Supervised Learning Sample Complexity [133])**.** *Let $|\mathcal{H}| < \infty$, and let δ, γ be fixed, the inequality $\epsilon(\hat{h}) \leq (\min_{h \in \mathcal{H}} \epsilon(h)) + 2\gamma = \eta$ holds with probability at least $1 - \delta$, when*

the training set size n satisfies:

$$n \geq \frac{1}{2\gamma^2} \log \frac{2|\mathcal{H}|}{\delta}, \tag{4.16}$$

where the generalization error (expected risk) of a hypothesis $\hat{h}$ is defined as:

$$\epsilon(\hat{h}) = \sum\nolimits_{s,a} p_{\pi_*}(s,a) \mathbf{1}\left\{\hat{h}(s) \neq a\right\}.$$

Condition 1 (Action values)**.** *We restrict the action values of RPG in certain range, i.e., $\lambda_i \in [0, c_q]$, where c_q is a positive constant.*

This condition can be easily satisfied, for example, we can use a sigmoid to cast the action values into $[0, 1]$. We can impose this constraint since in RPG we only focus on the relative relationship of action values. Given the mild condition and established on the prior work in statistical learning theory, we introduce the following results that connect the supervised learning and reinforcement learning.

Theorem 6 (Generalization Performance)**.** *Given a MDP where the UOP (Def 8) is deterministic, let $|\mathcal{H}|$ denote the size of hypothesis space, and δ, n be fixed, the following inequality holds with probability at least $1 - \delta$:*

$$\sum\nolimits_{\tau} p_\theta(\tau) r(\tau) \geq D(1+e)^{\eta(1-m)T},$$

where $D = |\mathcal{T}| \left(\Pi_{\tau \in \mathcal{T}} p_d(\tau)\right)^{\frac{1}{|\mathcal{T}|}}$, $p_d(\tau) = p(s_1)\Pi_{t=1}^{T} p(s_{t+1}|s_t, a_t)$ denotes the environment dynamics. η is the upper bound of supervised learning generalization performance, defined as $\eta = (\min_{h \in \mathcal{H}} \epsilon(h)) + 2\sqrt{\frac{1}{2n} \log \frac{2|\mathcal{H}|}{\delta}} = 2\sqrt{\frac{1}{2n} \log \frac{2|\mathcal{H}|}{\delta}}$.

Corollary 4 (Sample Complexity)**.** *Given a MDP where the UOP (Def 8) is deterministic, let $|\mathcal{H}|$ denotes the size of hypothesis space, and let δ be fixed. Then for the following inequality to hold with probability at least $1-\delta$:*

$$\sum\nolimits_{\tau} p_\theta(\tau) r(\tau) \geq 1-\epsilon,$$

it suffices that the number of state action pairs (training sample size n) from the uniformly optimal policy satisfies:

$$n \geq \frac{2(m-1)^2 T^2}{(\log_{1+e} \frac{D}{1-\epsilon})^2} \log \frac{2|\mathcal{H}|}{\delta} = \mathcal{O}\left(\frac{m^2 T^2}{\left(\log \frac{D}{1-\epsilon}\right)^2} \log \frac{|\mathcal{H}|}{\delta} \right).$$

The proofs of Theorem 6 and Corollary 4 are provided in Appendix A. Theorem 6 establishes the connection between the generalization performance of RL and the sample complexity of supervised learning. The lower bound of generalization performance decreases exponentially with respect to the horizon T and action space dimension m. This is aligned with our empirical observation that it is more difficult to learn the MDPs with a longer horizon and/or a larger action space. Furthermore, the generalization performance has a linear dependence on D, the transition probability of optimal trajectories. Therefore, T, m, and D jointly determines the difficulty of learning of the given MDP. As pointed out by Corollary 4, the smaller the D is, the higher the sample complexity. Note that T, m, and D all characterize intrinsic properties of MDPs, which cannot be improved by our learning algorithms. One advantage of RPG is that its sample complexity has no dependence on the state space, which enables the RPG to resolve large-scale complicated MDPs, as

demonstrated in our experiments. In the supervision stage, our goal is the same as in the traditional supervised learning: to achieve better generalization performance η.

4.2.9 Exploration stage: Exploration efficiency

The exploration efficiency is highly related to the MDP properties and the exploration strategy. To provide interpretation on how the MDP properties (state space dimension, action space dimension, horizon) affect the sample complexity through exploration efficiency, we characterize a simplified MDP as in [184] , in which we explicitly compute the exploration efficiency of a stationary policy (random exploration), as shown in Figure 4.8.

Definition 9 (Exploration Efficiency)**.** *We define the exploration efficiency of a certain exploration algorithm (A) within a MDP ($\mathcal{M}$) as the probability of sampling i distinct optimal trajectories in the first k episodes. We denote the exploration efficiency as $p_{A,\mathcal{M}}(n_{traj} \geq i|k)$. When $\mathcal{M}$, k, i and optimality threshold c are fixed, the higher the $p_{A,\mathcal{M}}(n_{traj} \geq i|k)$, the better the exploration efficiency. We use n_{traj} to denote the number of near-optimal trajectories in this subsection. If the exploration algorithm derives a series of learning policies, then we have $p_{A,\mathcal{M}}(n_{traj} \geq i|k) = p_{\{\pi_i\}_{i=0}^{t},\mathcal{M}}(n_{traj} \geq i|k)$, where t is the number of steps the algorithm A updated the policy. If we would like to study the exploration efficiency of a stationary policy, then we have $p_{A,\mathcal{M}}(n_{traj} \geq i|k) = p_{\pi,\mathcal{M}}(n_{traj} \geq i|k)$.*

Definition 10 (Expected Exploration Efficiency)**.** *The expected exploration efficiency of a certain exploration algorithm (A) within a MDP ($\mathcal{M}$) is defined as:*

$$E_{A,k,\mathcal{M}} = \sum_{i=0}^{k} p_{A,\mathcal{M}}(n_{traj} = i|k)i.$$

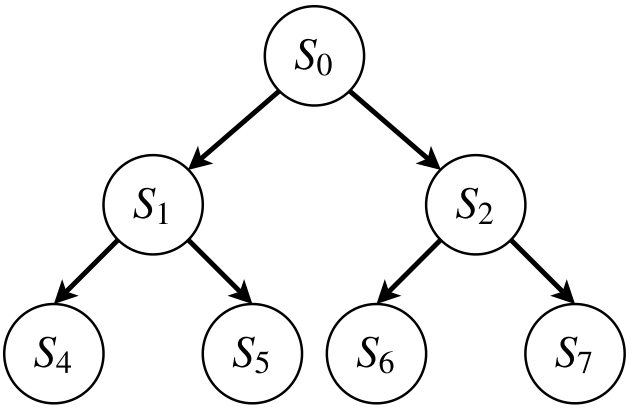


Figure 4.8: The binary tree structure MDP ($\mathcal{M}_1$) with one initial state, similar as discussed in [184]. In this subsection, we focus on the MDPs that have no duplicated states. The initial state distribution of the MDP is uniform and the environment dynamics is deterministic. For $\mathcal{M}_1$ the worst case exploration is random exploration and each trajectory will be visited at same probability under random exploration. Note that in this type of MDP, the Assumption 5 is satisfied.

The definitions provide a quantitative metric to evaluate the quality of exploration. Intuitively, the quality of exploration should be determined by how frequently it will hit different good trajectories. We use Def 9 for theoretical analysis and Def 10 for practical evaluation.

Lemma 2 (The Exploration Efficiency of Random Policy)**.** *The Exploration Efficiency of random exploration policy in a binary tree MDP ($\mathcal{M}_1$) is given as:*

$$p_{\pi_r,\mathcal{M}}(n_{traj} \geq i|k) = 1 - \sum_{i'=0}^{i-1} C_{|\mathcal{T}|}^{i'} \frac{\sum_{j=0}^{i'} (-1)^j C_{i'}^j (N - |\mathcal{T}| + i' - j)^k}{N^k},$$

where N denotes the total number of different trajectories in the MDP. In binary tree MDP $\mathcal{M}_1$, $N = |\mathcal{S}_0||\mathcal{A}|^T$, where the $|\mathcal{S}_0|$ denotes the number of distinct initial states. $|\mathcal{T}|$ denotes the number of optimal trajectories. π_r denotes the random exploration policy, which means the probability of hitting each trajectory in $\mathcal{M}_1$ is equal.

The proof of Lemma 2 is available in Appendix A.

4.2.10 Joint Analysis Combining Exploration and Supervision

In this section, we jointly consider the learning efficiency and exploration efficiency to study the generalization performance. Concretely, we would like to study if we interact with the environment a certain number of episodes, what is the worst generalization performance we can expect with certain probability, if RPG is applied.

Corollary 5 (RL Generalization Performance)**.** *Given a MDP where the UOP (Def 8) is deterministic, let $|\mathcal{H}|$ be the size of the hypothesis space, and let δ, n, k be fixed, the following inequality holds with probability at least $1-\delta'$:*

$$\sum\nolimits_{\tau} p_\theta(\tau) r(\tau) \geq D(1+e)^{\eta(1-m)T},$$

where k is the number of episodes we have explored in the MDP, n is the number of distinct optimal state-action pairs we needed from the UOP (i.e., size of training data.). n' denotes the number of distinct optimal state-action pairs collected by the random exploration. $\eta = 2\sqrt{\frac{1}{2n}\log\frac{2|\mathcal{H}|p_{\pi_r,\mathcal{M}}(n'\geq n|k)}{p_{\pi_r,\mathcal{M}}(n'\geq n|k)-1+\delta'}}$.

The proof of Corollary 5 is provided in Appendix A. Corollary 5 states that the probability of sampling optimal trajectories is the main bottleneck of exploration and generalization, instead of state space dimension. In general, the optimal exploration strategy depends on the properties of MDPs. In this work, we focus on improving learning efficiency, i.e., learning optimal ranking instead of estimating value functions. The discussion of optimal exploration is beyond the scope of this work.

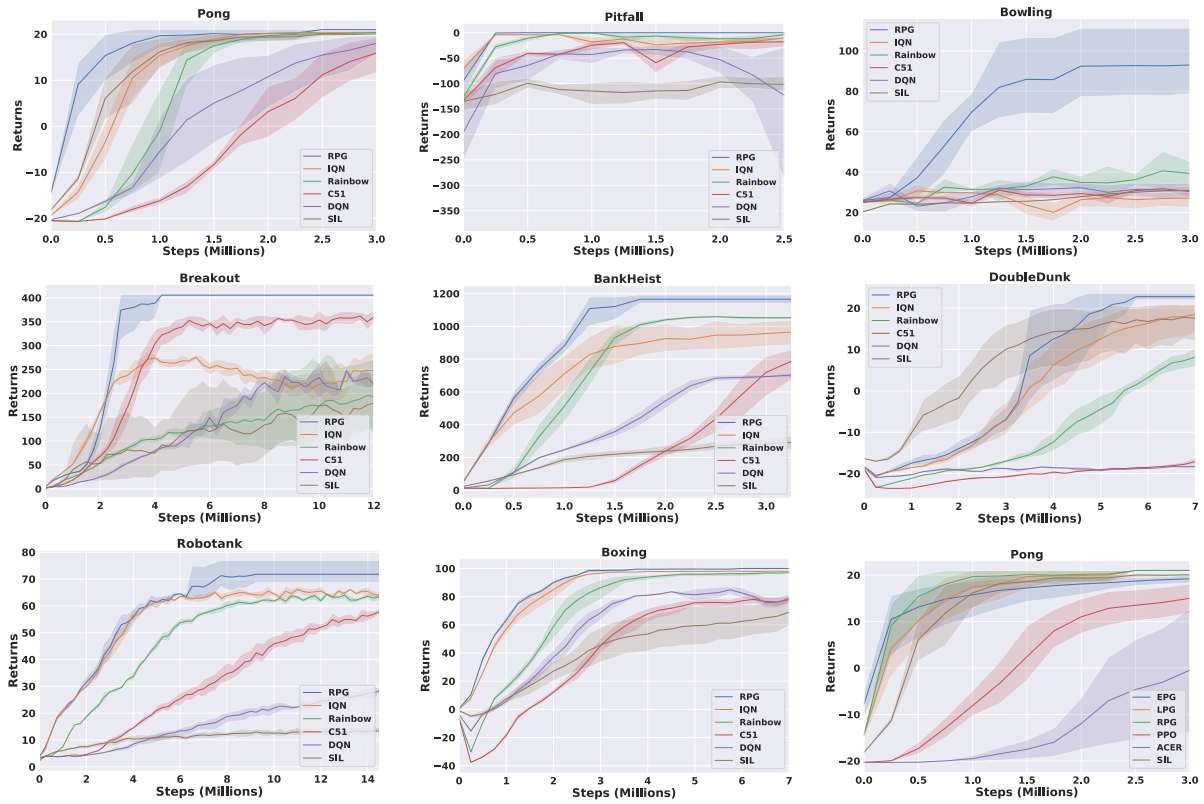


Figure 4.9: The training curves of the proposed RPG and state-of-the-art. All results are averaged over random seeds from 1 to 5. The x-axis represents the number of steps interacting with the environment (we update the model every four steps) and the y-axis represents the averaged training episodic return. The error bars are plotted with a confidence interval of 95%.

4.2.11 Experimental Results

To evaluate the sample-efficiency of Ranking Policy Gradient (RPG), we focus on Atari 2600 games in OpenAI gym [18, 24], without randomly repeating the previous action. We compare our method with the state-of-the-art baselines including DQN [132], C51 [17], IQN [42], RAINBOW [80], and self-imitation learning (SIL) [145]. For reproducibility, we use the implementation provided in Dopamine framework[1] [34] for all baselines and proposed methods, except for SIL using the official implementation. [2]. Follow the standard practice [145,

[1] https://github.com/google/dopamine

[2] https://github.com/junhyukoh/self-imitation-learning

80, 42, 17], we report the training performance of all baselines as the increase of interactions with the environment, or proportionally the number of training iterations. We run the algorithms with five random seeds and report the average rewards with 95% confidence intervals. The implementation details of the proposed RPG and its variants are given as follows[3]:

EPG: EPG is the stochastic listwise policy gradient (see Eq (4.10)) incorporated with the proposed off-policy learning. More concretely, we apply trajectory reward shaping (TRS, Def 6) to all trajectories encountered during exploration and train vanilla policy gradient using the off-policy samples. This is equivalent to minimizing the cross-entropy loss (see Appendix Eq (4.15)) over the near-optimal trajectories.

LPG: LPG is the deterministic listwise policy gradient with the proposed off-policy learning. The only difference between EPG and LPG is that LPG chooses action deterministically (see Appendix Eq (4.9)) during evaluation.

RPG: RPG explores the environment using a separate EPG agent in Pong and IQN in other games. Then RPG conducts supervised learning by minimizing the hinge loss Eq (4.14). It is worth noting that the exploration agent (EPG or IQN) can be replaced by any existing exploration method. In our RPG implementation, we collect all trajectories with the trajectory reward no less than the threshold c without eliminating the duplicated trajectories and we empirically found it is a reasonable simplification.

Sample-efficiency. As the results shown in Figure 4.9, our approach, RPG, significantly outperforms the state-of-the-art baselines in terms of sample-efficiency at all tasks. Furthermore, RPG not only achieved the most sample-efficient results, but also reached the highest final performance at Robotank, DoubleDunk, Pitfall, and Pong, comparing to any

[3]Code is available at https://github.com/illidanlab/rpg.

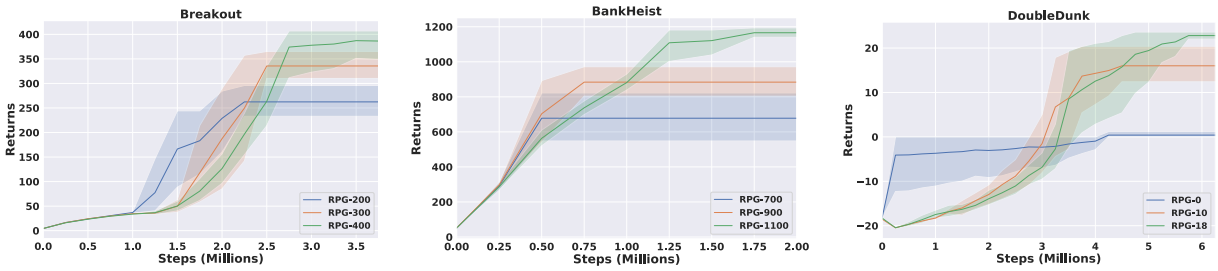


Figure 4.10: The trade-off between sample efficiency and optimality.

model-free state-of-the-art. In reinforcement learning, the stability of algorithm should be emphasized as an important issue. As we can see from the results, the performance of baselines varies from task to task. There is no single baseline consistently outperforms others. In contrast, due to the reduction from RL to supervised learning, RPG is consistently stable and effective across different environments. In addition to the stability and efficiency, RPG enjoys simplicity at the same time. In the environment Pong, it is surprising that RPG without any complicated exploration method largely surpassed the sophisticated value-function based approaches. More details of hyperparameters are provided in the Appendix Section A.

4.2.12 Ablation Study

The effectiveness of pairwise ranking policy and off-policy learning as supervised learning. To get a better understanding of the underlying reasons that RPG is more sample-efficient than DQN variants, we performed ablation studies in the Pong environment by varying the combination of policy functions with the proposed off-policy learning. The results of EPG, LPG, and RPG are shown in the bottom right, Figure 4.9. Recall that EPG and LPG use listwise policy gradient (vanilla policy gradient using softmax as policy function) to conduct exploration, the off-policy learning minimizes the cross-entropy loss Eq (4.15). In contrast, RPG shares the same exploration method as EPG and LPG while uses pairwise

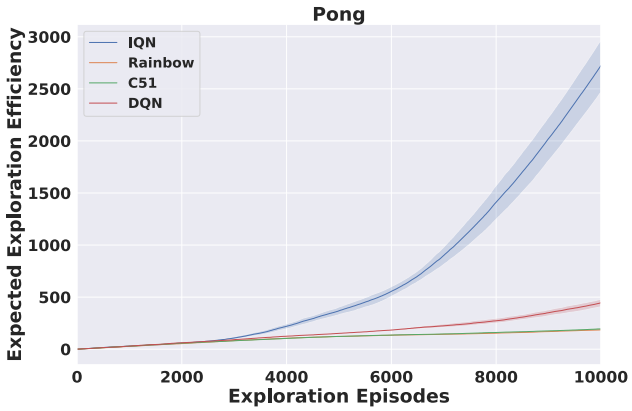


Figure 4.11: Expected exploration efficiency of state-of-the-art. The results are averaged over random seeds from 1 to 10.

ranking policy Eq (4.5) in off-policy learning that minimizes hinge loss Eq (4.14). We can see that RPG is more sample-efficient than EPG/LPG in learning deterministic optimal policy. We also compared the advanced on-policy method Proximal Policy Optimization (PPO) [170] with EPG, LPG, and RPG. The proposed off-policy learning largely surpassed the best on-policy method. Therefore, we conclude that off-policy as supervised learning contributes to the sample-efficiency substantially, while the pairwise ranking policy can further accelerate the learning. In addition, we compare RPG to representative off-policy policy gradient approach: ACER [208]. As the results shown, the proposed off-policy learning framework is more sample-efficient than the state-of-the-art off-policy policy gradient approaches.

On the Trade-off between Sample-Efficiency and Optimality. Results in Figure 4.10 show that there is a trade-off between sample efficiency and optimality, which is controlled by the trajectory reward threshold c. Recall that c determines how close is the learned UNOP to optimal policies. A higher value of c leads to a less frequency of near-optimal trajectories being collected and and thus a lower sample efficiency, and however the algorithm is expected to converge to a strategy of better performance. We note that c is the only parameter we tuned across all experiments.

Exploration Efficiency. We empirically evaluate the Expected Exploration Efficiency (Def 9) of the state-of-the-art on PONG. It is worth noting that the RL generalization performance is determined by both of learning efficiency and exploration efficiency. Therefore, higher exploration efficiency does not necessarily lead to more sample efficient algorithm due to the learning inefficiency, as demonstrated by RAINBOW and DQN (see Figure 4.11). Also, the Implicit Quantile achieves the best performance among baselines, since its exploration efficiency largely surpasses other baselines.

4.2.13 Conclusion

In this work, we introduced ranking policy gradient methods that, for the first time, approach the RL problem from a ranking perspective. Furthermore, towards the sample-efficient RL, we propose an off-policy learning framework, which trains RL agents in a supervised learning manner and thus largely facilitates the learning efficiency. The off-policy learning framework uses generalized policy iteration for exploration and exploits the stableness of supervised learning for deriving policy, which accomplishes the unbiasedness, variance reduction, off-policy learning, and sample efficiency at the same time. Besides, we provide an alternative approach to analyze the sample complexity of RL, and show that the sample complexity of RPG has no dependency on the state space dimension. Last but not least, empirical results show that RPG achieves superior performance as compared to the state-of-the-art.

Chapter 5

Collaborative Multi-Agent Learning

In this chapter, we investigate the scalability of collaborative learning in the context of multi-agent learning for a real-world fleet management application. We propose to transfer the coordination of a large number of learning agents into a linear programming problem, with proper domain knowledge to guide the optimization. We show the superiority of this global collaboration compared to individual learning through extensive evaluation on the real-world traffic data.

5.1 Introduction

Large-scale online ride-sharing platforms such as Uber [201], Lift [126], and Didi Chuxing [40] have transformed the way people travel, live and socialize. By leveraging the advances in and wide adoption of information technologies such as cellular networks and global positioning systems, the ride-sharing platforms redistribute underutilized vehicles on the roads to passengers in need of transportation. The optimization of transportation resources greatly alleviated traffic congestion and calibrated the once significant gap between transport demand and supply [112].

One key challenge in ride-sharing platforms is to balance the demands and supplies, i.e., orders of the passengers and drivers available for picking up orders. In large cities, although millions of ride-sharing orders are served everyday, an enormous number of passengers requests

remain unserviced due to the lack of available drivers nearby. On the other hand, there are plenty of available drivers looking for orders in other locations. If the available drivers were directed to locations with high demand, it will significantly increase the number of orders being served, and thus simultaneously benefit all aspects of the society: utility of transportation capacity will be improved, income of drivers and satisfaction of passengers will be increased, and market share and revenue of the company will be expanded. *fleet management* is a key technical component to balance the differences between demand and supply, by reallocating available vehicles ahead of time, to achieve high efficiency in serving future demand.

Even though rich historical demand and supply data are available, using the data to seek an optimal allocation policy is not an easy task. One major issue is that changes in an allocation policy will impact future demand-supply, and it is hard for supervised learning approaches to capture and model these real-time changes. On the other hand, the reinforcement learning (RL) [186], which learns a policy by interacting with a complicated environment, has been naturally adopted to tackle the fleet management problem [64, 65, 211]. However, the high-dimensional and complicated dynamics between demand and supply can hardly be modeled accurately by traditional RL approaches.

Recent years witnessed tremendous success in deep reinforcement learning (DRL) in modeling intellectual challenging decision-making problems [132, 174, 175] that were previously intractable. In the light of such advances, in this chapter we propose a novel DRL approach to learn highly efficient allocation policies for fleet management. There are significant technical challenges when modeling fleet management using DRL:

1) *Feasibility of problem setting.* The RL framework is reward-driven, meaning that a sequence of *actions* from the policy is evaluated solely by the *reward* signal from environment [11].

The definitions of agent, reward and action space are essential for RL. If we model the allocation policy using a centralized agent, the action space can be prohibitively large since an action needs to decide the number of available vehicles to reposition from each location to its nearby locations. Also, the policy is subject to a feasibility constraint enforcing that the number of repositioned vehicles needs to be no larger than the current number of available vehicles. To the best of our knowledge, this high-dimensional exact-constrain satisfaction policy optimization is not computationally tractable in DRL: applying it in a very small-scale problem could already incur high computational costs [154].

2) *Large-scale Agents.* One alternative approach is to instead use a multi-agent DRL setting, where each available vehicle is considered as an agent. The multi-agent recipe indeed alleviates the curse of dimensionality of action space. However, such setting creates thousands of agents interacting with the environment at each time. Training a large number of agents using DRL is again challenging: the environment for each agent is non-stationary since other agents are learning and affecting the environment at same the time. Most of existing studies [125, 60, 189] allow coordination among only a small set of agents due to high computational costs.

3) *Coordinations and Context Dependence of Action space* Facilitating coordination among large-scale agents remains a challenging task. Since each agent typically learns its own policy or action-value function that are changing over time, it is difficult to coordinate agents for a large number of agents. Moreover, the action space is dynamic changing over time since agents are navigating to different locations and the number of feasible actions depends on the geographic context of the location.

In this paper, we propose a contextual multi-agent DRL framework to resolve the aforementioned challenges. Our major contributions are listed as follows:

- We propose an efficient multi-agent DRL setting for large-scale fleet management problem by a proper design of agent, reward and state.

- We propose contextual multi-agent reinforcement learning framework in which three concrete algorithms: *contextual multi-agent actor-critic* (cA2C), *contextual deep Q-learning* (cDQN), and *Contextual multi-agent actor-critic with linear programming* (LP-cA2C) are developed. For the first time in multi-agent DRL, the contextual algorithms can not only achieve efficient coordination among thousands of learning agents at each time, but also adapt to dynamically changing action spaces.

- In order to train and evaluate the RL algorithm, we developed a simulator that simulates real-world traffic activities perfectly after calibrating the simulator using real historical data provided by Didi Chuxing [40].

- Last but not least, the proposed contextual algorithms significantly outperform the state-of-the-art methods in multi-agent DRL with a much less number of repositions needed.

The rest of this chapter is organized as follows. We first give a literature review on the related work in Sec 5.2. Then the problem statement is elaborated in Sec 5.3 and the simulation platform we built for training and evaluation are introduced in Sec 5.6. The methodology is described in Sec 5.4. Quantitative and qualitative results are presented in Sec 6.6. Finally, we conclude our work in Sec 5.8.

5.2 Related Works

Intelligent Transportation System. Advances in machine learning and traffic data analytics lead to widespread applications of machine learning techniques to tackle challenging traffic problems. One trending direction is to incorporate reinforcement learning algorithms in complicated traffic management problems. There are many previous studies that have demonstrated the possibility and benefits of reinforcement learning. Our work has close connections to these studies in terms of problem setting, methodology and evaluation. Among the traffic applications that are closely related to our work, such as taxi dispatch systems or traffic light control algorithms, multi-agent RL has been explored to model the intricate nature of these traffic activities [14, 172, 128]. The promising results motivated us to use multi-agent modeling in the fleet management problem. In [64], an adaptive dynamic programming approach was proposed to model stochastic dynamic resource allocation. It estimates the returns of future states using a piecewise linear function and delivers actions (assigning orders to vehicles, reallocate available vehicles) given states and one step future states values, by solving an integer programming problem. In [65], the authors further extended the approach to the situations that an action can span across multiple time periods. These methods are hard to be directly utilized in the real-world setting where orders can be served through the vehicles located in multiple nearby locations.

Multi-agent reinforcement learning. Another relevant research topic is multi-agent reinforcement learning [27] where a group of agents share the same environment, in which they receive rewards and take actions. [190] compared and contrasted independent Q-learning and a cooperative counterpart in different settings, and empirically showed that the learning speed can benefit from the cooperation among agents. Independent Q-learning is extended

into DRL in [189], where two agents are cooperating or competing with each other only through the reward. In [60], the authors proposed a counterfactual multi-agent policy gradient method that uses a centralized advantage to estimate whether the action of one agent would improve the global reward, and decentralized actors to optimize the agent policy. Ryan *et al.* also utilized the framework of decentralized execution and centralized training to develop multi-agent multi-agent actor-critic algorithm that can coordinate agents in mixed cooperative-competitive environments [125]. However, none of these methods were applied when there are a large number of agents due to the communication cost among agents. Recently, few works [230, 217] scaled DRL methods to a large number of agents, while it is not applicable to apply these methods to complex real applications such as fleet management. In [140, 141], the authors studied large-scale multi-agent planning for fleet management with explicitly modeling the expected counts of agents.

Deep reinforcement learning. DRL utilizes neural network function approximations and are shown to have largely improved the performance over challenging applications [175, 132]. Many sophisticated DRL algorithms such as DQN [132], A3C [131] were demonstrated to be effective in the tasks in which we have a clear understanding of rules and have easy access to millions of samples, such as video games [24, 18]. However, DRL approaches are rarely seen to be applied in complicated real-world applications, especially in those with high-dimensional and non-stationary action space, lack of well-defined reward function, and in need of coordination among a large number of agents. In this chapter, we show that through careful reformulation, the DRL can be applied to tackle the fleet management problem.

5.3 Problem Statement

In this chapter, we consider the problem of managing a large set of available homogeneous vehicles for online ride-sharing platforms. The goal of the management is to maximize the gross merchandise volume (GMV: the value of all the orders served) of the platform by repositioning available vehicles to the locations with larger demand-supply gap than the current one. This problem belongs to a variant of the classical fleet management problem [47]. A spatial-temporal illustration of the problem is available in Figure 5.1. In this example, we use *hexagonal-grid world* to represent the map and split the duration of one day into $T = 144$ time intervals (one for 10 minutes). At each time interval, the orders emerge stochastically in each grid and are served by the available vehicles in the same grid or six nearby grids. The goal of fleet management here is to decide how many available vehicles to relocate from each grid to its neighbors in ahead of time, so that most orders can be served.

To tackle this problem, we propose to formulate the problem using *multi-agent reinforcement learning* [27]. In this formulation, we use a set of homogeneous agents with small action spaces, and split the global reward into each grid. This will lead to a much more efficient learning procedure than the single agent setting, due to the simplified action dimension and the explicit credit assignment based on split reward. Formally, we model the fleet management problem as a Markov game G for N agents, which is defined by a tuple $G = (N, \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma)$, where $N, \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma$ are the number of agents, sets of states, joint action space, transition probability functions, reward functions, and a discount factor respectively. The definitions are given as follows:

- **Agent**: We consider an available vehicle (or equivalently an idle driver) as an agent, and the vehicles in the same spatial-temporal node are homogeneous, i.e., the vehicles

located at the same region at the same time interval are considered as same agents (where agents have the same policy). Although the number of unique heterogeneous agents is always N, the number of agents N_t is changing over time.

- **State** $\mathbf{s}_t \in \mathcal{S}$: We maintain a global state $\mathbf{s}_t$ at each time t, considering the spatial distributions of available vehicles and orders (i.e. the number of available vehicles and orders in each grid) and current time t (using one-hot encoding). The state of an agent i, $\mathbf{s}_t^i$, is defined as the identification of the grid it located and the shared global state i.e. $\mathbf{s}_t^i = [\mathbf{s}_t, \mathbf{g}_j] \in R^{N \times 3+T}$, where $\mathbf{g}_j$ is the one-hot encoding of the grid ID. We note that agents located at same grid have the same state $\mathbf{s}_t^i$.

- **Action** $a_t \in \mathcal{A} = \mathcal{A}_1 \times ... \times \mathcal{A}_{N_t}$: a *joint action* $\mathbf{a}_t = \{a_t^i\}_1^{N_t}$ instructing the allocation strategy of all available vehicles at time t. The action space $\mathcal{A}_i$ of an individual agent specifies where the agent is able to arrive at the next time, which gives a set of seven discrete actions denoted by $\{k\}_{k=1}^7$. The first six discrete actions indicate allocating the agent to one of its six neighboring grids, respectively. The last discrete action $a_t^i = 7$ means staying in the current grid. For example, the action $a_0^1 = 2$ means to relocate the 1st agent from the current grid to the second nearby grid at time 0, as shown in Figure 5.1. For a concise presentation, we also use $a_t^i \triangleq [\mathbf{g}_0, \mathbf{g}_1]$ to represent agent i moving from grid $\mathbf{g}_0$ to $\mathbf{g}_1$. Furthermore, the action space of agents depends on their locations. The agents located at corner grids have a smaller action space. We also assume that the action is deterministic: if $a_t^i \triangleq [\mathbf{g}_0, \mathbf{g}_1]$, then agent i will arrive at the grid $\mathbf{g}_1$ at time $t + 1$.

- **Reward function** $\mathcal{R}_i \in \mathcal{R} = \mathcal{S} \times \mathcal{A} \to \mathbb{R}$: Each agent is associated with a reward function $\mathcal{R}_i$ and all agents in the same location have the same reward function. The

i-th agent attempts to maximize its own expected discounted return: $\mathbb{E}\left[\sum_{k=0}^{\infty}\gamma^k r_{t+k}^i\right]$. The individual reward r_t^i for the i-th agent associated with the action $\mathbf{a}_t^i$ is defined as the averaged revenue of all agents arriving at the same grid as the i-th agent at time $t+1$. Since the individual rewards at same time and the same location are same, we denote this reward of agents at time t and grid $\mathbf{g}_j$ as $r_t(\mathbf{g}_j)$. Such design of rewards aims at avoiding greedy actions that send too many agents to the location with high value of orders, and aligning the maximization of each agent's return with the maximization of GMV (value of all served orders in one day). Its effectiveness is empirically verified in Sec 6.6.

- **State transition probability** $p(\mathbf{s}_{t+1}|\mathbf{s}_t, a_t) : \mathcal{S}\times\mathcal{A}\times\mathcal{S}\to[0,1]$: It gives the probability of transiting to $\mathbf{s}_{t+1}$ given a joint action $\mathbf{a}_t$ is taken in the current state $\mathbf{s}_t$. Notice that although the action is deterministic, new vehicles and orders will be available at different grids each time, and existing vehicles will become off-line via a random process.

To be more concrete, we give an example based on the above problem setting in Figure 5.1. At time $t=0$, agent 1 is repositioned from $\mathbf{g}_0$ to $\mathbf{g}_2$ by action a_0^1, and agent 2 is also repositioned from $\mathbf{g}_1$ to $\mathbf{g}_2$ by action a_0^2. At time $t=1$, two agents arrive at $\mathbf{g}_2$, and a new order with value 10 also emerges at same grid. Therefore, the reward r_1 for both a_0^1 and a_0^2 is the averaged value received by agents at $\mathbf{g}_2$, which is $10/2=5$.

It's worth to note that this reward design may not lead to the optimal reallocation strategy though it empirically leads to good reallocation policy. We give a simple example to illustrate this problem. We use the grid world map as show in Figure 5.1. At time $t=1$, there is an order with value 100 emerged in $\mathbf{g}_1$ and another order with value 10 emerged in $\mathbf{g}_0$. Suppose

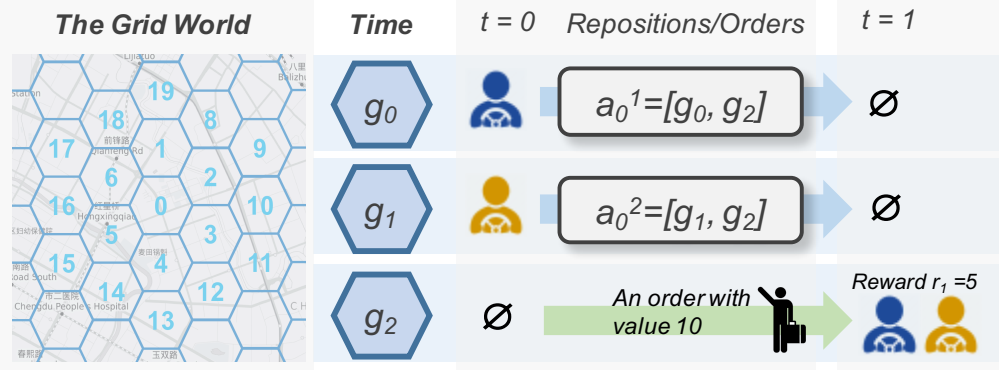


Figure 5.1: The grid world system and a spatial-temporal illustration of the problem setting.

we have two agents that are available in grid $\mathbf{g}_0$ at time $t = 0$. The optimal reallocation strategy in this case is to ask one agent stay in $\mathbf{g}_0$ and another go to $\mathbf{g}_1$, by which we can receive the total reward 110. However, in the current setting, each agent trys to maximize its own reward. As a result, both of them will go to $\mathbf{g}_1$ and receive 50 reward and none of them will go to $\mathbf{g}_1$ since the reward they can receive is less than 50. However, we show that there are few ways to approximate this global optimal allocation strategy using the individual action function of each agent.

5.4 Contextual Multi-Agent Reinforcement Learning

In this section, we present two novel contextual multi-agent RL approaches: contextual multi-agent actor-critic (cA2C) and contextual DQN (cDQN) algorithm. We first briefly introduce the basic multi-agent RL method.

5.4.1 Independent DQN

Independent DQN [189] combines independent Q-learning [190] and DQN [132]. A straightforward extension of independent DQN from small scale to a large number of agents, is to share

network parameters and distinguish agents with their IDs [230]. The network parameters can be updated by minimizing the following loss function, with respect to the transitions collected from all agents:

$$\mathbb{E}\left[Q(\mathbf{s}_t^i, a_t^i; \theta) - \left(r_{t+1}^i + \gamma \max_{a_{t+1}^i} Q(\mathbf{s}_{t+1}^i, a_{t+1}^i; \theta')\right)\right]^2, \tag{5.1}$$

where θ' includes parameters of the target Q network updated periodically, and θ includes parameters of behavior Q network outputting the action value for ϵ-greedy policy, same as the algorithm described in [132]. This method could work reasonably well after extensive tunning but it suffers from high variance in performance, and it also repositions too many vehicles. Moreover, coordination among massive agents is hard to achieve since each unique agent executes its action independently based on its action values.

5.4.2 Contextual DQN

Since we assume that the location transition of an agent after the allocation action is deterministic, the actions that lead the agents to the same grid should have the same action value. In this case, the number of unique action-values for all agents should be equal to the number of grids N. Formally, for any agent i where $\mathbf{s}_t^i = [\mathbf{s}_t, \mathbf{g}_i]$, $a_t^i \triangleq [\mathbf{g}_i, \mathbf{g}_d]$ and $\mathbf{g}_i \in Ner(\mathbf{g}_d)$, the following holds:

$$Q(\mathbf{s}_t^i, a_t^i) = Q(\mathbf{s}_t, \mathbf{g}_d) \tag{5.2}$$

Hence, at each time step, we only need N unique action-values $(Q(\mathbf{s}_t, \mathbf{g}_j), \forall j = 1, \ldots, N)$ and the optimization of Eq (5.1) can be replaced by minimizing the following mean-squared loss:

$$\left[Q(\mathbf{s}_t, \mathbf{g}_d; \theta) - \left(r_{t+1}(\mathbf{g}_d) + \gamma \max_{\mathbf{g}_p \in \mathrm{Ner}(\mathbf{g}_d)} Q(\mathbf{s}_{t+1}, \mathbf{g}_p; \theta') \right) \right]^2 . \tag{5.3}$$

This accelerates the learning procedure since the output dimension of the action value function is reduced from $\mathbb{R}^{|\mathbf{s}t|} \to \mathbb{R}^7$ to $\mathbb{R}^{|\mathbf{s}t|} \to \mathbb{R}$. Furthermore, we can build a centralized action-value table at each time for all agents, which can serve as the foundation for coordinating the actions of agents.

Geographic context. In hexagonal grids systems, border grids and grids surrounded by infeasible grids (e.g., a lake) have reduced action dimensions. To accommodate this, for each grid we compute a *geographic context* $\mathbf{G}_{\mathbf{g}_j} \in \mathbb{R}^7$, which is a binary vector that filters out invalid actions for agents in grid $\mathbf{g}_j$. The kth element of vector $\mathbf{G}_{\mathbf{g}_j}$ represents the validity of moving toward kth direction from the grid $\mathbf{g}_j$. Denote $\mathbf{g}_d$ as the grid corresponds to the kth direction of grid $\mathbf{g}_j$, the value of the kth element of $\mathbf{G}_{\mathbf{g}_j}$ is given by:

$$[\mathbf{G}_{t,\mathbf{g}_j}]_k = \begin{cases} 1, & \text{if } \mathbf{g}_d \text{ is valid grid,} \\ 0, & \text{otherwise,} \end{cases} \tag{5.4}$$

where $k = 0, \ldots, 6$ and last dimension of the vector represents direction staying in same grid, which is always 1.

Collaborative context. To avoid the situation that agents are moving in conflict directions (i.e., agents are repositioned from grid $\mathbf{g}_1$ to $\mathbf{g}_2$ and $\mathbf{g}_2$ to $\mathbf{g}_1$ at the same time.), we provide a *collaborative context* $\mathbf{C}_{t,\mathbf{g}_j} \in \mathbb{R}^7$ for each grid $\mathbf{g}_j$ at each time. Based on the centralized action values $Q(\mathbf{s}_t, \mathbf{g}_j)$, we restrict the valid actions such that agents at the grid $\mathbf{g}_j$ are

navigating to the neighboring grids with higher action values or staying unmoved. Therefore, the binary vector $\mathbf{C}_{t,\mathbf{g}_j}$ eliminates actions to grids with lower action values than the action staying unmoved. Formally, the kth element of vector $\mathbf{C}_{t,\mathbf{g}_j}$ that corresponds to action value $Q(\mathbf{s}_t, \mathbf{g}_i)$ is defined as follows:

$$[\mathbf{C}_{t,\mathbf{g}_j}]_k = \begin{cases} 1, & \text{if } Q(\mathbf{s}_t, \mathbf{g}_i) >= Q(\mathbf{s}_t, \mathbf{g}_j), \\ 0, & \text{otherwise.} \end{cases} \tag{5.5}$$

After computing both collaborative and geographic context, the ϵ-greedy policy is then performed based on the action values survived from the two contexts. Suppose the original action values of agent i at time t is $\mathbf{Q}(\mathbf{s}_t^i) \in \mathbb{R}^7_{\geq 0}$, given state $\mathbf{s}_t^i$, the valid action values after applying contexts is as follows:

$$\mathbf{q}(\mathbf{s}_t^i) = \mathbf{Q}(\mathbf{s}_t^i) * \mathbf{C}_{t,\mathbf{g}_j} * \mathbf{G}_{t,\mathbf{g}_j}. \tag{5.6}$$

The coordination is enabled because that the action values of different agents lead to the same location are restricted to be same so that they can be compared, which is impossible in independent DQN. This method requires that action values are always non-negative, which will always hold because that agents always receive nonnegative rewards. The algorithm of cDQN is elaborated in Alg 5.2.

5.4.3 Contextual Actor-Critic

We now present the contextual multi-agent actor-critic (cA2C) algorithm, which is a multi-agent policy gradient algorithm that tailors its policy to adapt to the dynamically changing action space. Meanwhile, it achieves not only a more stable performance but also a much

Algorithm 5.1: ϵ-greedy policy for cDQN

Require: Global state $\mathbf{s}_t$

Compute centralized action value $Q(\mathbf{s}_t, \mathbf{g}_j), \forall j = 1, \ldots, N$
for $i = 1$ to N_t **do**
  Compute action values $\mathbf{Q}^i$ by Eq (5.2), where $(\mathbf{Q}^i)_k = Q(\mathbf{s}_t^i, a_t^i = k)$.
  Compute contexts $\mathbf{C}_{t,\mathbf{g}_j}$ and $\mathbf{G}_{t,\mathbf{g}_j}$ for agent i.
  Compute valid action values $\mathbf{q}_t^i = \mathbf{Q}_t^i * \mathbf{C}_{t,\mathbf{g}_j} * \mathbf{G}_{t,\mathbf{g}_j}$.
  $a_t^i = \text{argmax}_k \mathbf{q}_t^i$ with probability $1 - \epsilon$ otherwise choose an action randomly from the valid actions.
end for
return Joint action $\mathbf{a}_t = \{a_t^i\}_1^{N_t}$.

Algorithm 5.2: Contextual Deep Q-learning (cDQN)

Initialize replay memory D to capacity M
Initialize action-value function with random weights θ or pre-trained parameters.
for $m = 1$ to *max-iterations* **do**
  Reset the environment and reach the initial state $\mathbf{s}_0$.
  for $t = 0$ to T **do**
    Sample joint action $\mathbf{a}_t$ using Alg. 5.1, given $\mathbf{s}_t$.
    Execute a_t in simulator and observe reward $\mathbf{r}_t$ and next state $\mathbf{s}_{t+1}$
    Store the transitions of all agents $(\mathbf{s}_t^i, a_t^i, r_t^i, \mathbf{s}_{t+1}^i, \forall i = 1, ..., N_t)$ in D.
  end for
  for $k = 1$ to M_1 **do**
    Sample a batch of transitions $(\mathbf{s}_t^i, a_t^i, r_t^i, \mathbf{s}_{t+1}^i)$ from D,
    Compute target $y_t^i = r_t^i + \gamma * \max_{a_{t+1}^i} Q(\mathbf{s}_{t+1}^i, a_{t+1}^i; \theta')$.
    Update Q-network as $\theta \leftarrow \theta + \nabla_\theta (y_t^i - Q(\mathbf{s}_t^i, a_t^i; \theta))^2$,
  end for
end for

more efficient learning procedure in a non-stationary environment. There are two main ideas in the design of cA2C: 1) A centralized value function shared by all agents with an expected update; 2) Policy context embedding that establishes explicit coordination among agents, enables faster training and enjoys the flexibility of regulating policy to different action spaces. The centralized state-value function is learned by minimizing the following loss function

derived from Bellman equation:

$$L(\theta_v) = (V_{\theta_v}(\mathbf{s}_t^i) - V_{\text{target}}(\mathbf{s}_{t+1}; \theta_v', \pi))^2, \tag{5.7}$$

$$V_{\text{target}}(\mathbf{s}_{t+1}; \theta_v', \pi) = \sum\nolimits_{a_t^i} \pi(a_t^i|\mathbf{s}_t^i)(r_{t+1}^i + \gamma V_{\theta_v'}(\mathbf{s}_{t+1}^i)). \tag{5.8}$$

where we use θ_v to denote the parameters of the value network and θ_v' to denote the target value network. Since agents staying unmoved at the same time are treated homogeneous and share the same internal state, there are N unique agent states, and thus N unique state-values $(V(\mathbf{s}_t, \mathbf{g}_j), \forall j = 1, ..., N)$ at each time. The state-value output is denoted by $\mathbf{v}_t \in \mathbb{R}^N$, where each element $(\mathbf{v}_t)_j = V(\mathbf{s}_t, \mathbf{g}_j)$ is the expected return received by agent arriving at grid $\mathbf{g}_j$ on time t. In order to stabilize learning of the value function, we fix a target value network parameterized by θ_v', which is updated at the end of each episode. Note that the expected update in Eq (5.7) and training actor/critic in an offline fashion are different from the updates in n-step actor-critic online training using TD error [131], whereas the expected updates and training paradigm are found to be more stable and sample-efficient. This is also in line with prior work in applying actor-critic to real applications [12]. Furthermore, efficient coordination among multiple agents can be established upon this centralized value network.

Policy Context Embedding. Coordination is achieved by masking available action space based on the context. At each time step, the geographic context is given by Eq (5.4) and the collaborative context is computed according to the value network output:

$$[\mathbf{C}_{t,\mathbf{g}_j}]_k = \begin{cases} 1, & if\ V(\mathbf{s}_t, \mathbf{g}_i) >= V(\mathbf{s}_t, \mathbf{g}_j), \\ 0, & \text{otherwise}, \end{cases} \tag{5.9}$$

where the kth element of vector $\mathbf{C}_{t,\mathbf{g}_j}$ corresponds to the probability of the kth action

$\pi(a_t^i = k|\mathbf{s}_t^i)$. Let $\mathbf{P}(\mathbf{s}_t^i) \in \mathbb{R}_{>0}^7$ denote the original logits from the policy network output for the ith agent conditioned on state $\mathbf{s}_t^i$. Let $\mathbf{q}_{\text{valid}}(\mathbf{s}_t^i) = \mathbf{P}(\mathbf{s}_t^i) * \mathbf{C}_{t,\mathbf{g}_j} * \mathbf{G}_{\mathbf{g}_j}$ denote the valid logits considering both geographic and collaborative context for agent i at grid $\mathbf{g}_j$, where $*$ denotes an element-wise multiplication. In order to achieve effective masking, we restrict the output logits $\mathbf{P}(\mathbf{s}_t^i)$ to be positive. The probability of valid actions for all agents in the grid $\mathbf{g}_j$ are given by:

$$\pi_{\theta_p}(a_t^i = k|\mathbf{s}_t^i) = [\mathbf{q}_{valid}(\mathbf{s}_t^i)]_k = \frac{[\mathbf{q}_{valid}(\mathbf{s}_t^i)]_k}{\|\mathbf{q}_{valid}(\mathbf{s}_t^i)\|_1}. \tag{5.10}$$

The gradient of policy can then be written as:

$$\nabla_{\theta_p} J(\theta_p) = \nabla_{\theta_p} \log \pi_{\theta_p}(a_t^i|\mathbf{s}_t^i) A(\mathbf{s}_t^i, a_t^i), \tag{5.11}$$

where θ_p denotes the parameters of policy network and the advantage $A(\mathbf{s}_t^i, a_t^i)$ is computed as follows:

$$A(\mathbf{s}_t^i, a_t^i) = r_{t+1}^i + \gamma V_{\theta_v'}(\mathbf{s}_{t+1}^i) - V_{\theta_v}(\mathbf{s}_t^i). \tag{5.12}$$

The detailed description of cA2C is summarized in Alg 5.4.

5.5 Efficient allocation with linear programming

In this section, we present the proposed LP-cA2C that utilizes the state value functions learned by cA2C and compute the reallocations in a centralized view, which achieves the best performance with higher efficiency.

Algorithm 5.3: Contextual Multi-agent Actor-Critic Policy forward

Require: The global state $\mathbf{s}_t$.

1: Compute centralized state-value $\mathbf{v}_t$
2: **for** i $= 1$ to N_t **do**
3: Compute contexts $\mathbf{C}_{t,\mathbf{g}_j}$ and $\mathbf{G}_{t,\mathbf{g}_j}$ for agent i.
4: Compute action probability distribution $\mathbf{q}_{valid}(\mathbf{s}_t^i)$ for agent i in grid $\mathbf{g}_j$ (Eq (5.10)).
5: Sample action for agent i in grid $\mathbf{g}_j$ based on action probability $\mathbf{p}^i$.
6: **end for**
7: **return** Joint action $\mathbf{a}_t = \{a_t^i\}_1^{N_t}$.

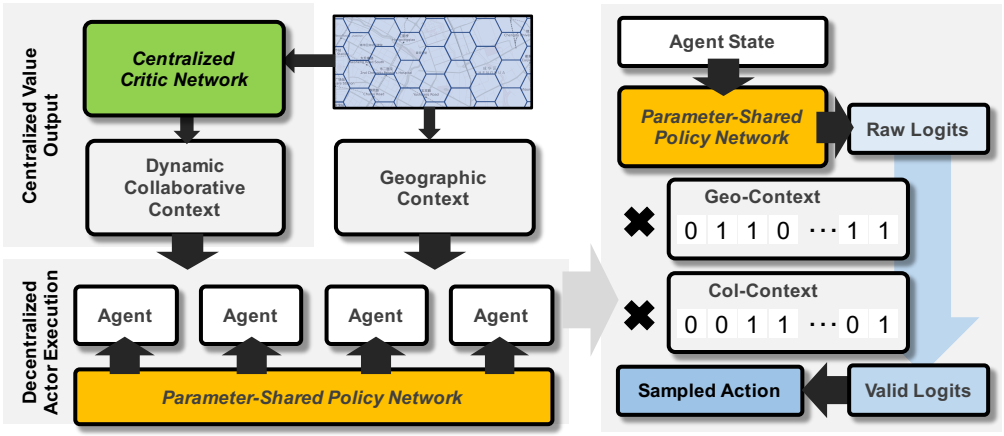


Figure 5.2: Illustration of contextual multi-agent actor-critic. The left part shows the coordination of decentralized execution based on the output of centralized value network. The right part illustrates embedding context to policy network.

From another perspective, if we formulate this problem as a MDP where we have a meta-agent that controls the decisions of all drivers, our goal is to maximize the long term reward of the platform:

$$Q^c(\mathbf{s}, \mathbf{a}) = \mathbf{E}[\sum_{t=1}^{\infty} \gamma^{t-1} r_t(\mathbf{s}_t, \mathbf{a}_t) | \mathbf{s}_0 = \mathbf{s}, \mathbf{a}_0 = \mathbf{a}, \pi^*].$$

The π^* in above formulation denotes the optimal global reallocation strategy. Although the sum of immediate reward received by all agents is equal to the total reward of the platform, maximizing the long term reward of each agent is not equal to maximize the long term reward of the platform, i.e. $\sum_i \max_{a_i} Q(s^i, a^i) \neq \max_{\mathbf{a}} Q^c(\mathbf{s}, \mathbf{a})$. In cooperative multi-agent

Algorithm 5.4: Contextual Multi-agent Actor-Critic Algorithm for N agents

Initialization:
Initialize the value network with fixed value table.
for $m = 1$ to max-iterations **do**
  Reset environment, get initial state $\mathbf{s}_0$.
  Stage 1: Collecting experience
  for $t = 0$ to T **do**
    Sample actions $\mathbf{a}_t$ according to Alg 5.3, given $\mathbf{s}_t$.
    Execute $\mathbf{a}_t$ in simulator and observe reward r_t and next state $\mathbf{s}_{t+1}$.
    Compute value network target as Eq (5.8) and advantage as Eq (5.12) for policy network and store the transitions.
  end for
  Stage 2: Updating parameters
  for $m_1 = 1$ to M_1 **do**
    Sample a batch of experience: $\mathbf{s}_t^i, V_{target}(\mathbf{s}_t^i; \theta_v', \pi)$
    Update value network by minimizing the value loss Eq (5.7) over the batch.
  end for
  for $m_2 = 1$ to M_2 **do**
    Sample a batch of experience: $\mathbf{s}_t^i, a_t^i, A(\mathbf{s}_t^i, a_t^i), \mathbf{C}_{t,g_j}, \mathbf{G}_{g_j}$.
    Update policy network as $\theta_p \leftarrow \theta_p + \nabla_{\theta_p} J(\theta_p)$.
  end for
end for

reinforcement learning, the sum of rewards of multiple agents is the global reward we want to maximize. In this case, given a centralized policy (π^*) for all agents, the summation of long term reward should be equal to the global long term reward.

$$\begin{aligned}
\sum_{i=1}^{N} Q^i(\mathbf{s}^i, a^i) &= \sum_{i=1}^{N} E_{\pi^*}\left[\sum_{t=1}^{\infty} \gamma^{t-1} r_t^i \Big| \mathbf{s}_0^i = \mathbf{s}^i, a_0^i = a^i\right] \\
&= E_{\pi^*}\left[\sum_{t=1}^{\infty} \gamma^{t-1} \sum_{i=1}^{N} r_t^i \Big| \mathbf{s}_0 = \mathbf{s}, \mathbf{a}_0 = \mathbf{a}\right] \\
&= E_{\pi^*}\left[\sum_{t=1}^{\infty} \gamma^{t-1} r_t \Big| \mathbf{s}_0 = \mathbf{s}, \mathbf{a}_0 = \mathbf{a}\right] = Q^c(\mathbf{s}, \mathbf{a})
\end{aligned}$$

However, in this work, this simple relationship does not hold mainly since the number of agents (N_t) is not static. As shown in Eq (5.13), the global reward at time $t+1$ of the platform

is not equal to the sum of all current agents' reward (i.e. $\sum_{i=1}^{N_t} r_{t+1}^i \neq \sum_{i=1}^{N_{t+1}} r_{t+1}^i = r_{t+1}$) even given a centralized policy π^*.

$$\sum_{i=1}^{N_t} Q(\mathbf{s}_t^i, a_t^i) = \sum_{i=1}^{N_t} \mathbf{E}_{\pi^*}[r_{t+1}^i + \gamma \max_{a_{t+1}^i} Q(s_{t+1}^i, a_{t+1}^i)] \tag{5.13}$$

Ideally, we would like to directly learn the centralized action value function Q^c while it's computational intractable to explore and optimize the Q^c in the case we have substantially large action space. Therefore, we need to leverage the averaged long term reward of each agent to approximate the maximization of the centralized action-value function Q^c. In cDQN, we approximate this allocation by avoiding the greedy allocation with $\epsilon-$greedy strategy even during the evaluation stage. In cA2C, the policy will allocate the agents in the same location to its nearby locations with certain probability according to the state-values. In fact, we uses this empirical strategy to better align the joint actions of each individual agent with the action from optimal reallocation. However, both of the cA2C and cDQN try to coordinate agents from a localized view, in which each agent only consider its nearby situation when they are coordinating. Therefore, the redundant reallocation still exists in those two methods. Other methods that can approximate the centralized action-value function such as VDN [185] and QMIX [160] are not able to scale to large number of agents.

In this work, we propose to approximate the centralized policy by formulating the reallocation as a linear programming problem.

$$\begin{aligned} \max_{\mathbf{y}(\mathbf{s}_t)} \quad & \left(\mathbf{v}(\mathbf{s}_t)^T \mathbf{A}_t - \mathbf{c}_t^T\right) \mathbf{y}(\mathbf{s}_t) - \lambda \|\mathbf{D}\left(\mathbf{o}_{t+1} - \mathbf{A}_t \mathbf{y}(\mathbf{s}_t)\right)\|_2^2 \\ \text{s.t.} \quad & \mathbf{y}(\mathbf{s}_t) \geq 0 \\ & \mathbf{B}_t \mathbf{y}(\mathbf{s}_t) = \mathbf{d}_t \end{aligned} \tag{5.14}$$

where the vector $\mathbf{y}(\mathbf{s}_t) \in R^{N_r(t)\times 1}$ denotes the feasible repositions for all agents at current time step t. Each element in $\mathbf{y}(\mathbf{s}_t)$ represents one reposition from current grid to its nearby grid. $N_r(t)$ is the total number of feasible reposition direction. The number of feasible repositions depends on the current state values in each grid since we reallocate agents from location with lower state value to the grid with higher state value. $\mathbf{A} \in R^{N\times N_r(t)}$ is a indicator matrix that denotes the allocations that dispatch drivers into the grid, i.e. $\mathbf{A}_{i,j} \in \{0, 1\}$. $\mathbf{A}_{i,j} = 1$ means the j-th reposition reallocates agents into the i-th grid. Similarly, $\mathbf{B} \in R^{N\times N_r(t)}$ is the indicator matrix that denotes the allocations that dispatch drivers out of the grid. $\mathbf{D} \in \{0, 1\}^{N\times N}$ is the adjacency matrix denotes the connectivity of the grid world. $\mathbf{o}_{t+1}$ denotes the estimated number of orders in each grid at next time step. $\mathbf{c}_t \in R^{N_r(t)\times 1}$ denotes the cost associated with each reposition and $\mathbf{s}(\mathbf{s}_t) \in R^{N\times 1}$ denotes the state value for each grid in time step t.

The first term in Eq (5.14) approximates our goal that we want to maximize the long term reward of the platform. Since the state value can be interpreted as the averaged long term reward one agent will receive if it appears in certain grid, the first term represents the total reward minus the total cost associated with the repositions. However, optimizing the first term will lead to a greedy solution that reallocates all the agents to the nearby grid with highest state value minus the cost. To alleviate this greedy reallocation, we add the second term to regularize the number of agents reallocated to each grid. Since the agent in current grid can pick up the orders emerged in nearby grids, we utilize the adjacency matrix to regularize the number of agents reallocated into a group of nearby grids should be close to the number of orders emerged in a group of nearby grids. From another point of view, the second term more focus on the immediate reward since it prefer the solution that allocates right amount of agents to pick-up the orders without consider the future income that an

agent can receive by that reposition. The regularization parameter λ is used to balance the long term reward and the immediate reward. The two flow conservation constrains requires the number of repositions should be positive and the number of repositions from current grid should be equal to the number of available agents in current grids.

Ideally, we need to solve a integer programming problem where our solution satisfies $\mathbf{y}(\mathbf{s}_t) \in \mathcal{Z}^{N_r}$. However, solving integer programming is NP-hard in worst case while solving its linear programming relaxation is in P. In practice, we solve the linear programming relaxation and round the solution into integers [49].

5.6 Simulator Design

A fundamental challenge of applying RL algorithm in reality is the learning environment. Unlike the standard supervised learning problems where the data is stationary to the learning algorithms and can be evaluated by the training-testing paradigm, the interactive nature of RL introduces intricate difficulties on training and evaluation. One common solution in traffic studies is to build simulators for the environment [211, 172, 128]. In this section, we introduce a simulator design that models the generation of orders, procedure of assigning orders and key driver behaviors such as distributions across the city, on-line/off-line status control in the real world. The simulator serves as the training environment for RL algorithms, as well as their evaluation. More importantly, our simulator allows us to calibrate the key performance index with the historical data collected from a fleet management system, and thus the policies learned are well aligned with real-world traffics.

The Data Description The data provided by Didi Chuxing includes orders and trajectories of vehicles in two cities including Chengdu and Wuhan. Chengdu is covered by a hexagonal

grids world consisting of 504 grids. Wuhan contains more than one thousands grids. The order information includes order price, origin, destination and duration. The trajectories contain the positions (latitude and longitude) and status (on-line, off-line, on-service) of all vehicles every few seconds.

Timeline Design. In one time interval (10 minutes), the main activities are conducted sequentially, also illustrated in Figure 5.4.

- *Vehicle status updates:* Vehicles will be stochastically set offline (i.e., off from service) or online (i.e., start working) following a spatiotemporal distribution learned from real data using the maximum likelihood estimation (MLE). Other types of vehicle status updates include finishing current service or allocation. In other words, if a vehicle is about to finish its service at the current time step, or arriving at the dispatched grid, the vehicles are available for taking new orders or being repositioned to a new destination.

- *Order generation:* The new orders generated at the current time step are bootstrapped from real orders occurred in the same time interval. Since the order will naturally reposition vehicles in a wide range, this procedure keeps the reposition from orders similar to the real data.

- *Interact with agents:* This step computes state as input to fleet management algorithm and applies the allocations for agents.

- *Order assignments:* All available orders are assigned through a two-stage procedure. In the first stage, the orders in one grid are assigned to the vehicles in the same grid. In the second stage, the remaining unfilled orders are assigned to the vehicles in its neighboring grids. In reality, the platform dispatches order to a nearby vehicle within

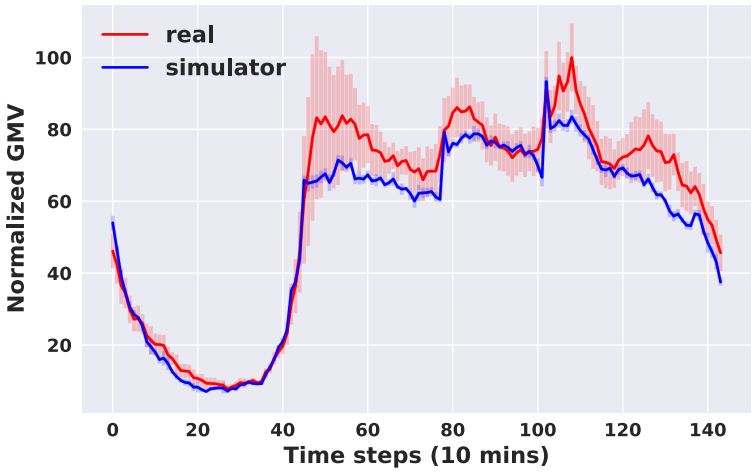


Figure 5.3: The simulator calibration in terms of GMV. The red curves plot the GMV values of real data averaged over 7 days with standard deviation, in 10-minute time granularity. The blue curves are simulated results averaged over 7 episodes.

a certain distance, which is approximately the range covered by the current grid and its adjacent grids. Therefore, the above two-stage procedure is essential to stimulate these real-world activities and the following calibration. This setting differentiates our problem from the previous fleet management problem setting (i.e., demands are served by those resources at the same location only.) and make it impossible to directly apply the classic methods such as adaptive dynamic programming approaches proposed in [64, 65].

Calibration. The effectiveness of the simulator is guaranteed by calibration against the real data regarding the most important performance measurement: the gross merchandise volume (GMV). As shown in Figure 5.3, after the calibration procedure, the GMV in the simulator is very similar to that from the ride-sharing platform. The r^2 between simulated GMV and real GMV is 0.9331 and the Pearson correlation is 0.9853 with p-value $p < 0.00001$.

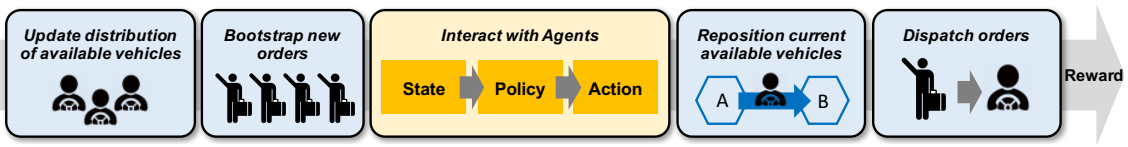


Figure 5.4: Simulator time line in one time step (10 minutes).

5.7 Experiments

In this section, we conduct extensive experiments to evaluate the effectiveness of our proposed method.

5.7.1 Experimental settings

In the following experiments, both of training and evaluation are conducted on the simulator introduced in Sec 5.6. For all the competing methods, we prescribe two sets of random seed that control the dynamics of the simulator for training and evaluation, respectively. Examples of dynamics in simulator include order generations, and stochastically status update of all vehicles. In this setting, we can test the generalization performance of algorithms when it encounters unseen dynamics as in real scenarios. The performance is measured by GMV (the total value of orders served in the simulator) gained by the platform over one episode (144 time steps in the simulator), and order response rate (ORR), which is the averaged number of orders served divided by the number of orders generated. We use the first 15 episodes for training and conduct evaluation on the following ten episodes for all learning methods. The number of available vehicles at each time in different locations is counted by a pre-dispatch procedure. This procedure runs a virtual two-stage order dispatching process to compute the remaining available vehicles in each location. On average, the simulator has 5356 agents per time step waiting for management. All the quantitative results of learning methods presented in this section are averaged over three runs.

5.7.2 Performance comparison

In this subsection, the performance of following methods are extensively evaluated by the simulation.

- **Simulation:** This baseline simulates the real scenario without any fleet management. The simulated results are calibrated with real data in Sec 5.6.

- **Diffusion:** This method diffuses available vehicles to neighboring grids randomly.

- **Rule-based:** This baseline computes a $T \times N$ value table $\mathbf{V}_{rule}$, where each element $\mathbf{V}_{rule}(t, j)$ represents the averaged reward of an agent staying in grid $\mathbf{g}_j$ at time step t. The rewards are averaged over ten episodes controlled by random seeds that are different with testing episodes. With the value table, the agent samples its action based on the probability mass function normalized from the values of neighboring grids at the next time step. For example, if an agent located in $\mathbf{g}_1$ at time t and the current valid actions are $[\mathbf{g}_1, \mathbf{g}_2]$ and $[\mathbf{g}_1, \mathbf{g}_1]$, the rule-based method sample its actions from $p(a_t^j \triangleq [\mathbf{g}_1, \mathbf{g}_j]) = \mathbf{V}_{rule}(t+1, j)/(\mathbf{V}_{rule}(t+1, 2) + \mathbf{V}_{rule}(t+1, 1)), \forall j = 1, 2.$

- **Value-Iter:** It dynamically updates the value table based on policy evaluation [186]. The allocation policy is computed based on the new value table, the same used in the rule-based method, while the collaborative context is considered.

- **T-Q learning**: The standard independent tabular Q-learning [186] learns a table $\mathbf{q}_{tabular} \in \mathbb{R}^{T \times N \times 7}$ with ϵ-greedy policy. In this case the state reduces to time and the location of the agent.

- **T-SARSA**: The independent tabular SARSA [186] learns a table $\mathbf{q}_{sarsa} \in \mathbb{R}^{T \times N \times 7}$ with same setting of states as T-Q learning.

- **DQN**: The independent DQN is currently the state-of-the-art as we introduced in Sec 5.4.1. Our Q network is parameterized by a three-layer ELUs [41] and we adopt the ϵ-greedy policy as the agent policy. The ϵ is annealed linearly from 0.5 to 0.1 across the first 15 training episodes and fixed as $\epsilon = 0.1$ during the testing.

- **cDQN**: The contextual DQN as we introduced in Sec 5.4.2. The ϵ is annealed the same as in DQN. At the end of each episode, the Q-network is updated over 4000 batches, i.e. $M_1 = 4000$ in Alg 5.2. To ensure a valid context masking, the activation function of the output layer of the Q-network is ReLU + 1.

- **cA2C**: The contextual multi-agent actor-critic as we introduced in Sec 5.4.3. At the end of each episode, both the policy network and the value network are updated over 4000 batches, i.e. $M_1 = M_2 = 4000$ in Alg 5.2. Similar to cDQN, The output layer of the policy network uses ReLU + 1 as the activation function to ensure that all elements in the original logits $\mathbf{P}(\mathbf{s}_t^i)$ are positive.

- **LP-cA2C**: The contextual multi-agent actor-critic with linear programming as introduced in Sec 5.5. During the training state, we use cA2C to explore the environment and learn the state value function. During the evaluation, we conduct the policy given by linear programming.

Except for the first baseline, the geographic context is considered in all methods so that the agents will not navigate to the invalid grid. Unless other specified, the value function approximations and policy network in contextual algorithms are parameterized by a three-layer ReLU [78] with node sizes of 128, 64 and 32, from the first layer to the third layer. The batch size of all deep learning methods is fixed as 3000, and we use AdamOptimizer with a learning rate of $1e-3$. Since performance of DQN varies a lot when there are a large number

of agents, the first column in the Table 5.1 for DQN is averaged over the best three runs out of six runs, and the results for all other methods are averaged over three runs. Also, the centralized critics of cDQN and cA2C are initialized from a pre-trained value network using the historical mean of order values computed from ten episodes simulation, with different random seeds from both training and evaluation.

To test the robustness of proposed method, we evaluate all competing methods under different numbers of initial vehicles accross different cities. The results are summarized in Table 5.1, 5.2, 5.3. The results of *Diffusion* improved the performance a lot in Table 5.1, possibly because that the method sometimes encourages the available vehicles to leave the grid with high density of available vehicles, and thus the imbalanced situation is alleviated. However, in a more realistic setting that we consider reposition cost, this method can lead to negative effective due to the highly inefficient reallocations. The *Rule-based* method that repositions vehicles to the grids with a higher demand value, improves the performance of random repositions. The *Value-Iter* dynamically updates the value table according to the current policy applied so that it further promotes the performance upon *Rule-based*. Comparing the results of *Value-Iter*, *T-Q learning* and *T-SARSA*, the first method consistently outperforms the latter two, possibly because that the usage of a centralized value table enables coordinations, which helps to avoid conflict repositions. The above methods simplify the state representation into a spatial-temporal value representation, whereas the DRL methods account both complex dynamics of supply and demand using neural network function approximations. As the results shown in last three rows of Table 5.1, 5.2, 5.3, the methods with deep learning outperforms the previous one. Furthermore, the contextual algorithms largely outperform the independent DQN (DQN), which is the state-of-the-art among large-scale multi-agent DRL method and all other competing methods. Last but not least, the lp-cA2C acheive the

Table 5.1: Performance comparison of competing methods in terms of GMV and order response rate without reposition cost.

	100% initial vehicles		90% initial vehicles		10% initial vehicles	
	Normalized GMV	ORR	Normalized GMV	ORR	Normalized GMV	ORR
Simulation	100.00 ± 0.60	$81.80\% \pm 0.37\%$	98.81 ± 0.50	$80.64\% \pm 0.37\%$	92.78 ± 0.79	$70.29\% \pm 0.64\%$
Diffusion	105.68 ± 0.64	$86.48\% \pm 0.54\%$	104.44 ± 0.57	$84.93\% \pm 0.49\%$	99.00 ± 0.51	$74.51\% \pm 0.28\%$
Rule-based	108.49 ± 0.40	$90.19\% \pm 0.33\%$	107.38 ± 0.55	$88.70\% \pm 0.48\%$	100.08 ± 0.50	$75.58\% \pm 0.36\%$
Value-Iter	110.29 ± 0.70	$90.14\% \pm 0.62\%$	109.50 ± 0.68	$89.59\% \pm 0.69\%$	102.60 ± 0.61	$77.17\% \pm 0.53\%$
T-Q learning	108.78 ± 0.51	$90.06\% \pm 0.38\%$	107.71 ± 0.42	$89.11\% \pm 0.42\%$	100.07 ± 0.55	$75.57\% \pm 0.40\%$
T-SARSA	109.12 ± 0.49	$90.18\% \pm 0.38\%$	107.69 ± 0.49	$88.68\% \pm 0.42\%$	99.83 ± 0.50	$75.40\% \pm 0.44\%$
DQN	114.06 ± 0.66	$93.01\% \pm 0.20\%$	113.19 ± 0.60	$91.99\% \pm 0.30\%$	103.80 ± 0.96	$77.03\% \pm 0.23\%$
cDQN	115.19 ± 0.46	$94.77\% \pm 0.32\%$	**114.29** ± 0.66	**94.00%** $\pm 0.53\%$	105.29 ± 0.70	$79.28\% \pm 0.58\%$
cA2C	**115.27** ± 0.70	**94.99%** $\pm 0.48\%$	113.85 ± 0.69	$93.99\% \pm 0.47\%$	**105.62** ± 0.66	**79.57%** $\pm 0.51\%$

Table 5.2: Performance comparison of competing methods in terms of GMV, order response rate (ORR), and return on invest (ROI) in Xian considering reposition cost.

	100% initial vehicles			90% initial vehicles			10% initial vehicles		
	Normalized GMV	ORR	ROI	Normalized GMV	ORR	ROI	Normalized GMV	ORR	ROI
Simulation	100.00 ± 0.60	$81.80\% \pm 0.37\%$	-	98.81 ± 0.50	$80.64\% \pm 0.37\%$	-	92.78 ± 0.79	$70.29\% \pm 0.64\%$	-
Diffusion	103.02 ± 0.41	$86.49\% \pm 0.42\%$	0.5890	102.35 ± 0.51	$85.00\% \pm 0.47\%$	0.7856	97.41 ± 0.55	$74.51\% \pm 0.46\%$	1.5600
Rule-based	106.21 ± 0.43	$90.00\% \pm 0.43\%$	1.4868	105.30 ± 0.42	$88.58\% \pm 0.37\%$	1.7983	99.37 ± 0.36	$75.83\% \pm 0.48\%$	3.2829
Value-Iter	108.26 ± 0.65	$90.28\% \pm 0.50\%$	2.0092	107.69 ± 0.82	$89.53\% \pm 0.56\%$	2.5776	101.56 ± 0.65	$77.11\% \pm 0.44\%$	4.5251
T-Q learning	107.55 ± 0.58	$90.12\% \pm 0.52\%$	2.9201	106.60 ± 0.52	$89.17\% \pm 0.41\%$	4.2052	99.99 ± 1.28	$75.97\% \pm 0.91\%$	5.2527
T-SARSA	107.73 ± 0.46	$89.93\% \pm 0.34\%$	3.3881	106.88 ± 0.45	$88.82\% \pm 0.37\%$	5.1559	99.11 ± 0.40	$75.23\% \pm 0.35\%$	6.8805
DQN	110.81 ± 0.68	$92.50\% \pm 0.50\%$	1.7811	110.16 ± 0.60	$91.79\% \pm 0.29\%$	2.3790	103.40 ± 0.51	$77.14\% \pm 0.26\%$	4.3770
cDQN	112.49 ± 0.42	$94.88\% \pm 0.33\%$	2.2207	112.12 ± 0.40	$94.17\% \pm 0.36\%$	2.7708	104.25 ± 0.55	$79.41\% \pm 0.48\%$	4.8340
cA2C	112.70 ± 0.64	$94.74\% \pm 0.57\%$	3.1062	112.05 ± 0.45	$93.97\% \pm 0.37\%$	3.8085	104.19 ± 0.70	$79.25\% \pm 0.68\%$	5.2124
LP-cA2C	**113.60** ± 0.56	**95.27%** $\pm 0.36\%$	**4.4633**	**112.75** ± 0.65	**94.62%** $\pm 0.47\%$	**5.2719**	**105.37** ± 0.58	**80.15%** $\pm 0.46\%$	**7.2949**

best performance in terms of return on investment (the gmv gain per reallocation), GMV, and order response rate.

5.7.3 On the Efficiency of Reallocations

In reality, each reposition comes with a cost. In this subsection, we consider such reposition costs and estimated them by fuel costs. Since the travel distance from one grid to another is approximately 1.2km and the fuel cost is around 0.5 RMB/km, we set the cost of each reposition as $c = 0.6$. In this setting, the definition of agent, state, action and transition probability is same as we stated in Sec 5.3. The only difference is that the repositioning cost is included in the reward when the agent is repositioned to different locations. Therefore, the GMV of one episode is the sum of all served order value substracted by the total of reposition cost in one episode. For example, the objective function for DQN now includes the reposition

Table 5.3: Performance comparison of competing methods in terms of GMV, order response rate (ORR), and return on invest (ROI) in Wuhan considering reposition cost.

	Normalized GMV	ORR	ROI
Simulation	100.00 ± 0.48	$76.56\% \pm 0.45\%$	-
Diffusion	98.84 ± 0.44	$80.07\% \pm 0.24\%$	-0.2181
Rule-based	103.84 ± 0.63	$84.91\% \pm 0.25\%$	0.5980
Value-Iter	107.13 ± 0.70	$85.06\% \pm 0.45\%$	1.6156
T-Q learning	107.10 ± 0.61	$85.28\% \pm 0.28\%$	1.8302
T-SARSA	107.14 ± 0.64	$84.99\% \pm 0.28\%$	2.0993
DQN	108.45 ± 0.62	$86.67\% \pm 0.33\%$	1.0747
cDQN	108.93 ± 0.57	$89.03\% \pm 0.26\%$	1.1001
cA2C	113.31 ± 0.54	$88.57\% \pm 0.45\%$	4.4163
LP-cA2C	**114.92** ± 0.65	**89.29%** $\pm 0.39\%$	**6.1417**

Table 5.4: Effectiveness of contextual multi-agent actor-critic considering reposition costs.

	Normalized GMV	ORR	Repositions
DQN	110.81 ± 0.68	$92.50\% \pm 0.50\%$	606932
cDQN	112.49 ± 0.42	$94.88\% \pm 0.33\%$	562427
cA2C	112.70 ± 0.64	$94.74\% \pm 0.57\%$	408859
LP-cA2C	113.60 ± 0.56	$95.27\% \pm 0.36\%$	304752

cost as follows:

$$\mathbb{E}\left[Q(\mathbf{s}_t^i, a_t^i; \theta) - \left(r_{t+1}^i - c + \gamma \max_{a_{t+1}^i} Q(\mathbf{s}_{t+1}^i, a_{t+1}^i; \theta')\right)\right]^2, \tag{5.15}$$

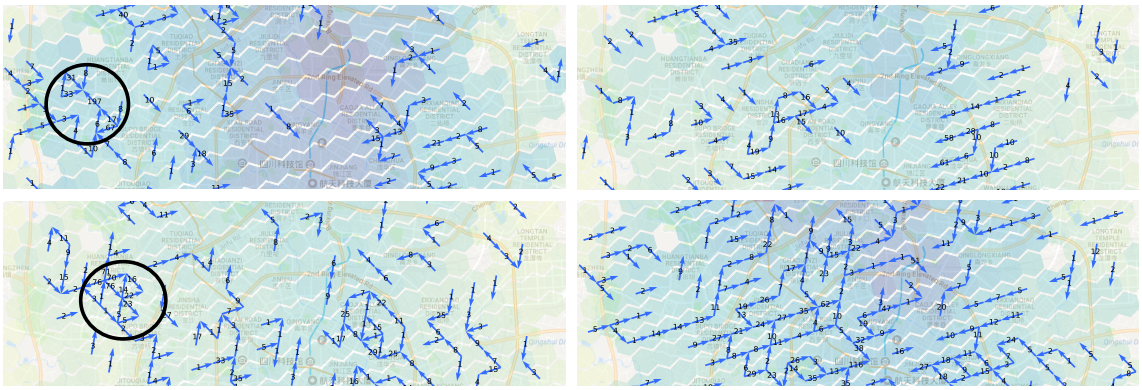
where $a_t^i \triangleq [\mathbf{g}_o, \mathbf{g}_d]$, and if $\mathbf{g}_d = \mathbf{g}_o$ then $c = 0$, otherwise $c = 0.6$. Similarly, we can consider the costs in cA2C. However, it is hard to apply them to cDQN because that the assumption, that different actions that lead to the same location should share the same action value, which is not held in this setting. Therefore, instead of considering the reposition cost in the objective function, we only incorporate the reposition cost when we actually conduct our policy based on cDQN. Under this setting, the learning objective of action value of cDQN is

same as in Eq (5.3) while the context embedding is changed from Eq (5.4) to the following:

$$[\mathbf{C}_{t,\mathbf{g}_j}]_k = \begin{cases} 1, & \text{if } Q(\mathbf{s}_t, \mathbf{g}_i) >= Q(\mathbf{s}_t, \mathbf{g}_j) + c, \\ 0, & \text{otherwise.} \end{cases} \tag{5.16}$$

For LP-cA2C, the cost effect is naturally incorporated in the objective function as in Eq (5.14). As the results shown in Table 5.4, the DQN tends to reposition more agents while the contextual algorithms achieve better performance in terms of both GMV and order response rate, with lower cost. More importantly, the LP-cA2C outperforms other methods in both of the performance and efficiency. The reason is that this method formulate the coordination among agents into an optimization problem, which approximates the maximization of the platform's long term reward in a centralized version. The centralized optimization problem can avoid lots of redundant reallocations compared to previous methods. The training procedures and the network architecture are the same as described in the previous section.

To be more concrete, we give a specific scenario to demonstrate that the efficiency of LP-cA2C. Imaging we would like to ask drivers to move from grid A to nearby grid B while there is a grid C that is adjacent to both grid A and B. In the previous algorithms, since the allocation is jointly given by each agent, it's very likely that we reallocate agents by the short path $A \to B$ and longer path $A \to C \to B$ when there are sufficient amount of agents can arrive at B from A. These inefficient reallocations can be avoided by LP-cA2C naturally since the longer path only incurs a higher cost which will be the suboptimal solution to our objective function compared to the solution only contains the first path. As shown in Figure 5.5 (a), the allocation computed by cA2C contains many *triangle* repositions as denoted by the black circle, while we didn't observe these inefficient allocations in Figure 5.5



(a) cA2C (b) LP-cA2C

Figure 5.5: Illustration of allocations of cA2C and LP-cA2C at 18:40 and 19:40, respsectively.

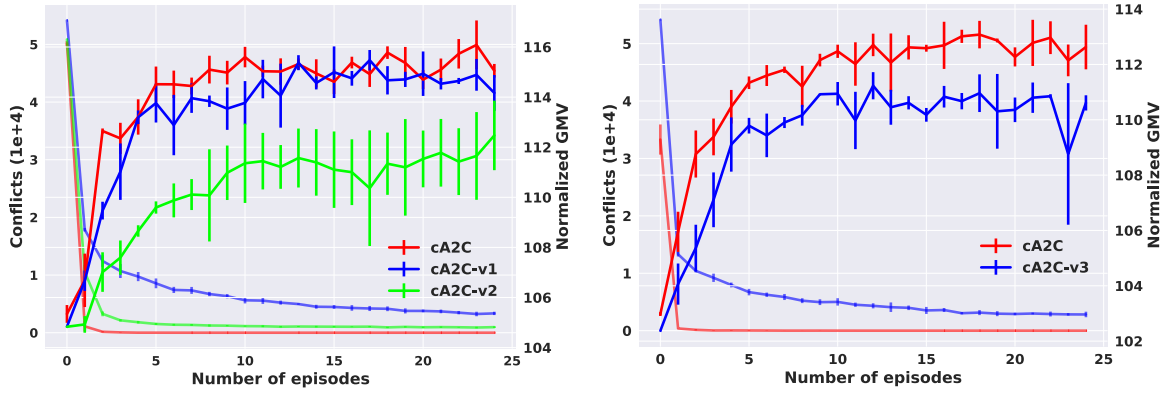
(b). Therefore, the allocation policy delivered by LP-cA2C is more efficient than those given by previous algorithms.

5.7.4 The effectiveness of averaged reward design

In multi-agent RL, the reward design for each agent is essential for the success of learning. In fully cooperative multi-agent RL, the reward for all agents is a single global reward [27], while it suffers from the credit assignment problem for each agent's action. Splitting the reward to each agent will alleviate this problem. In this subsection, we compare two different designs for the reward of each agent: the averaged reward of a grid as stated in Sec 5.3 and the total reward of a grid that does not average on the number of available vehicles at that time. As shown in table 5.5, the methods with averaged reward (cA2C, cDQN) largely outperform those using total reward, since this design naturally encourages the coordinations among agents. Using total reward, on the other hand, is likely to reposition an excessive number of agents to the location with high demand.

Table 5.5: Effectiveness of averaged reward design.

	Proposed methods	Raw Reward
	Normalized GMV/ORR	Normalized GMV/ORR
cA2C	115.27±0.70/94.99% ± 0.48%	105.75 ± 1.17/88.09% ± 0.74%
cDQN	115.19 ± 0.46/94.77% ± 0.32%	108.00 ± 0.35/89.53% ± 0.31%



(a) Without reposition cost (b) With reposition cost

Figure 5.6: Convergence comparison of cA2C and its variations without using context embedding in both settings, with and without reposition costs. The X-axis is the number of episodes. The left Y-axis denotes the number of conflicts and the right Y-axis denotes the normalized GMV in one episode.

Table 5.6: Effectiveness of context embedding.

	Normalized GMV/ORR	Repositions
	Without reposition cost	
cA2C	115.27 ± 0.70/94.99% ± 0.48%	460586
cA2C-v1	114.78 ± 0.67/94.52% ± 0.49%	704568
cA2C-v2	111.39 ± 1.65/92.12% ± 1.03%	846880
	With reposition cost	
cA2C	112.70 ± 0.64/94.74% ± 0.57%	408859
cA2C-v3	110.43 ± 1.16/93.79% ± 0.75%	593796

5.7.5 Ablations on policy context embedding

In this subsection, we evaluate the effectiveness of context embedding, including explicitly coordinating the actions of different agents through the collaborative context, and eliminating the invalid actions with geographic context. The following variations of proposed methods are investigated in different settings.

- cA2C-v1: This variation drops collaborative context of cA2C in the setting that does not consider reposition cost.
- cA2C-v2: This variation drops both geographic and collaborative context of cA2C in the setting that does not consider reposition cost.
- cA2C-v3: This variation drops collaborative context of cA2C in the setting that considers reposition cost.

The results of above variations are summarized in Table 5.6 and Figure 5.6. As seen in the first two rows of Table 5.6 and the red/blue curves in Figure 5.6 (a), in the setting of zero reposition cost, cA2C achieves the best performance with much less repositions (65.37%) comparing with cA2C-v1. Furthermore, collaborative context embedding achieves significant advantages when the reposition cost is considered, as shown in the last two rows in Table 5.6 and Figure 5.6 (b). It not only greatly improves the performance but also accelerates the convergence. Since the collaborative context largely narrows down the action space and leads to a better policy solution in the sense of both effectiveness and efficiency, we can conclude that coordination based on collaborative context is effective. Also, comparing the performances of cA2C and cA2C-v2 (red/green curves in Figure 5.6 (a)), apparently the policy context embedding (considering both geographic and collaborative context) is essential to performance, which greatly reduces the redundant policy search.

5.7.6 Ablation study on grouping the locations

This section studies the effectiveness of our regularization design for LP-cA2C. One key difference between our work and traditional fleet management works [64, 65] is that we didn't assume the drivers in one location can only pick up the orders in the same location. On

the contrary, one agent can also serve the orders emerged in the nearby locations, which is a more realistic and complicated setting. In this case, we regularize the number of agents repositioned into a set of nearby grids close to the number of estimated orders at next time step. This grouping regularization in Eq (5.14) is more efficient than the regularization in Eq (5.17) requiring the number of agents repositioned into each grid is close to the number of estimated orders at that gird since lots of reposition inside the same group can be avoided. As the results shown in Table 5.7, using the group regularization in Eq (5.14) reallocates less agents while achieves same best performance as the one in Eq (5.17) (LP-cA2C').

$$\max_{\mathbf{y}(\mathbf{s}_t)} \quad (\mathbf{v}(\mathbf{s}_t)^T \mathbf{A}_t - \mathbf{c}_t^T)\mathbf{y}(\mathbf{s}_t) - \lambda(\mathbf{o}_t - \mathbf{A}_t\mathbf{y}(\mathbf{s}_t))^2 \tag{5.17}$$

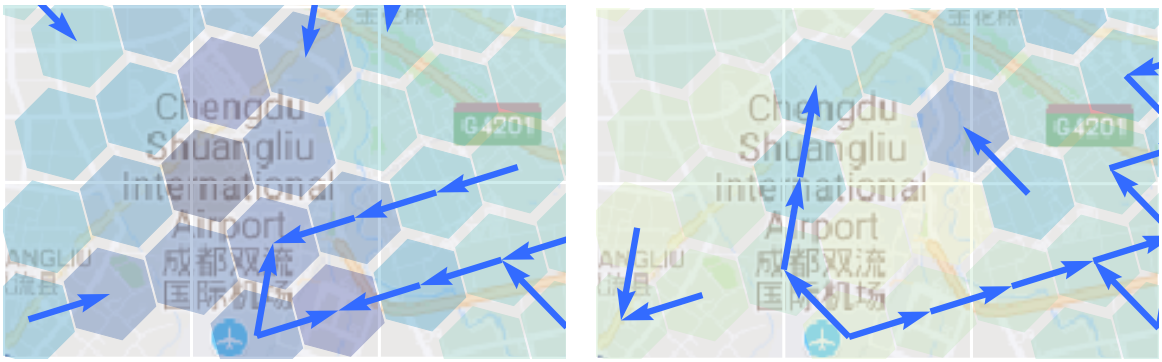
Table 5.7: Effectiveness of group regularization design

	Normalized GMV	ORR	Repositions	ROI
LP-cA2C	113.56 ± 0.61	$95.24\% \pm 0.40\%$	341774	3.9663
LP-cA2C'	113.60 ± 0.56	$95.27\% \pm 0.36\%$	304752	4.4633

5.7.7 Qualitative study

In this section, we analyze whether the learned value function can capture the demand-supply relation ahead of time, and the rationality of allocations. To see this, we present a case study on the region nearby the airport. The state value and allocation policy is acquired from cA2C that was trained for ten episodes. We then run the well-trained cA2C on one testing episode, and qualitatively exam the state value and allocations under the unseen dynamics. The sum of state values and demand-supply gap (defined as the number of orders minus the number of vehicles) of seven grids that cover the CTU airport is visualized. As seen in Figure 5.8, the state value can capture the future dramatic changes of demand-supply gap. Furthermore, the

spatial distribution of state values can be seen in Figure 5.7. After the midnight, the airport has a large number of orders, and less available vehicles, and therefore the state values of airport are higher than other locations. During the daytime, more vehicles are available at the airport so that each will receive less reward and the state values are lower than other regions, as shown in Figure 5.7 (b). In Figure 5.7 and Figure 5.8, we can conclude that the value function can estimate the relative shift of demand-supply gap from both spatial and temporal perspectives. It is crucial to the performance of cA2C since the coordination is built upon the state values. Moreover, as illustrated by blue arrows in Figure 5.7, we see that the allocation policy gives consecutive allocations from lower value grids to higher value grids, which can thus fill the future demand-supply gap and increase the GMV.



(a) At 01:50 am. (b) At 06:40 pm.

Figure 5.7: Illustration on the repositions nearby the airport at 1:50 am and 06:40 pm. The darker color denotes the higher state value and the blue arrows denote the repositions.

5.8 Conclusion

In this chapter, we first formulate the large-scale fleet management problem into a feasible setting for deep reinforcement learning. Given this setting, we propose contextual multi-agent reinforcement learning framework, in which two contextual algorithms cDQN and cA2C are

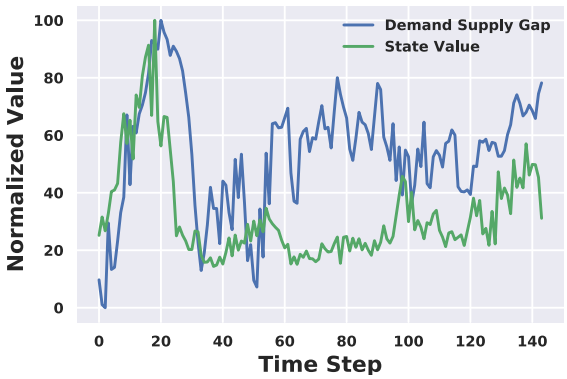


Figure 5.8: The normalized state value and demand-supply gap over one day.

developed and both of them achieve the large scale agents' coordination in fleet management problem. cA2C enjoys both flexibility and efficiency by capitalizing a centralized value network and decentralized policy execution embedded with contextual information. It is able to adapt to different action space in an end-to-end training paradigm. A simulator is developed and calibrated with the real data provided by Didi Chuxing, which served as our training and evaluation platform. Extensive empirical studies under different settings in simulator have demonstrated the effectiveness of the proposed framework.

Chapter 6

The Provable Advantage of Collaborative Learning

6.1 Introduction

Federated learning (FL) is a machine learning setting where many clients (e.g., mobile devices or organizations) collaboratively train a model under the orchestration of a central server (e.g., service provider), while keeping the training data decentralized [176, 94]. In recent years, FL has swiftly emerged as an important learning paradigm [129, 109]–one that enjoys widespread success in applications such as personalized recommendation [36], virtual assistant [106], and keyboard prediction [77], to name a few–for at least two reasons: First, the rapid proliferation of smart devices that are equipped with both computing power and data-capturing capabilities provided the infrastructure core for FL. Second, the rising awareness of privacy and the exponential growth of computational power (blessed by Moore's law) in mobile devices have made it increasingly attractive to push the computation to the edge.

Despite its promise and broad applicability in our current era, the potential value FL delivers is coupled with the unique challenges it brings forth. In particular, when FL learns a single statistical model using data from across all the devices while keeping each individual device's data isolated (and hence protects privacy) [94], it faces two challenges that are absent

in centralized optimization and distributed (stochastic) optimization [231, 178, 99, 113, 205, 213, 206, 92, 219, 218, 98, 104]:

1) **Data heterogeneity:** data distributions in devices are different (and data can't be shared);

2) **System heterogeneity:** only a subset of devices may access the central server at each time both because the communications bandwidth profiles vary across devices and because there is no central server that has control over when a device is active.

To address these challenges, Federated Averaging (FedAvg) [129] was proposed as a particularly effective heuristic, which has enjoyed great empirical success [77]. This success has since motivated a growing line of research efforts into understanding its theoretical convergence guarantees in various settings. For instance, [75] analyzed FedAvg (for non-convex smooth problems satisfying PL conditions) under the assumption that each local device's minimizer is the same as the minimizer of the joint problem (if all devices' data is aggregated together), an overly restrictive assumption. Very recently, [110] furthered the progress and established an $\mathcal{O}(\frac{1}{T})$ convergence rate for FedAvg for strongly convex smooth problems. At the same time, [84] studied the Nesterov accelerated FedAvg for non-convex smooth problems and established an $\mathcal{O}(\frac{1}{\sqrt{T}})$ convergence rate to stationary points.

However, despite these very recent fruitful pioneering efforts into understanding the theoretical convergence properties of FedAvg, it remains open as to how the number of devices–particularly the number of devices that participate in the computation–affects the convergence speed. In particular, do we get linear speedup of FedAvg? What about when FedAvg is accelerated? These aspects are currently unexplored in FL. We fill in the gaps here by providing affirmative answers.

Our Contributions We provide a comprehensive convergence analysis of FedAvg and its

Table 6.1: Convergence results for FedAvg and accelerated FedAvg. Throughout the paper, N is the total number of local devices, and $K \leq N$ is the maximal number of devices that are accessible to the central server. T is the total number of stochastic updates performed by each local device, E is the local steps between two consecutive server communications (and hence T/E is the number of communications). † In the linear regression setting, we have $\kappa = \kappa_1$ for FedAvg and $\kappa = \sqrt{\kappa_1 \tilde{\kappa}}$ for accelerated FedAvg, where κ_1 and $\sqrt{\kappa_1 \tilde{\kappa}}$ are condition numbers defined in Section 6.5. Since $\kappa_1 \geq \tilde{\kappa}$, this implies a speedup factor of $\sqrt{\frac{\kappa_1}{\tilde{\kappa}}}$ for accelerated FedAvg.

Objective function / Participation	Strongly Convex	Convex	Overparameterized general case	Overparameterized linear regression
Full	$\mathcal{O}(\frac{1}{NT} + \frac{E^2}{T^2})$	$\mathcal{O}\left(\frac{1}{\sqrt{NT}} + \frac{NE^2}{T}\right)$	$\mathcal{O}(\exp(-\frac{NT}{E\kappa_1}))$	$\mathcal{O}(\exp(-\frac{NT}{E\kappa}))^\dagger$
Partial	$\mathcal{O}\left(\frac{E^2}{KT} + \frac{E^2}{T^2}\right)$	$\mathcal{O}\left(\frac{E^2}{\sqrt{KT}} + \frac{KE^2}{T}\right)$	$\mathcal{O}(\exp(-\frac{KT}{E\kappa_1}))$	$\mathcal{O}(\exp(-\frac{KT}{E\kappa}))^\dagger$

accelerated variants in the presence of both data and system heterogeneity. Our contributions are threefold.

First, we establish an $\mathcal{O}(1/KT)$ convergence rate under FedAvg for strongly convex and smooth problems and an $\mathcal{O}(1/\sqrt{KT})$ convergence rate for convex and smooth problems (where K is the number of participating devices), thereby establishing that FedAvg enjoys the desirable linear speedup property in the FL setup. Prior to our work here, the best and the most related convergence analysis is given by [110], which established an $\mathcal{O}(\frac{1}{T})$ convergence rate for strongly convex smooth problems under FedAvg. Our rate matches the same (and optimal) dependence on T, but also completes the picture by establishing the linear dependence on K.

Second, we establish the same convergence rates–$\mathcal{O}(1/KT)$ for strongly convex and smooth problems and $\mathcal{O}(1/\sqrt{KT})$ for convex and smooth problems–for Nesterov accelerated FedAvg. We analyze the accelerated version of FedAvg here because empirically it tends to perform better; yet, its theoretical convergence guarantee is unknown. To the best of our knowledge, these are the first results that provide a linear speedup characterization of Nesterov accelerated FedAvg in those two problem classes (that FedAvg and Nesterov accelerated FedAvg share the same convergence rate is to be expected: this is the case even for centralized stochastic

optimization).

Third, we study a subclass of strongly convex smooth problems where the objective is over-parameterized and establish a faster $\mathcal{O}(\exp(-\frac{KT}{\kappa}))$ convergence rate for FedAvg. Within this class, we further consider the linear regression problem and establish an even sharper rate under FedAvg. In addition, we propose a new variant of accelerated FedAvg–MaSS accelerated FedAvg–and establish a faster convergence rate (compared to if no acceleration is used). This stands in contrast to generic (strongly) convex stochastic problems where theoretically no rate improvement is obtained when one accelerates FedAvg. The detailed convergence results are summarized in Table 6.1.

6.2 Setup

In this chapter, we study the following federated learning problem:

$$\min_{\mathbf{w}} \left\{ F(\mathbf{w}) \triangleq \sum\nolimits_{k=1}^{N} p_k F_k(\mathbf{w}) \right\}, \tag{6.1}$$

where N is the number of local devices (users/nodes/workers) and p_k is the k-th device's weight satisfying $p_k \geq 0$ and $\sum_{k=1}^{N} p_k = 1$. In the k-th local device, there are n_k data points: $\mathbf{x}_k^1, \mathbf{x}_k^2, \ldots, \mathbf{x}_k^{n_k}$. The local objective $F_k(\cdot)$ is defined as: $F_k(\mathbf{w}) \triangleq \frac{1}{n_k} \sum_{j=1}^{n_k} \ell\left(\mathbf{w}; \mathbf{x}_k^j\right)$, where ℓ denotes a user-specified loss function. Each device only has access to its local data, which gives rise to its own local objective F_k. Note that we do not make any assumptions on the data distributions of each local device. The local minimum $F_k^* = \min_{\mathbf{w} \in R^d} F_k(\mathbf{w})$ can be far from the global minimum of Eq (6.1).

6.2.1 The Federated Averaging (FedAvg) Algorithm

We first introduce the standard Federated Averaging (FedAvg) algorithm [129]. FedAvg updates the model in each device by local Stochastic Gradient Descent (SGD) and sends the latest model to the central server every E steps. The central server conducts a weighted average over the model parameters received from active devices and broadcasts the latest averaged model to all devices. Formally, the updates of FedAvg at round t is described as follows:

$$\mathbf{v}_{t+1}^k = \mathbf{w}_t^k - \alpha_t \mathbf{g}_{t,k}, \quad \mathbf{w}_{t+1}^k = \begin{cases} \mathbf{v}_{t+1}^k & \text{if } t+1 \notin \mathcal{I}_E, \\ \sum_{k \in \mathcal{S}_{t+1}} \mathbf{v}_{t+1}^k & \text{if } t+1 \in \mathcal{I}_E, \end{cases}$$

where $\mathbf{w}_t^k$ is the local model parameter maintained in the k-th device at the t-th iteration, $\mathbf{g}_{t,k} := \nabla F_k(\mathbf{w}_t^k, \xi_t^k)$ is the stochastic gradient based on ξ_t^k, the data sampled from k-th device's local data uniformly at random. $\mathcal{I}_E = \{E, 2E, \dots\}$ is the set of global communication steps. We use $\mathcal{S}_{t+1}$ to represent the set of active devices at $t+1$.

Since federated learning usually involves an enormous amount of local devices, it is often more realistic to assume only a subset of local devices is active at each communication round (system heterogeneity). In this work, we consider both the case of **full participation** where the model is averaging over all devices at the communication round, i.e., $\mathbf{w}_{t+1}^k = \sum_{k=1}^N p_k \mathbf{v}_{t+1}^k$, and the case of **partial participation** where $|\mathcal{S}_{t+1}| < N$. With partial participation, $\mathcal{S}_{t+1}$ is obtained by two types of sampling schemes to simulate practical scenarios [110]. For example, one scheme establishes $\mathcal{S}_{t+1}$ by *i.i.d.* sampling the devices with probability p_k with replacement. Both schemes guarantee that gradient updates in FedAvg are unbiased stochastic versions of updates in FedAvg with full participation. For more details on the

notations and setup, please refer to Section B in the appendix.

6.2.2 Assumptions

We make the following standard assumptions on the objective function $F_1, \ldots, F_N$. Assumptions 7 and 8 are commonly satisfied by a range of popular objective functions, such as ℓ^2-regularized logistic regression and cross-entropy loss functions.

Assumption 7 (L-smooth)**.** *$F_1, \cdots, F_N$ are all L-smooth: for all $\mathbf{v}$ and $\mathbf{w}$, $F_k(\mathbf{v}) \leq F_k(\mathbf{w}) + (\mathbf{v}-\mathbf{w})^T \nabla F_k(\mathbf{w}) + \frac{L}{2}\|\mathbf{v}-\mathbf{w}\|_2^2$.*

Assumption 8 (Strongly-convex)**.** *$F_1, \cdots, F_N$ are all μ -strongly convex: for all v and $\mathbf{w}$, $F_k(\mathbf{v}) \geq F_k(\mathbf{w}) + (\mathbf{v}-\mathbf{w})^T \nabla F_k(\mathbf{w}) + \frac{\mu}{2}\|\mathbf{v}-\mathbf{w}\|_2^2$*

Assumption 9 (Bounded local variance)**.** *Let ξ_t^k be sampled from the k-th device's local data uniformly at random. The variance of stochastic gradients in each device is bounded: $\mathbb{E}\left\|\nabla F_k\left(\mathbf{w}_t^k, \xi_t^k\right) - \nabla F_k\left(\mathbf{w}_t^k\right)\right\|^2 \leq \sigma_k^2$, for $k = 1, \cdots, N$ and any $\mathbf{w}_t^k$. Let $\sigma^2 = \sum_{k=1}^N p_k \sigma_k^2$.*

Assumption 10 (Bounded local gradient)**.** *The expected squared norm of stochastic gradients is uniformly bounded. i.e., $\mathbb{E}\left\|\nabla F_k\left(\mathbf{w}_t^k, \xi_t^k\right)\right\|^2 \leq G^2$, for all $k = 1, ..., N$ and $t = 0, \ldots, T-1$.*

Assumptions 9 and 10 have been made in many previous works in federated learning, e.g. [219, 110, 178]. We provide further justification for their generality. As model average parameters become closer to $\mathbf{w}^*$, the L-smoothness property implies that $\mathbb{E}\|\nabla F_k(\mathbf{w}_t^k, \xi_t^k)\|^2$ and $\mathbb{E}\|\nabla F_k(\mathbf{w}_t^k, \xi_t^k) - \nabla F_k(\mathbf{w}_t^k)\|^2$ approach $\mathbb{E}\|\nabla F_k(\mathbf{w}^*, \xi_t^k)\|^2$ and $\mathbb{E}\|\nabla F_k(\mathbf{w}^*, \xi_t^k) - \nabla F_k(\mathbf{w}^*)\|^2$. Therefore, there is no substantial difference between these assumptions and assuming the

bounds at $\mathbf{w}^*$ only. Furthermore, compared to assuming *bounded gradient diversity* as in related work [75, 109], Assumption 10 is much less restrictive. When the optimality gap converges to zero, bounded gradient diversity restricts local objectives to have the same minimizer as the global objective, contradicting the heterogeneous data setting. For detailed discussions of our assumptions, please refer to Appendix Section B.

6.3 Linear Speedup Analysis of FedAvg

In this section, we provide convergence analyses of FedAvg for convex objectives in the general setting with both heterogeneous data and partial participation. We show that for strongly convex and smooth objectives, the convergence of the optimality gap of averaged parameters across devices is $\mathcal{O}(1/NT)$, while for convex and smooth objectives, the rate is $\mathcal{O}(1/\sqrt{NT})$. Detailed proofs are deferred to Appendix Section B.

6.3.1 Strongly Convex and Smooth Objectives

We first show that FedAvg has an $\mathcal{O}(1/NT)$ convergence rate for μ-strongly convex and L-smooth objectives. The result improves on the $\mathcal{O}(1/T)$ rate of [110] with a linear speedup in the number of devices N. Moreover, it implies a distinction in communication efficiency that guarantees this linear speedup for FedAvg with full and partial device participation. With full participation, E can be chosen as large as $\mathcal{O}(\sqrt{T/N})$ without degrading the linear speedup in the number of workers. On the other hand, with partial participation, E must be $\mathcal{O}(1)$ to guarantee $\mathcal{O}(1/NT)$ convergence.

Theorem 7. *Let* $\overline{\mathbf{w}}_T = \sum_{k=1}^N p_k \mathbf{w}_T^k$, $\nu_{\max} = \max_k Np_k$, *and set decaying learning rates* $\alpha_t = \frac{1}{4\mu(\gamma+t)}$ *with* $\gamma = \max\{32\kappa, E\}$ *and* $\kappa = \frac{L}{\mu}$. *Then under Assumptions 7 to 10 with full*

device participation,

$$\mathbb{E}F(\overline{\mathbf{w}}_T) - F^* = \mathcal{O}\left(\frac{\kappa\nu_{\max}^2\sigma^2/\mu}{NT} + \frac{\kappa^2E^2G^2/\mu}{T^2}\right),$$

and with partial device participation with at most K sampled devices at each communication round,

$$\mathbb{E}F(\overline{\mathbf{w}}_T) - F^* = \mathcal{O}\left(\frac{\kappa E^2G^2/\mu}{KT} + \frac{\kappa\nu_{\max}^2\sigma^2/\mu}{NT} + \frac{\kappa^2E^2G^2/\mu}{T^2}\right).$$

Linear speedup. We first compare our bound with that in [110], which is $\mathcal{O}(\frac{1}{NT} + \frac{E^2}{KT} + \frac{E^2G^2}{T})$. Because the term $\frac{E^2G^2}{T}$ is also $\mathcal{O}(1/T)$ without a dependence on N, for any choice of E their bound cannot achieve linear speedup. The improvement of our bound comes from the term $\frac{\kappa^2E^2G^2/\mu}{T^2}$, which now is $\mathcal{O}(E^2/T^2)$. As a result, all leading terms scale with $1/N$ in the full device participation setting, and with $1/K$ in the partial participation setting. This implies that in both settings, there is a *linear speedup* in the number of active workers during a communication round. We also emphasize that the reason one cannot recover the full participation bound by setting $K = N$ in the partial participation bound is due to the variance generated by sampling which depends on E.

Communication Complexity. Our bound implies a distinction in the choice of E between the full and partial participation settings. With full participation there is linear speedup $\mathcal{O}(1/NT)$ as long as $E = \mathcal{O}(\sqrt{T/N})$ since then $\mathcal{O}(E^2/T^2) = \mathcal{O}(1/NT)$ matches the leading term. This corresponds to a communication complexity of $T/E = \mathcal{O}(\sqrt{NT})$. In contrast, the bound in [110] does not allow E to scale with $\sqrt{T}$ to preserve $\mathcal{O}(1/T)$ rate, even for full participation. On the other hand, with partial participation, $\frac{\kappa E^2G^2/\mu}{KT}$ is also a leading term,

and so E must be $\mathcal{O}(1)$. In this case, our bound still yields a linear speedup in K, which is also confirmed by experiments. The requirement $E = \mathcal{O}(1)$ in partial participation likely cannot be removed for our sampling schemes, as the sampling variance is $\Omega(E^2/T^2)$ and the dependence on E is tight.

Comparison with related work. To better understand the significance of the obtained bound, we compare our rates to the best-known results in related settings. [75] proves a linear speedup $\mathcal{O}(1/NT)$ result for strongly convex and smooth objectives, with $\mathcal{O}(N^{1/3}T^{2/3})$ communication complexity with *i.i.d.* data and partial participation. However, their results build on the bounded gradient diversity assumption, which implies the existence of $\mathbf{w}^*$ that minimizes all local objectives (see discussions in Section 6.2.2), effectively removing system heterogeneity. The bound in [104] matches our bound in the full participation case, but their framework excludes partial participation [104, Proposition 1].

6.3.2 Convex Smooth Objectives

Next we provide linear speedup analyses of FedAvg with convex and smooth objectives and show that the optimality gap is $\mathcal{O}(1/\sqrt{NT})$. This result complements the strongly convex case in the previous part, as well as the non-convex smooth setting in [92, 219, 75], where $\mathcal{O}(1/\sqrt{NT})$ results are given in terms of averaged gradient norm.

Theorem 8. *Under assumptions 7,9,10 and constant learning rate* $\alpha_t = \mathcal{O}(\sqrt{\frac{N}{T}})$,

$$\min_{t \leq T} F(\overline{\mathbf{w}}_t) - F(\mathbf{w}^*) = \mathcal{O}\left(\frac{\nu_{\max}^2 \sigma^2}{\sqrt{NT}} + \frac{NE^2LG^2}{T}\right)$$

with full participation, and with partial device participation with K sampled devices at each

communication round and learning rate $\alpha_t = \mathcal{O}(\sqrt{\frac{K}{T}})$,

$$\min_{t \leq T} F(\overline{\mathbf{w}}_t) - F(\mathbf{w}^*) = \mathcal{O}\left(\frac{\nu_{\max}^2 \sigma^2}{\sqrt{KT}} + \frac{E^2 G^2}{\sqrt{KT}} + \frac{K E^2 L G^2}{T}\right).$$

Choice of E and linear speedup. With full participation, as long as $E = \mathcal{O}(T^{1/4}/N^{3/4})$, the convergence rate is $\mathcal{O}(1/\sqrt{NT})$ with $\mathcal{O}(N^{3/4}T^{3/4})$ communication rounds. In the partial participation setting, E must be $O(1)$ in order to achieve linear speedup of $\mathcal{O}(1/\sqrt{KT})$. Our result again demonstrates the difference in communication complexities between full and partial participation, and is to our knowledge the first result on linear speedup in the general federated learning setting with both heterogeneous data and partial participation for convex objectives.

The valid range of N or K for linear speedup. Given specific values of E, T, and other constants in the bound, we can solve an optimal N or K, which can serve as a valid range of the number of devices for linear speedup. For example, with partial participation, the optimal number of participated devices is $K_{\text{opt}} = \mathcal{O}\left(\sqrt{T}(\gamma_{\max}\delta^2 + G^2)/G^2 L\right)^{2/3}$, neglecting other constant coefficients. Since increasing the value of N/K larger than K_{opt} will not bring any benefit for the convergence, from the perspective of increasing the number of devices to improve convergence, the valid range of linear speedup is $[1, K_{\text{opt}}]$. On the other hand, as long as the number of devices satisfies $K = \mathcal{O}(T^{1/3})$, the linear speedup is guaranteed.

6.4 Linear Speedup Analysis of Nesterov Accelerated FedAvg

A natural extension of the FedAvg algorithm is to use momentum-based local updates instead of local SGD updates. To our knowledge, the only convergence analyses of FedAvg with momentum-based stochastic updates focus on the non-convex smooth case [84, 218, 109]. In this section, we complete the picture with $\mathcal{O}(1/NT)$ and $\mathcal{O}(1/\sqrt{NT})$ convergence results for Nesterov-accelerated FedAvg for convex objectives that match the rates from the previous section. As we know from stochastic optimization, Nesterov and other momentum updates may fail to accelerate over SGD [119, 100, 122, 221]. Therefore in Section 6.5 we will specialize to overparameterized problems where we demonstrate that a particular FedAvg variant with momentum updates is able to accelerate over the original FedAvg algorithm. Detailed proofs of convergence results in this section are deferred to Appendix Section B.

6.4.1 Strongly Convex and Smooth Objectives

The Nesterov Accelerated FedAvg algorithm follows the updates:

$$\mathbf{v}_{t+1}^k = \mathbf{w}_t^k - \alpha_t \mathbf{g}_{t,k},$$

$$\mathbf{w}_{t+1}^k = \begin{cases} \mathbf{v}_{t+1}^k + \beta_t(\mathbf{v}_{t+1}^k - \mathbf{v}_t^k) & \text{if } t+1 \notin \mathcal{I}_E, \\ \sum_{k \in \mathcal{S}_{t+1}} \left[\mathbf{v}_{t+1}^k + \beta_t(\mathbf{v}_{t+1}^k - \mathbf{v}_t^k) \right] & \text{if } t+1 \in \mathcal{I}_E, \end{cases}$$

where $\mathbf{g}_{t,k} := \nabla F_k(\mathbf{w}_t^k, \xi_t^k)$ is the stochastic gradient sampled on the k-th device at time t.

Theorem 9. *Let* $\overline{\mathbf{v}}_T = \sum_{k=1}^N p_k \mathbf{v}_T^k$ *and set learning rates* $\beta_{t-1} = \frac{3}{14(t+\gamma)(1-\frac{6}{t+\gamma})\max\{\mu,1\}}$,

$\alpha_t = \frac{6}{\mu}\frac{1}{t+\gamma}$. *Then under Assumptions 7,8,9,10 with full device participation,*

$$\mathbb{E}F(\overline{\mathbf{v}}_T) - F^* = \mathcal{O}\left(\frac{\kappa\nu_{\max}^2\sigma^2/\mu}{NT} + \frac{\kappa^2E^2G^2/\mu}{T^2}\right),$$

and with partial device participation with K sampled devices at each communication round,

$$\mathbb{E}F(\overline{\mathbf{v}}_T) - F^* = \mathcal{O}\left(\frac{\kappa\nu_{\max}^2\sigma^2/\mu}{NT} + \frac{\kappa E^2G^2/\mu}{KT} + \frac{\kappa^2E^2G^2/\mu}{T^2}\right).$$

To our knowledge, this is the first convergence result for Nesterov accelerated FedAvg in the strongly convex and smooth setting. The same discussion about linear speedup of FedAvg applies to the Nesterov accelerated variant. In particular, to achieve $\mathcal{O}(1/NT)$ linear speedup, T iterations of the algorithm require only $\mathcal{O}(\sqrt{NT})$ communication rounds with full participation.

6.4.2 Convex Smooth Objectives

We now show that the optimality gap of Nesterov Accelerated FedAvg has $\mathcal{O}(1/\sqrt{NT})$ rate. This result complements the strongly convex case in the previous part, as well as the non-convex smooth setting in [84, 218, 109], where a similar $\mathcal{O}(1/\sqrt{NT})$ rate is given in terms of averaged gradient norm.

Theorem 10. *Set learning rates $\alpha_t = \beta_t = \mathcal{O}(\sqrt{\frac{N}{T}})$. Then under Assumptions 7,9,10 Nesterov accelerated FedAvg with full device participation has rate*

$$\min_{t\leq T} F(\overline{\mathbf{v}}_t) - F^* = \mathcal{O}\left(\frac{\nu_{\max}^2\sigma^2}{\sqrt{NT}} + \frac{NE^2LG^2}{T}\right),$$

and with partial device participation with K sampled devices at each communication round,

$$\min_{t \leq T} F(\overline{\mathbf{v}}_t) - F^* = \mathcal{O}\left(\frac{\nu_{\max}^2 \sigma^2}{\sqrt{KT}} + \frac{E^2 G^2}{\sqrt{KT}} + \frac{K E^2 L G^2}{T}\right).$$

It is possible to extend the results in this section to accelerated FedAvg algorithms with other momentum-based updates. However, in the stochastic optimization setting, none of these methods can achieve a better rate than the original FedAvg with SGD updates for general problems [100]. For this reason, we will instead turn to the overparameterized setting [127, 119, 30] in the next section where we show that FedAvg enjoys geometric convergence and it is possible to improve its convergence rate with momentum-based updates.

6.5 Geometric Convergence of FedAvg in the Overparameterized Setting

Overparameterization is a prevalent machine learning setting where the statistical model has much more parameters than the number of training samples and the existence of parameter choices with zero training loss is ensured [5, 224]. Due to the property of *automatic variance reduction* in overparameterization, a line of recent works proved that SGD and accelerated methods achieve geometric convergence [127, 134, 138, 168, 181]. A natural question is whether such a result still holds in the federated learning setting. In this section, we provide the first geometric convergence rate of FedAvg for the overparameterized strongly convex and smooth problems, and show that it preserves linear speedup at the same time. We then sharpen this result in the special case of linear regression. Inspired by recent advances in accelerating SGD [123, 87], we further propose a novel momentum-based FedAvg algorithm,

which enjoys an improved convergence rate over FedAvg. Detailed proofs are deferred to Appendix Section B. In particular, we do not need Assumptions 9 and 10 and use modified versions of Assumptions 7 and 8 detailed in this section.

6.5.1 Geometric Convergence of FedAvg in the Overparameterized Setting

Recall the FL problem $\min_w \sum_{k=1}^N p_k F_k(\mathbf{w})$ with $F_k(\mathbf{w}) = \frac{1}{n_k}\sum_{j=1}^{n_k} \ell(\mathbf{w}; \mathbf{x}_k^j)$. In this section, we consider the standard Empirical Risk Minimization (ERM) setting where ℓ is non-negative, l-smooth, and convex, and as before, each $F_k(\mathbf{w})$ is L-smooth and μ-strongly convex. Note that $l \geq L$. This setup includes many important problems in practice. In the overparameterized setting, there exists $\mathbf{w}^* \in \arg\min_w \sum_{k=1}^N p_k F_k(\mathbf{w})$ such that $\ell(\mathbf{w}^*; \mathbf{x}_k^j) = 0$ for all $\mathbf{x}_k^j$. We first show that FedAvg achieves geometric convergence with linear speedup in the number of workers.

Theorem 11. *In the overparameterized setting, FedAvg with communication every E iterations and constant step size $\overline{\alpha} = \mathcal{O}(\frac{1}{E}\frac{N}{l\nu_{\max} + L(N-\nu_{\min})})$ has geometric convergence:*

$$\mathbb{E}F(\overline{\mathbf{w}}_T) \leq \frac{L}{2}(1-\overline{\alpha})^T \|\mathbf{w}_0 - \mathbf{w}^*\|^2 = \mathcal{O}\left(L\exp\left(-\frac{\mu}{E}\frac{NT}{l\nu_{\max} + L(N-\nu_{\min})}\right)\cdot\|\mathbf{w}_0 - \mathbf{w}^*\|^2\right).$$

Linear speedup and Communication Complexity The linear speedup factor is on the order of $\mathcal{O}(N/E)$ for $N \leq \mathcal{O}(\frac{l}{L})$, i.e. FedAvg with N workers and communication every E iterations provides a geometric convergence speedup factor of $\mathcal{O}(N/E)$, for $N \leq \mathcal{O}(\frac{l}{L})$. When N is above this threshold, however, the speedup is almost constant in the number of workers. This matches the findings in [127]. Our result also illustrates that E can be taken $\mathcal{O}(T^\beta)$ for any $\beta < 1$ to achieve geometric convergence, achieving better communication

efficiency than the standard FL setting.

6.5.2 Overparameterized Linear Regression Problems

We now turn to quadratic problems and show that the bound in Theorem 11 can be improved to $\mathcal{O}(\exp(-\frac{N}{E\kappa_1}t))$ for a larger range of N. We then propose a variant of FedAvg that has provable acceleration over FedAvg with SGD updates. The local device objectives are now given by the sum of squares $F_k(\mathbf{w}) = \frac{1}{2n_k}\sum_{j=1}^{n_k}(\mathbf{w}^T\mathbf{x}_k^j - z_k^j)^2$, and there exists $\mathbf{w}^*$ such that $F(\mathbf{w}^*) \equiv 0$. Two notions of condition number are important in our results: κ_1 which is based on local Hessians, and $\tilde{\kappa}$, which is termed the statistical condition number [119, 87]. For their detailed definitions, please refer to Appendix Section B. Here we use the fact $\tilde{\kappa} \leq \kappa_1$. Recall $\nu_{\max} = \max_k p_k N$ and $\nu_{\min} = \min_k p_k N$.

Theorem 12. *For the overparamterized linear regression problem, FedAvg with communication every E iterations with constant step size $\overline{\alpha} = \mathcal{O}(\frac{1}{E}\frac{N}{l\nu_{\max}+\mu(N-\nu_{\min})})$ has geometric convergence:*

$$\mathbb{E}F(\overline{\mathbf{w}}_T) \leq \mathcal{O}\left(L\exp(-\frac{NT}{E(\nu_{\max}\kappa_1 + (N-\nu_{\min}))})\|\mathbf{w}_0 - \mathbf{w}^*\|^2\right).$$

When $N = \mathcal{O}(\kappa_1)$, the convergence rate is $\mathcal{O}((1-\frac{N}{E\kappa_1})^T) = \mathcal{O}(\exp(-\frac{NT}{E\kappa_1}))$, which exhibits linear speedup in the number of workers, as well as a $1/\kappa_1$ dependence on the condition number κ_1. Inspired by [119], we propose the **MaSS accelerated FedAvg algorithm** (FedMaSS):

$$\mathbf{w}_{t+1}^k = \begin{cases} \mathbf{u}_t^k - \eta_1^k \mathbf{g}_{t,k} & \text{if } t+1 \notin \mathcal{I}_E, \\ \sum_{k\in\mathcal{S}_{t+1}}\left[\mathbf{u}_t^k - \eta_1^k \mathbf{g}_{t,k}\right] & \text{if } t+1 \in \mathcal{I}_E, \end{cases}$$

$$\mathbf{u}_{t+1}^k = \mathbf{w}_{t+1}^k + \gamma^k(\mathbf{w}_{t+1}^k - \mathbf{w}_t^k) + \eta_2^k \mathbf{g}_{t,k}.$$

When $\eta_2^k \equiv 0$, this algorithm reduces to the Nesterov accelerated FedAvg algorithm. In the next theorem, we demonstrate that FedMaSS improves the convergence to $\mathcal{O}(\exp(-\frac{NT}{E\sqrt{\kappa_1\tilde{\kappa}}}))$. To our knowledge, this is the first acceleration result of FedAvg with momentum updates over SGD updates.

Theorem 13. *For the overparamterized linear regression problem, FedMaSS with communication every E iterations and constant step sizes $\overline{\eta}_1 = \mathcal{O}(\frac{1}{E}\frac{N}{l\nu_{\max}+\mu(N-\nu_{\min})}), \overline{\eta}_2 = \frac{\overline{\eta}_1(1-\frac{1}{\tilde{\kappa}})}{1+\frac{1}{\sqrt{\kappa_1\tilde{\kappa}}}}, \overline{\gamma} = \frac{1-\frac{1}{\sqrt{\kappa_1\tilde{\kappa}}}}{1+\frac{1}{\sqrt{\kappa_1\tilde{\kappa}}}}$ has geometric convergence:*

$$\mathbb{E}F(\overline{\mathbf{w}}_T) \leq \mathcal{O}\left(L\exp(-\frac{NT}{E(\nu_{\max}\sqrt{\kappa_1\tilde{\kappa}}+(N-\nu_{\min}))})\|\mathbf{w}_0-\mathbf{w}^*\|^2\right).$$

Speedup of FedMaSS over FedAvg To better understand the significance of the above result, we briefly discuss related works on accelerating SGD. Nesterov and Heavy Ball updates are known to fail to accelerate over SGD in both the overparameterized and convex settings [119, 100, 122, 221]. Thus in general one cannot hope to obtain acceleration results for the FedAvg algorithm with Nesterov and Heavy Ball updates. Luckily, recent works in SGD [87, 119] introduced an additional compensation term to the Nesterov updates to address the non-acceleration issue. Surprisingly, we show the same approach can effectively improve the rate of FedAvg. Comparing the convergence rate of FedMass (Theorem 13) and FedAvg (Theorem 12), when $N = \mathcal{O}(\sqrt{\kappa_1\tilde{\kappa}})$, the convergence rate is $\mathcal{O}((1-\frac{N}{E\sqrt{\kappa_1\tilde{\kappa}}})^T) = \mathcal{O}(\exp(-\frac{NT}{E\sqrt{\kappa_1\tilde{\kappa}}}))$ as opposed to $\mathcal{O}(\exp(-\frac{NT}{E\kappa_1}))$. Since $\kappa_1 \geq \tilde{\kappa}$, this implies a speedup factor of $\sqrt{\frac{\kappa_1}{\tilde{\kappa}}}$ for FedMaSS. On the other hand, the same linear speedup in the number of workers

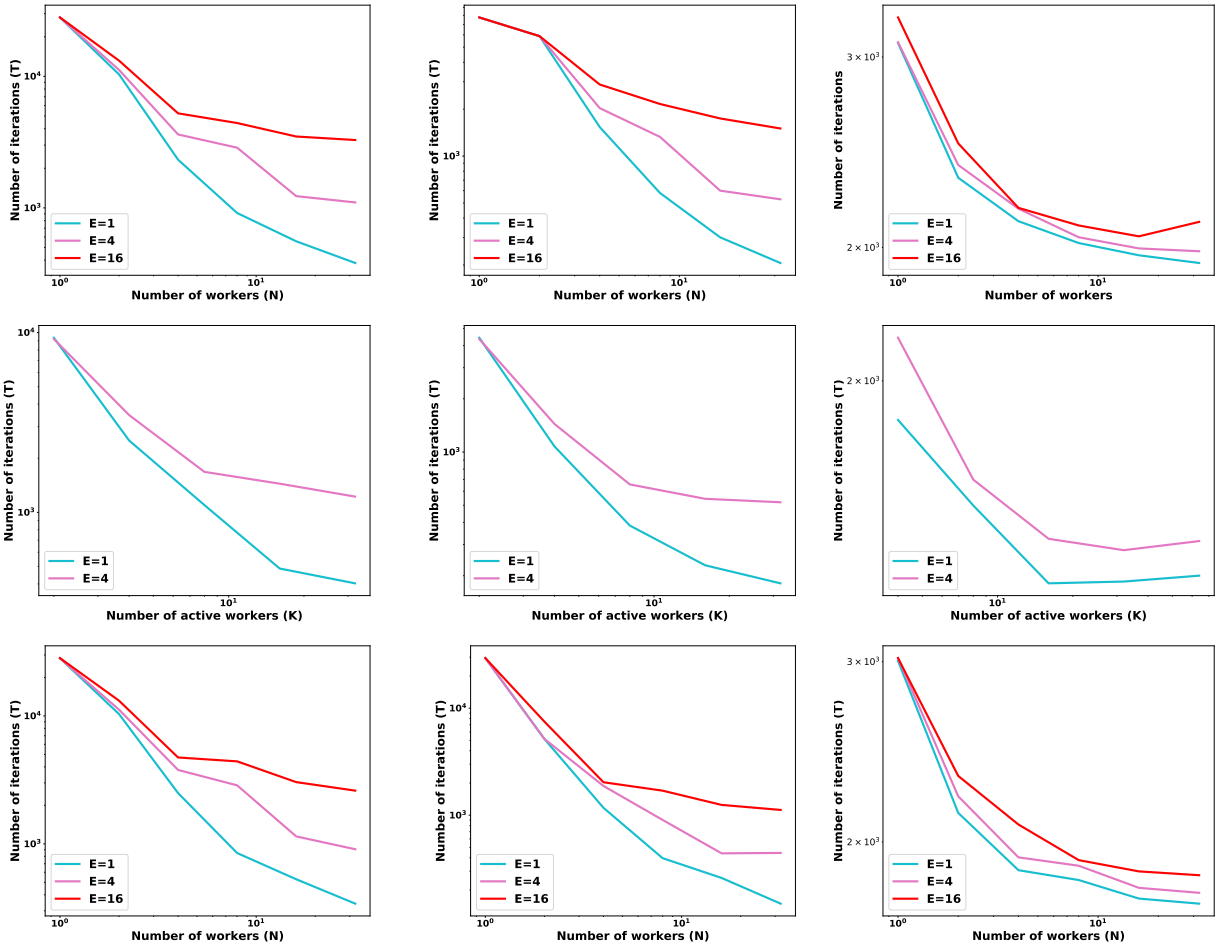
holds for N in a smaller range of values.

6.6 Numerical Experiments

In this section, we empirically examine the linear speedup convergence of FedAvg and Nesterov accelerated FedAvg in various settings, including strongly convex function, convex smooth function, and overparameterized objectives, as analyzed in previous sections.

Setup. Following the experimental setting in [178], we conduct experiments on both synthetic datasets and real-world dataset w8a [155] ($d = 300, n = 49749$). We consider the distributed objectives $F(\mathbf{w}) = \sum_{k=1}^{N} p_k F_k(\mathbf{w})$, and the objective function on the k-th local device includes three cases: 1) **Strongly convex objective**: the regularized binary logistic regression problem, $F_k(\mathbf{w}) = \frac{1}{N_k} \sum_{i=1}^{N_k} \log(1 + \exp(-y_i^k \mathbf{w}^T \mathbf{x}_i^k) + \frac{\lambda}{2}\|\mathbf{w}\|^2$. The regularization parameter is set to $\lambda = 1/n \approx 2e - 5$. 2) **Convex smooth objective**: the binary logistic regression problem without regularization. 3) **Overparameterized setting**: the linear regression problem without adding noise to the label, $F_k(\mathbf{w}) = \frac{1}{N_k} \sum_{i=1}^{N_k} (\mathbf{w}^T \mathbf{x}_i^k + b - y_i^k)^2$.

Linear speedup of FedAvg and Nesterov accelerated FedAvg. To verify the linear speedup convergence as shown in Theorems 7 8 9 10, we evaluate the number of iterations needed to reach ϵ-accuracy in three objectives. We initialize all runs with $\mathbf{w}_0 = \mathbf{0}_d$ and measure the number of iterations to reach the target accuracy ϵ. For each configuration (E, K), we extensively search the learning rate from $\min(\eta_0, \frac{nc}{1+t})$, where $\eta_0 \in \{0.1, 0.12, 1, 32\}$ according to different problems and c can take the values $c = 2^i \ \forall i \in \mathbb{Z}$. As the results shown in Figure 6.1, the number of iterations decreases as the number of (active) workers increasing, which is consistent for FedAvg and Nesterov accelerated FedAvg across all scenarios. For additional experiments on the impact of E, detailed experimental setup, and hyperparameter



(a) Strongly convex objective (b) Convex smooth objective (c) Linear regression

Figure 6.1: The linear speedup of FedAvg in full participation, partial participation, and the linear speedup of Nesterov accelerated FedAvg, respectively.

setting, please refer to the Appendix Section B.

Chapter 7

Conclusion

In this dissertation, we considered the problem of collaborative learning, aiming to find effective ways to leverage knowledge from peers for efficient learning and better generalization. To start, we formally defined the collaborative learning problem and discussed several challenges we need to resolve under this systematic framework. The first challenge we focus on is the flexibility and interactive model-driven collaboration. We present algorithms that capture high-order interactions and interactively incorporate the human expert knowledge to guide the collaboration. Then to generalize the collaboration to heterogeneous learning agents and heterogeneous tasks, we propose data-driven collaborative algorithms, where the learning agents transfer knowledge from a selective and dynamic dataset. In addition to the various form of collaborations, we also study the scalability of collaboration, where we propose linear programming based collaborative multi-agent learning algorithm in the context of a large-scale fleet management application. Last but not least, the empirical success of collaborative learning motivates us to dig into the reason why collaborative learning can be beneficial. We provide rigorous theoretical analysis on the convergence improvement with respect to the increasing number of learning agents.

There are various domains that can benefit from collaborative learning, including but not limited to multi-task meta-learning, transfer learning, federated learning, multi-agent reinforcement learning, etc. The research in the community has been devoted to pushing

the frontier of each domain in-depth, while seldom study their intrinsic connections, which can be essential towards building collaborative machine intelligence. There are emerging researches to reveal the relations across different fields such as the connection between federated learning and multi-task learning [176], federated learning and meta-learning [57], etc, which could serve as the initial step towards bridging the gaps across multiple fields. One central motivation of this dissertation that views those domains as an integrated framework is that the collaboration should be emphasized as a significant learning objective instead of an auxiliary product, along with accomplishing other goals. Our vision is that towards the building the machine intelligence that is comparable to human intelligence, the rigorous understanding of collaborative learning is inevitable.

More concretely, there many future directions under the grant picture of collaborative learning. First and foremost, one fundamental question is what type of tasks can be learned collaboratively, Or when can we expect collaborative learning benefit the performance comparing to learning individually? This is closely related to the negative transfer [207] and and task interference [220] in multi-task learning. In Chapter 6, we quantify a simplified setting in supervised learning where the gradient variance across heterogeneous tasks are bounded, while this is far from desire. In practice, what is the efficient and testing standard before considering collaboration? Another perspective of thinking this problem is that is there always exists a collaboration strategy that works better than individual learning?

Despite the long-term goal of collaborative learning, a promising direction would be learning to collaborate. The current collaboration strategies are mostly predefined. We manually set up the rules of the collaboration according to certain domain knowledge. Can we parameterize the collaboration and learn the intrinsic principle of collaborative learning that is generalizable? Recently, we notice a trend of meta-learning and AI-generating

algorithms [146, 59], while similar efforts haven't been found in collaborative learning. Human can easily generalize the structure of organizations, communication protocol, interaction patterns to solve different tasks. To develop collaborative learning solutions along this direction would be incredibly valuable for generating human-like intelligence.

APPENDICES

Appendix A

Ranking Policy Gradient

Discussion of Existing Efforts on Connecting Reinforcement Learning to Supervised Learning.

There are two main distinctions between supervised learning and reinforcement learning. In supervised learning, the data distribution $\mathcal{D}$ is static and training samples are assumed to be sampled *i.i.d.* from $\mathcal{D}$. On the contrary, the data distribution is dynamic in reinforcement learning and the sampling procedure is not independent. First, since the data distribution in RL is determined by both environment dynamics and the learning policy, and the policy keeps being updated during the learning process. This updated policy results in dynamic data distribution in reinforcement learning. Second, policy learning depends on previously collected samples, which in turn determines the sampling probability of incoming data. Therefore, the training samples we collected are not independently distributed. These intrinsic difficulties of reinforcement learning directly cause the sample-inefficient and unstable performance of current algorithms.

On the other hand, most state-of-the-art reinforcement learning algorithms can be shown to have a supervised learning equivalent. To see this, recall that most reinforcement learning algorithms eventually acquire the policy either explicitly or implicitly, which is a mapping from a state to an action or a probability distribution over the action space. The use of such

a mapping implies that ultimately there exists a supervised learning equivalent to the original reinforcement learning problem, if optimal policies exist. The paradox is that it is almost impossible to construct this supervised learning equivalent on the fly, without knowing any optimal policy.

Although the question of how to construct and apply proper supervision is still an open problem in the community, there are many existing efforts providing insightful approaches to reduce reinforcement learning into its supervised learning counterpart over the past several decades. Roughly, we can classify the existing efforts into the following categories:

- *Expectation-Maximization (EM)*: [45, 152, 102, 1], etc.
- *Entropy-Regularized RL (ERL)*: [144, 145, 74], etc.
- *Interactive Imitation Learning (IIL)*: [44, 188, 163, 165, 184], etc.

The early approaches in the EM track applied Jensen's inequality and approximation techniques to transform the reinforcement learning objective. Algorithms are then derived from the transformed objective, which resemble the Expectation-Maximization procedure and provide policy improvement guarantee [45]. These approaches typically focus on a simplified RL setting, such as assuming that the reward function is not associated with the state [45], approximating the goal to maximize the expected immediate reward and the state distribution is assumed to be fixed [153]. Later on in [102], the authors extended the EM framework from targeting immediate reward into episodic return. Recently, [1] used the EM-framework on a relative entropy objective, which adds a parameter prior as regularization. It has been found that the estimation step using *Retrace* [135] can be unstable even with a linear function approximation [197]. In general, the estimation step in EM-based algorithms involves on-policy evaluation, which is one challenge shared among policy gradient methods.

On the other hand, off-policy learning usually leads to a much better sample efficiency, and is one main motivation that we want to reformulate RL into a supervised learning task.

To achieve off-policy learning, PGQ [144] connected the entropy-regularized policy gradient with Q-learning under the constraint of small regularization. In the similar framework, Soft Actor-Critic [74] was proposed to enable sample-efficient and faster convergence under the framework of entropy-regularized RL. It is able to converge to the optimal policy that optimizes the long-term reward along with policy entropy. It is an efficient way to model the suboptimal behavior and empirically it is able to learn a reasonable policy. Although recently the discrepancy between the entropy-regularized objective and original long-term reward has been discussed in [143, 56], they focus on learning stochastic policy while the proposed framework is feasible for both learning deterministic optimal policy (Corollary 1) and stochastic optimal policy (Corollary 2). In [145], this work shares similarity to our work in terms of the method we collecting the samples. They collect good samples based on the past experience and then conduct the imitation learning w.r.t those good samples. However, we differentiate at how do we look at the problem theoretically. This self-imitation learning procedure was eventually connected to lower-bound-soft-Q-learning, which belongs to entropy-regularized reinforcement learning. We comment that there is a trade-off between sample-efficiency and modeling suboptimal behaviors. The more strict requirement we have on the samples collected we have less chance to hit the samples while we are more close to imitating the optimal behavior.

From the track of interactive imitation learning, early efforts such as [163, 165] pointed out that the main discrepancy between imitation learning and reinforcement learning is the violation of *i.i.d.* assumption. SMILE [163] and DAGGER [165] are proposed to overcome the distribution mismatch. Theorem 2.1 in [163] quantified the performance degradation from the

Table A.1: A comparison of studies reducing RL to SL. The *Objective* column denotes whether the goal is to maximize long-term reward. The *Cont. Action* column denotes whether the method is applicable to both continuous and discrete action spaces. The *Optimality* denotes whether the algorithms can model the optimal policy. ✓† denotes the optimality achieved by ERL is w.r.t. the entropy regularize objective instead of the original objective on return. The *Off-Policy* column denotes if the algorithms enable off-policy learning. The *No Oracle* column denotes if the algorithms need to access to a certain type of oracle (expert policy or expert demonstrations).

Methods	Objective	Cont. Action	Optimality	Off-Policy	No Oracle
EM	✓	✓	✓	✗	✓
ERL	✗	✓	✓†	✓	✓
IIL	✓	✓	✓	✓	✗
RPG	✓	✗	✓	✓	✓

expert considering that the learned policy fails to imitate the expert with a certain probability. The theorem seems to resemble the long-term performance theorem (Thm. 5) in this chapter. However, it studied the scenario that the learning policy is trained through a state distribution induced by the expert, instead of state-action distribution as considered in Theorem 5. As such, Theorem 2.1 in [163] may be more applicable to the situation where an interactive procedure is needed, such as querying the expert during the training process. On the contrary, the proposed work focuses on directly applying supervised learning without having access to the expert to label the data. The optimal state-action pairs are collected during exploration and conducting supervised learning on the replay buffer will provide a performance guarantee in terms of long-term expected reward. Concurrently, a resemble of Theorem 2.1 in [163] is Theorem 1 in [188], where the authors reduced the apprenticeship learning to classification, under the assumption that the apprentice policy is deterministic and the misclassification rate is bounded at all time steps. In this work, we show that it is possible to circumvent such a strong assumption and reduce RL to its SL. Furthermore, our theoretical framework also leads to an alternative analysis of sample-complexity. Later on AggreVaTe [164] was proposed to incorporate the information of action costs to facilitate imitation learning,

and its differentiable version AggreVaTeD [184] was developed in succession and achieved impressive empirical results. Recently, hinge loss was introduced to regular Q-learning as a pre-training step for learning from demonstration [81], or as a surrogate loss for imitating optimal trajectories [148]. In this work, we show that hinge loss constructs a new type of policy gradient method and can be used to learn optimal policy directly.

In conclusion, our method approaches the problem of reducing RL to SL from a unique perspective that is different from all prior work. With our reformulation from RL to SL, the samples collected in the replay buffer satisfy the *i.i.d.* assumption, since the state-action pairs are now sampled from the data distribution of UNOP. A multi-aspect comparison between the proposed method and relevant prior studies is summarized in Table A.1.

Ranking Policy Gradient Theorem

The Ranking Policy Gradient Theorem (Theorem 2) formulates the optimization of long-term reward using a ranking objective. The proof below illustrates the formulation process.

Proof. The following proof is based on direct policy differentiation [153, 212]. For a concise presentation, the subscript t for action value λ_i, λ_j, and p_{ij} is omitted.

$$
\begin{aligned}
\nabla_\theta J(\theta) =& \nabla_\theta \sum\nolimits_\tau p_\theta(\tau) r(\tau) \\
=& \sum\nolimits_\tau p_\theta(\tau) \nabla_\theta \log p_\theta(\tau) r(\tau) \\
=& \sum\nolimits_\tau p_\theta(\tau) \nabla_\theta \log \left(p(s_0) \Pi_{t=1}^{T} \pi_\theta(a_t|s_t) p(s_{t+1}|s_t, a_t) \right) r(\tau) \\
=& \sum\nolimits_\tau p_\theta(\tau) \sum\nolimits_{t=1}^{T} \nabla_\theta \log \pi_\theta(a_t|s_t) r(\tau) \\
=& \mathbf{E}_{\tau \sim \pi_\theta} \left[\sum\nolimits_{t=1}^{T} \nabla_\theta \log \pi_\theta(a_t|s_t) r(\tau) \right]
\end{aligned}
\tag{A.1}
$$

$$
\begin{aligned}
=&\mathbf{E}_{\tau\sim\pi_\theta}\left[\sum_{t=1}^{T}\nabla_\theta \log\left(\prod_{j=1,j\neq i}^{m} p_{ij}\right) r(\tau)\right] \\
=&\mathbf{E}_{\tau\sim\pi_\theta}\left[\sum_{t=1}^{T}\nabla_\theta \sum_{j=1,j\neq i}^{m} \log\left(\frac{e^{\lambda_{ij}}}{1+e^{\lambda_{ij}}}\right) r(\tau)\right] \\
=&\mathbf{E}_{\tau\sim\pi_\theta}\left[\sum_{t=1}^{T}\nabla_\theta \sum_{j=1,j\neq i}^{m} \log\left(\frac{1}{1+e^{\lambda_{ji}}}\right) r(\tau)\right] \qquad \text{(A.2)} \\
\approx&\mathbf{E}_{\tau\sim\pi_\theta}\left[\sum_{t=1}^{T}\nabla_\theta \left(\sum_{j=1,j\neq i}^{m} (\lambda_i-\lambda_j)/2\right) r(\tau)\right], \qquad \text{(A.3)}
\end{aligned}
$$

where the trajectory is a series of state-action pairs from $t = 1, ..., T$, *i.e.*$\tau = s_1, a_1, s_2, a_2, ..., s_T$. From Eq (A.2) to Eq (A.3), we use the first-order Taylor expansion of $\log(1 + e^x)|_{x=0} = \log 2 + \frac{1}{2}x + O(x^2)$ to further simplify the ranking policy gradient. □

Probability Distribution in Ranking Policy Gradient

In this section, we discuss the output property of the pairwise ranking policy. We show in Corollary 6 that the pairwise ranking policy gives a valid probability distribution when the dimension of the action space $m = 2$. For cases when $m > 2$ and the range of Q-value satisfies Condition 2, we show in Corollary 7 how to construct a valid probability distribution.

Corollary 6. *The pairwise ranking policy as shown in Eq (4.5) constructs a probability distribution over the set of actions when the action space m is equal to 2, given any action values $\lambda_i, i = 1, 2$. For the cases with $m > 2$, this conclusion does not hold in general.*

It is easy to verify that $\pi(a_i|s) > 0$, $\sum_{i=1}^{2} \pi(a_i|s) = 1$ holds and the same conclusion cannot be applied to $m > 2$ by constructing counterexamples. However, we can introduce a dummy action a' to form a probability distribution for RPG. During policy learning, the algorithm increases the probability of best actions and the probability of dummy action decreases. Ideally, if RPG converges to an optimal deterministic policy, the probability of

taking best action is equal to 1 and $\pi(a'|s) = 0$. Similarly, we can introduce a dummy trajectory τ' with the trajectory reward $r(\tau') = 0$ and $p_\theta(\tau') = 1 - \sum_\tau p_\theta(\tau)$. The trajectory probability forms a probability distribution since $\sum_\tau p_\theta(\tau) + p_\theta(\tau') = 1$ and $p_\theta(\tau) \geq 0 \ \forall \tau$ and $p_\theta(\tau') \geq 0$. The proof of a valid trajectory probability is similar to the following proof on $\pi(a|s)$ to be a valid probability distribution with a dummy action. Its practical influence is negligible since our goal is to increase the probability of (near)-optimal trajectories. To present in a clear way, we avoid mentioning dummy trajectory τ' in Proof A while it can be seamlessly included.

Condition 2 (The range of action-value)**.** *We restrict the range of action-values in RPG so that it satisfies* $\lambda_m \geq \ln(m^{\frac{1}{m-1}} - 1)$*, where* $\lambda_m = \min_{i,j} \lambda_{ji}$ *and* m *is the dimension of the action space.*

This condition can be easily satisfied since in RPG we only focus on the relative relationship of λ and we can constrain the range of action-values so that λ_m satisfies the condition 2. Furthermore, since we can see that $m^{\frac{1}{m-1}} > 1$ is decreasing w.r.t to action dimension m. The larger the action dimension, the less constraint we have on the action values.

Corollary 7. *Given Condition 2, we introduce a dummy action* a' *and set* $\pi(a = a'|s) = 1 - \sum_i \pi(a = a_i|s)$*, which constructs a valid probability distribution* $(\pi(a|s))$ *over the action space* $\mathcal{A} \cup a'$*.*

Proof. Since we have $\pi(a = a_i|s) > 0 \ \forall i = 1, \ldots, m$ and $\sum_i \pi(a = a_i|s) + \pi(a = a'|s) = 1$. To prove that this is a valid probability distribution, we only need to show that $\pi(a = a'|s) \geq 0, \ \forall m \geq 2$, i.e. $\sum_i \pi(a = a_i|s) \leq 1, \ \forall m \geq 2$. Let $\lambda_m = \min_{i,j} \lambda_{ji}$,

$$\sum\nolimits_i \pi(a = a_i|s)$$

$$
\begin{aligned}
&= \sum_i \prod_{j=1, j\neq i}^{m} p_{ij} \\
&= \sum_i \prod_{j=1, j\neq i}^{m} \frac{1}{1+e^{\lambda_{ji}}} \\
&\leq \sum_i \prod_{j=1, j\neq i}^{m} \frac{1}{1+e^{\lambda_m}} \\
&= m \left(\frac{1}{1+e^{\lambda_m}}\right)^{m-1} \leq 1 \qquad \text{(Condition 2).}
\end{aligned}
$$

This thus concludes the proof. □

Condition of Preserving Optimality

The following condition describes what types of MDPs are directly applicable to the trajectory reward shaping (TRS, Def 6):

Condition 3 (Initial States)**.** *The (near)-optimal trajectories will cover all initial states of MDP. i.e.* $\{s(\tau,1)|\ \forall\tau \in \mathcal{T}\} = \{s(\tau,1)|\ \forall\tau\}$*, where* $\mathcal{T} = \{\tau|w(\tau)=1\} = \{\tau|r(\tau)\geq c\}$*.*

The MDPs satisfying this condition cover a wide range of tasks such as Dialogue System [111], Go [175], video games [18] and all MDPs with only one initial state. If we want to preserve the optimality by TRS, the optimal trajectories of a MDP need to cover all initial states or equivalently, all initial states must lead to at least one optimal trajectory. Similarly, the near-optimality is preserved for all MDPs that its near-optimal trajectories cover all initial states.

Theoretically, it is possible to transfer more general MDPs to satisfy Condition 3 and preserve the optimality with potential-based reward shaping [139]. More concretely, consider the deterministic binary tree MDP ($\mathcal{M}_1$) with the set of initial states $\mathcal{S}_1 = \{s_1, s_1'\}$ as defined in Figure A.1. There are eight possible trajectories in $\mathcal{M}_1$. Let $r(\tau_1) = 10 = R_{\max}, r(\tau_8) =$

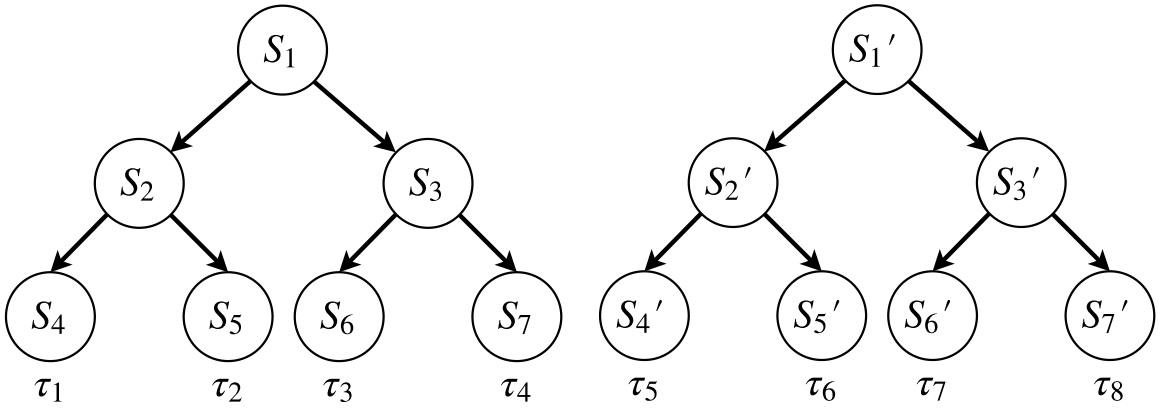


Figure A.1: The binary tree structure MDP with two initial states.

3, $r(\tau_i) = 2,\ \forall i = 2, \ldots, 7$. Therefore, this MDP does not satisfy Condition 3. We can compensate the trajectory reward of the best trajectory starting from s_1' to the $R_{\max}$ by shaping the reward with the potential-based function $\phi(s_7') = 7$ and $\phi(s) = 0, \forall s \neq s_7'$. This reward shaping requires more prior knowledge, which may not be feasible in practice. A more realistic method is to design a dynamic trajectory reward shaping approach. In the beginning, we set $c(s) = \min_{s \in \mathcal{S}_1} r(\tau | s(\tau, 1) = s), \forall s \in \mathcal{S}_1$. Take $\mathcal{M}_1$ as an example, $c(s) = 3,\ \forall s \in \mathcal{S}_1$. During the exploration stage, we track the current best trajectory of each initial state and update $c(s)$ with its trajectory reward.

Nevertheless, if the Condition 3 is not satisfied, we need more sophisticated prior knowledge other than a predefined trajectory reward threshold c to construct the replay buffer (training dataset of UNOP). The practical implementation of trajectory reward shaping and rigorously theoretical study for general MDPs are beyond the scope of this work.

Proof of Long-term Performance Theorem 5

Lemma 3. *Given a specific trajectory τ, the log-likelihood of state-action pairs over horizon T is equal to the weighted sum over the entire state-action space, i.e.:*

$$\frac{1}{T}\sum\nolimits_{t=1}^{T}\log\pi_\theta(a_t|s_t)=\sum\nolimits_{s,a}p(s,a|\tau)\log\pi_\theta(a|s),$$

where the sum in the right hand side is the summation over all possible state-action pairs. It is worth noting that $p(s,a|\tau)$ is not related to any policy parameters. It is the probability of a specific state-action pair (s,a) in a specific trajectory τ.

Proof. Given a trajectory $\tau=\{(s(\tau,1),a(\tau,1)),\ldots,(s(\tau,T),a(\tau,T))\}=\{(s_1,a_1),\ldots,(s_T,a_T)\}$, denote the unique state-action pairs in this trajectory as $U(\tau)=\{(s_i,a_i)\}_{i=1}^{n}$, where n is the number of unique state-action pairs in τ and $n\leq T$. The number of occurrences of a state-action pair (s_i,a_i) in the trajectory τ is denoted as $|(s_i,a_i)|$. Then we have the following:

$$\begin{aligned}
&\frac{1}{T}\sum\nolimits_{t=1}^{T}\log\pi_\theta(a_t|s_t)\\
=&\sum\nolimits_{i=1}^{n}\frac{|(s_i,a_i)|}{T}\log\pi_\theta(a_i|s_i)\\
=&\sum\nolimits_{i=1}^{n}p(s_i,a_i|\tau)\log\pi_\theta(a_i|s_i)\\
=&\sum\nolimits_{(s,a)\in U(\tau)}p(s,a|\tau)\log\pi_\theta(a|s) &\text{(A.4)}\\
=&\sum\nolimits_{(s,a)\in U(\tau)}p(s,a|\tau)\log\pi_\theta(a|s)+\sum\nolimits_{(s,a)\notin U(\tau)}p(s,a|\tau)\log\pi_\theta(a|s) &\text{(A.5)}\\
=&\sum\nolimits_{(s,a)}p(s,a|\tau)\log\pi_\theta(a|s)
\end{aligned}$$

From Eq (A.4) to Eq (A.5) we used the fact:

$$\sum_{(s,a)\in U(\tau)} p(s,a|\tau) = \sum_{i=1}^{n} p(s_i,a_i|\tau) = \sum_{i=1}^{n} \frac{|(s_i,a_i)|}{T} = 1,$$

and therefore we have $p(s,a|\tau) = 0,\ \forall (s,a) \notin U(\tau)$. This thus completes the proof. □

Now we are ready to prove the Theorem 5:

Proof. The following proof holds for an arbitrary subset of trajectories $\mathcal{T}$ determined by the threshold c in Def 8. The π_* is associated with c and this subset of trajectories. We present the following lower bound of the expected long-term performance:

$$\begin{aligned}
&\arg\max_\theta \sum\nolimits_\tau p_\theta(\tau) w(\tau) \\
&\because w(\tau) = 0, \text{if } \tau \notin \mathcal{T} \\
&= \arg\max_\theta \frac{1}{|\mathcal{T}|} \sum\nolimits_{\tau\in\mathcal{T}} p_\theta(\tau) w(\tau) \\
&\text{use Lemma 5 } \because p_\theta(\tau) > 0 \text{ and } w(\tau) > 0, \therefore \sum\nolimits_{\tau\in\mathcal{T}} p_\theta(\tau) w(\tau) > 0 \\
&= \arg\max_\theta \log\left(\frac{1}{|\mathcal{T}|} \sum\nolimits_{\tau\in\mathcal{T}} p_\theta(\tau) w(\tau)\right) \\
&\because \log\left(\sum\nolimits_{i=1}^{n} x_i/n\right) \geq \sum\nolimits_{i=1}^{n} \log(x_i)/n, \forall i, x_i > 0, \text{we have:} \\
&\log\left(\frac{1}{|\mathcal{T}|} \sum\nolimits_{\tau\in\mathcal{T}} p_\theta(\tau) w(\tau)\right) \geq \sum\nolimits_{\tau\in\mathcal{T}} \frac{1}{|\mathcal{T}|} \log p_\theta(\tau) w(\tau),
\end{aligned}$$

where the lower bound holds when $p_\theta(\tau)w(\tau) = \frac{1}{|\mathcal{T}|}, \forall \tau \in \mathcal{T}$. To this end, we maximize the lower bound of the expected long-term performance:

$$\arg\max_\theta \sum\nolimits_{\tau\in\mathcal{T}} \frac{1}{|\mathcal{T}|} \log p_\theta(\tau) w(\tau)$$

$$= \arg\max_\theta \sum\nolimits_{\tau\in\mathcal{T}} \log(p(s_1)\prod\nolimits_{t=1}^{T}(\pi_\theta(a_t|s_t)p(s_{t+1}|s_t,a_t))w(\tau))$$

$$= \arg\max_\theta \sum\nolimits_{\tau\in\mathcal{T}} \log\left(p(s_1)\prod\nolimits_{t=1}^{T}\pi_\theta(a_t|s_t)\prod\nolimits_{t=1}^{T}p(s_{t+1}|s_t,a_t)w(\tau)\right)$$

$$= \arg\max_\theta \sum\nolimits_{\tau\in\mathcal{T}} \left(\log p(s_1) + \sum\nolimits_{t=1}^{T}\log p(s_{t+1}|s_t,a_t) + \sum\nolimits_{t=1}^{T}\log\pi_\theta(a_t|s_t) + \log w(\tau)\right) \tag{A.6}$$

The above shows that $w(\tau)$ can be set as an arbitrary positive constant

$$= \arg\max_\theta \frac{1}{|\mathcal{T}|}\sum\nolimits_{\tau\in\mathcal{T}}\sum\nolimits_{t=1}^{T}\log\prod\nolimits_\theta(a_t|s_t)$$

$$= \arg\max_\theta \frac{1}{|\mathcal{T}|T}\sum\nolimits_{\tau\in\mathcal{T}}\sum\nolimits_{t=1}^{T}\log\prod\nolimits_\theta(a_t|s_t) \tag{A.7}$$

$$= \arg\max_\theta \frac{1}{|\mathcal{T}|}\sum\nolimits_{\tau\in\mathcal{T}}\frac{1}{T}\sum\nolimits_{t=1}^{T}\log\pi_\theta(a_t|s_t) \quad \text{(the existence of UNOP in Assumption 5)}$$

$$= \arg\max_\theta \sum\nolimits_{\tau\in\mathcal{T}} p_{\pi_*}(\tau)\frac{1}{T}\left(\sum\nolimits_{t=1}^{T}\log\pi_\theta(a_t|s_t)\right)$$

where π_* is a UNOP (Def 8) $\Rightarrow p_{\pi_*}(\tau) = 0 \quad \forall\tau \notin \mathcal{T}$ (A.8)

Eq (A.8) can be established based on $\sum_{\tau\in\mathcal{T}} p_{\pi_*}(\tau) = \sum_{\tau\in\mathcal{T}} 1/|\mathcal{T}| = 1$

$$= \arg\max_\theta \sum\nolimits_{\tau} p_{\pi_*}(\tau)\frac{1}{T}\left(\sum\nolimits_{t=1}^{T}\log\pi_\theta(a_t|s_t)\right) \qquad \text{(Lemma 3)}$$

$$= \arg\max_\theta \sum\nolimits_{\tau} p_{\pi_*}(\tau)\sum\nolimits_{s,a} p(s,a|\tau)\log\pi_\theta(a|s)$$

The 2nd sum is over all possible state-action pairs. (A.9)

(s,a) represents a specific state-action pair.

$$= \arg\max_\theta \sum\nolimits_{\tau}\sum\nolimits_{s,a} p_{\pi_*}(\tau)p(s,a|\tau)\log\pi_\theta(a|s)$$

$$= \arg\max_\theta \sum\nolimits_{s,a}\sum\nolimits_{\tau} p_{\pi_*}(\tau)p(s,a|\tau)\log\pi_\theta(a|s)$$

$$= \arg\max_\theta \sum\nolimits_{s,a} p_{\pi_*}(s,a)\log\pi_\theta(a|s). \tag{A.10}$$

In this proof we use $s_t = s(\tau,t)$ and $a_t = a(\tau,t)$ as abbreviations, which denote the t-th state

and action in the trajectory τ, respectively. $|\mathcal{T}|$ denotes the number of trajectories in $\mathcal{T}$. We also use the definition of $w(\tau)$ to only focus on near-optimal trajectories. We set $w(\tau) = 1$ for simplicity but it will not affect the conclusion if set to other constants.

Optimality: Furthermore, the optimal solution for the objective function Eq (A.10) is a uniformly (near)-optimal policy π_*.

$$
\begin{aligned}
&\arg\max_\theta \sum\nolimits_{s,a} p_{\pi_*}(s,a) \log \pi_\theta(a|s) \\
=&\arg\max_\theta \sum\nolimits_{s} p_{\pi_*}(s) \sum\nolimits_{a} \pi_*(a|s) \log \pi_\theta(a|s) \\
=&\arg\max_\theta \sum\nolimits_{s} p_{\pi_*}(s) \sum\nolimits_{a} \pi_*(a|s) \log \pi_\theta(a|s) - \sum\nolimits_{s} p_{\pi_*}(s) \sum\nolimits_{a} \log \pi_*(a|s) \\
=&\arg\max_\theta \sum\nolimits_{s} p_{\pi_*}(s) \sum\nolimits_{a} \pi_*(a|s) \log \frac{\pi_\theta(a|s)}{\pi_*(a|s)} \\
=&\arg\max_\theta \sum\nolimits_{s} p_{\pi_*}(s) \sum\nolimits_{a} -KL(\pi_*(a|s)||\pi_\theta(a|s)) = \pi_*
\end{aligned}
$$

Therefore, the optimal solution of Eq (A.10) is also the (near)-optimal solution for the original RL problem since $\sum_\tau p_{\pi_*}(\tau) r(\tau) = \sum_{\tau \in \mathcal{T}} \frac{1}{|\mathcal{T}|} r(\tau) \geq c = R_{\max} - \epsilon$. The optimal solution is obtained when we set $c = R_{\max}$. □

Lemma 4. *Given any optimal policy π of MDP satisfying Condition 3, $\forall \tau \notin \mathcal{T}$, we have $p_\pi(\tau) = 0$, where $\mathcal{T}$ denotes the set of all possible optimal trajectories in this lemma. If $\exists \tau \notin \mathcal{T}$, such that $p_\pi(\tau) > 0$, then π is not an optimal policy.*

Proof. We prove this by contradiction. We assume π is an optimal policy. If $\exists \tau' \notin \mathcal{T}$, such that 1) $p_\pi(\tau') \neq 0$, or equivalently: $p_\pi(\tau') > 0$ since $p_\pi(\tau') \in [1, 0]$. and 2) $\tau' \notin \mathcal{T}$. We can find a better policy π' by satisfying the following three conditions:

$$p_{\pi'}(\tau') = 0 \text{ and}$$

$$p_{\pi'}(\tau_1) = p_\pi(\tau_1) + p_\pi(\tau'), \tau_1 \in \mathcal{T} \text{ and}$$

$$p_{\pi'}(\tau) = p_\pi(\tau), \forall \tau \notin \{\tau', \tau_1\}$$

Since $p_{\pi'}(\tau) \geq 0, \forall \tau$ and $\sum_\tau p_{\pi'}(\tau) = 1$, therefore $p_{\pi'}$ constructs a valid probability distribution. Then the expected long-term performance of π' is greater than that of π:

$$\begin{aligned}
&\sum\nolimits_\tau p_{\pi'}(\tau)w(\tau) - \sum\nolimits_\tau p_\pi(\tau)w(\tau) \\
=&\sum\nolimits_{\tau \notin \{\tau', \tau_1\}} p_{\pi'}(\tau)w(\tau) + p_{\pi'}(\tau_1)w(\tau_1) + p_{\pi'}(\tau')w(\tau') \\
&\quad - \left(\sum\nolimits_{\tau \notin \{\tau', \tau_1\}} p_\pi(\tau)w(\tau) + p_\pi(\tau_1)w(\tau_1) + p_\pi(\tau')w(\tau')\right) \\
=&p_{\pi'}(\tau_1)w(\tau_1) + p_{\pi'}(\tau')w(\tau') - (p_\pi(\tau_1)w(\tau_1) + p_\pi(\tau')w(\tau')) \\
&\because \tau' \notin \mathcal{T}, \therefore w(\tau') = 0 \text{ and } \tau_1 \in \mathcal{T}, \therefore w(\tau) = 1 \\
=&p_{\pi'}(\tau_1) - p_\pi(\tau_1) \\
=&p_\pi(\tau_1) + p_\pi(\tau') - p_\pi(\tau_1) = p_\pi(\tau') > 0.
\end{aligned}$$

Essentially, we can find a policy π' that has higher probability on the optimal trajectory τ_1 and zero probability on τ'. This indicates that it is a better policy than π. Therefore, π is not an optimal policy and it contradicts our assumption, which proves that such τ' does not exist. Therefore, $\forall \tau \notin \mathcal{T}$, we have $p_\pi(\tau) = 0$. □

Lemma 5 (Policy Performance)**.** *If the policy takes the form as in Eq (4.7) or Eq (4.5), then we have* $\forall \tau$, $p_\theta(\tau) > 0$. *This means for all possible trajectories allowed by the environment, the policy takes the form of either ranking policy or softmax will generate this trajectory with probability* $p_\theta(\tau) > 0$. *Note that because of this property,* π_θ *is not an optimal policy according to Lemma 4, though it can be arbitrarily close to an optimal policy.*

Proof.

The trajectory probability is defined as: $p(\tau) = p(s_1)\Pi_{t=1}^{T}(\pi_\theta(a_t|s_t)p(s_{t+1}|s_t, a_t))$

Then we have:

The policy takes the form as in Eq (4.7) or Eq (4.5) $\Rightarrow \pi_\theta(a_t|s_t) > 0$.

$p(s_1) > 0, p(s_{t+1}|s_t, a_t) > 0. \Rightarrow p_\theta(\tau) = 0.$

$p(s_{t+1}|s_t, a_t) = 0$ or $p(s_1) = 0, \Rightarrow p_\theta(\tau) = 0,$ which means τ is not a possible trajectory.

In summary, for all possible trajectories, $p_\theta(\tau) > 0$.

This thus completes the proof. □

Proof of Corollary 1

Corollary 8 (Ranking performance policy gradient)**.** *The lower bound of expected long-term performance by ranking policy can be approximately optimized by the following loss:*

$$\min_\theta \sum\nolimits_{s,a_i} p_{\pi*}(s, a_i)L(s_i, a_i) \tag{A.11}$$

where the pair-wise loss $L(s_i, a_i)$ is defined as:

$$\mathcal{L}(s, a_i) = \sum\nolimits_{j=1,j\neq i}^{|A|} \max(0, 1 + \lambda(s, a_j) - \lambda(s, a_i))$$

Proof. In RPG, the policy $\pi_\theta(a|s)$ is defined as in Eq (4.5). We then replace the action

probability distribution in Eq (4.13) with the RPG policy.

$$\because \pi(a = a_i|s) = \Pi_{j=1,j\neq i}^{m} p_{ij} \tag{A.12}$$

Because RPG is fitting a deterministic optimal policy,

we denote the optimal action given sate s as a_i, then we have

$$\max_{\theta} \sum_{s,a_i} p_{\pi*}(s, a_i) \log \pi(a_i|s) \tag{A.13}$$

$$= \max_{\theta} \sum_{s,a_i} p_{\pi*}(s, a_i) \log(\Pi_{j\neq i,j=1}^{m} p_{ij}) \tag{A.14}$$

$$= \max_{\theta} \sum_{s,a_i} p_{\pi*}(s, a_i) \log \Pi_{j\neq i,j=1}^{m} \frac{1}{1 + e^{\lambda_{ji}}} \tag{A.15}$$

$$= \min_{\theta} \sum_{s,a_i} p_{\pi*}(s, a_i) \sum_{j\neq i,j=1}^{m} \log(1 + e^{\lambda_{ji}}) \text{ first order Taylor expansion} \tag{A.16}$$

$$\approx \min_{\theta} \sum_{s,a_i} p_{\pi*}(s, a_i) \sum_{j\neq i,j=1}^{m} \lambda_{ji} \qquad \text{s.t. } |\lambda_{ij}| = c < 1, \forall i, j, s \tag{A.17}$$

$$= \min_{\theta} \sum_{s,a_i} p_{\pi*}(s, a_i) \sum_{j\neq i,j=1}^{m} (\lambda_j - \lambda_i) \qquad \text{s.t. } |\lambda_i - \lambda_j| = c < 1, \forall i, j, s \tag{A.18}$$

$$\Rightarrow \min_{\theta} \sum_{s,a_i} p_{\pi*}(s, a_i) L(s_i, a_i) \tag{A.19}$$

where the pairwise loss $L(s, a_i)$ is defined as:

$$\mathcal{L}(s, a_i) = \sum_{j=1,j\neq i}^{|A|} \max(0, \text{margin} + \lambda(s, a_j) - \lambda(s, a_i)), \tag{A.20}$$

where the margin in Eq (A.19) is a small positive constant. (A.21)

From Eq (A.18) to Eq (A.19), we consider learning a deterministic optimal policy $a_i = \pi^*(s)$, where we use index i to denote the optimal action at each state. The optimal λ-values minimizing Eq (A.18) (denoted by λ^1) need to satisfy $\lambda_i^1 = \lambda_j^1 + c, \forall j \neq i, s$. The optiaml λ-

values minimizing Eq (A.19) (denoted by λ^2) need to satisfy $\lambda_i^2 = \max_{j \neq i} \lambda_j^2 + \text{margin}, \forall j \neq i, s$. In both cases, the optimal policies from solving Eq (A.18) and Eq (A.18) are the same: $\pi(s) = \arg\max_k \lambda_k^1 = \arg\max_k \lambda_k^2 = a_i$. Therefore, we use Eq (A.19) as a surrogate optimization problem of Eq (A.18). □

Policy gradient variance reduction

Corollary 9 (Variance reduction)**.** *Given a stationary policy, the upper bound of the variance of each dimension of policy gradient is* $\mathcal{O}(T^2C^2R_{\max}^2)$*. The upper bound of gradient variance of maximizing the lower bound of long-term performance Eq (4.13) is* $\mathcal{O}(C^2)$*, where* C *is the maximum norm of log gradient based on Assumption 6. The supervised learning has reduced the upper bound of gradient variance by an order of* $\mathcal{O}(T^2R_{\max}^2)$ *as compared to the regular policy gradient, considering* $R_{\max} \geq 1, T \geq 1$*, which is a very common situation in practice.*

Proof. The regular policy gradient of policy π_θ is given as [212]:

$$\sum\nolimits_\tau p_\theta(\tau)[\sum_{t=1}^{T} \nabla_\theta \log(\pi_\theta(a(\tau,t)|s(\tau,t)))r(\tau)]$$

The regular policy gradient variance of the i-th dimension is denoted as follows:

$$Var\left(\sum_{t=1}^{T} \nabla_\theta \log(\pi_\theta(a(\tau,t)|s(\tau,t))_i)r(\tau)\right)$$

We denote $x_i(\tau) = \sum_{t=1}^{T} \nabla_\theta \log(\pi_\theta(a(\tau,t)|s(\tau,t))_i)r(\tau)$ for convenience. Therefore, x_i is a

random variable. Then apply $var(x) = \mathbf{E}_{p_\theta(\tau)}[x^2] - \mathbf{E}_{p_\theta(\tau)}[x]^2$, we have:

$$
\begin{aligned}
& Var\left(\sum_{t=1}^{T} \nabla_\theta \log(\pi_\theta(a(\tau,t)|s(\tau,t))_i) r(\tau)\right) \\
=& Var\left(x_i(\tau)\right) \\
=& \sum\nolimits_\tau p_\theta(\tau) x_i(\tau)^2 - [\sum\nolimits_\tau p_\theta(\tau) x_i(\tau)]^2 \\
\leq& \sum\nolimits_\tau p_\theta(\tau) x_i(\tau)^2 \\
=& \sum\nolimits_\tau p_\theta(\tau) [\sum\nolimits_{t=1}^{T} \nabla_\theta \log(\pi_\theta(a(\tau,t)|s(\tau,t))_i) r(\tau)]^2 \\
\leq& \sum\nolimits_\tau p_\theta(\tau) [\sum\nolimits_{t=1}^{T} \nabla_\theta \log(\pi_\theta(a(\tau,t)|s(\tau,t))_i)]^2 R_{max}^2 \\
=& R_{max}^2 \sum\nolimits_\tau p_\theta(\tau) [\sum\nolimits_{t=1}^{T} \sum\nolimits_{k=1}^{T} \nabla_\theta \log(\pi_\theta(a(\tau,t)|s(\tau,t))_i) \nabla_\theta \log(\pi_\theta(a(\tau,k)|s(\tau,k)_i)] \\
& \text{(Assumption 6)} \\
\leq& R_{max}^2 \sum\nolimits_\tau p_\theta(\tau) [\sum_{t=1}^{T} \sum_{k=1}^{T} C^2] \\
=& R_{max}^2 \sum\nolimits_\tau p_\theta(\tau) T^2 C^2 \\
=& T^2 C^2 R_{max}^2
\end{aligned}
$$

The policy gradient of long-term performance (Def 7): $\sum_{s,a} p_{\pi_*}(s,a) \nabla_\theta \log \pi_\theta(a|s)$. The policy gradient variance of the i-th dimension is denoted as: $var(\nabla_\theta \log \pi_\theta(a|s)_i)$. Then the upper bound is given by

$$
\begin{aligned}
& var(\nabla_\theta \log \pi_\theta(a|s)_i) \\
&= \sum\nolimits_{s,a} p_{\pi_*}(s,a) [\nabla_\theta \log \pi_\theta(a|s)_i]^2 - [\sum\nolimits_{s,a} p_{\pi_*}(s,a) \nabla_\theta \log \pi_\theta(a|s)_i]^2 \\
&\leq \sum\nolimits_{s,a} p_{\pi_*}(s,a) [\nabla_\theta \log \pi_\theta(a|s)_i]^2 \qquad \text{(Assumption 6)} \\
&\leq \sum\nolimits_{s,a} p_{\pi_*}(s,a) C^2
\end{aligned}
$$

$$= C^2$$

This thus completes the proof. □

Discussions of Assumption 5

In this section, we show that UNOP exists in a range of MDPs. Notice that the lemma 6 shows the sufficient conditions of satisfying Asumption 5 rather than necessary conditions.

Lemma 6. *For MDPs defined in Section 4.2.3 satisfying the following conditions:*

- *Each initial state leads to one optimal trajectory. This also indicates* $|\mathcal{S}_1| = |\mathcal{T}|$*, where* $\mathcal{T}$ *denotes the set of optimal trajectories in this lemma,* $\mathcal{S}_1$ *denotes the set of initial states.*
- *Deterministic transitions, i.e.,* $p(s'|s,a) \in \{0,1\}$.
- *Uniform initial state distribution, i.e.,* $p(s_1) = \frac{1}{|\mathcal{T}|}, \forall s_1 \in \mathcal{S}_1$.

Then we have: $\exists \pi_*$*, where s.t.* $p_{\pi_*}(\tau) = \frac{1}{|\mathcal{T}|}, \forall \tau \in \mathcal{T}$*. It means that a deterministic uniformly optimal policy always exists for this MDP.*

Proof. We can prove this by construction. The following analysis applies for any $\tau \in \mathcal{T}$.

$$
\begin{aligned}
& p_{\pi_*}(\tau) = \frac{1}{|\mathcal{T}|} \\
\iff & \log p_{\pi_*}(\tau) = -\log |\mathcal{T}| \\
\iff & \log p(s_1) + \sum_{t=1}^{T} \log p(s_{t+1}|s_t, a_t) + \sum_{t=1}^{T} \log \pi_*(a_t|s_t) = -\log |\mathcal{T}| \\
\iff & \sum_{t=1}^{T} \log \pi_*(a_t|s_t) = -\log p(s_1) - \sum_{t=1}^{T} \log p(s_{t+1}|s_t, a_t) - \log |\mathcal{T}|
\end{aligned}
$$

where we use a_t, s_t as abbreviations of $a(\tau, t), s(\tau, t)$.

We denote $D(\tau) = -\log p(s_1) - \sum_{t=1}^{T} \log p(s_{t+1}|s_t, a_t) > 0$

$\iff \sum_{t=1}^{T} \log \pi_*(a_t|s_t) = D(\tau) - \log |\mathcal{T}|$

$\therefore$ we can obtain a uniformly optimal policy by solving the nonlinear programming:

$$\sum_{t=1}^{T} \log \pi_*(a(\tau, t)|s(\tau, t)) = D(\tau) - \log |\mathcal{T}| \ \forall \tau \in \mathcal{T} \tag{A.22}$$

$$\log \pi_*(a(\tau, t)|s(\tau, t)) = 0, \ \forall \tau \in \mathcal{T}, t = 1, ..., T \tag{A.23}$$

$$\sum_{i=1}^{m} \pi_*(a_i|s(\tau, t)) = 1, \ \forall \tau \in \mathcal{T}, t = 1, ..., T \tag{A.24}$$

Use the condition $p(s_1) = \frac{1}{|\mathcal{T}|}$, then we have:

$$\because \sum_{t=1}^{T} \log \pi_*(a(\tau, t)|s(\tau, t)) \tag{A.25}$$

$$= \sum_{t=1}^{T} \log 1 = 0 \text{ (LHS of } Eq \ (A.22)) \tag{A.26}$$

$$\because -\log p(s_1) - \sum_{t=1}^{T} \log p(s_{t+1}|s_t, a_t) - \log |\mathcal{T}| = \log |\mathcal{T}| - 0 - \log |\mathcal{T}| = 0 \tag{A.27}$$

$$\text{(RHS of } Eq \ (A.22)) \tag{A.28}$$

$$\therefore D(\tau) - \log |\mathcal{T}| = \sum_{t=1}^{T} \log \pi_*(a(\tau, t)|s(\tau, t)), \ \forall \tau \in \mathcal{T}.$$

Also the deterministic optimal policy satisfies the conditions in Eq (A.23 A.24). Therefore,

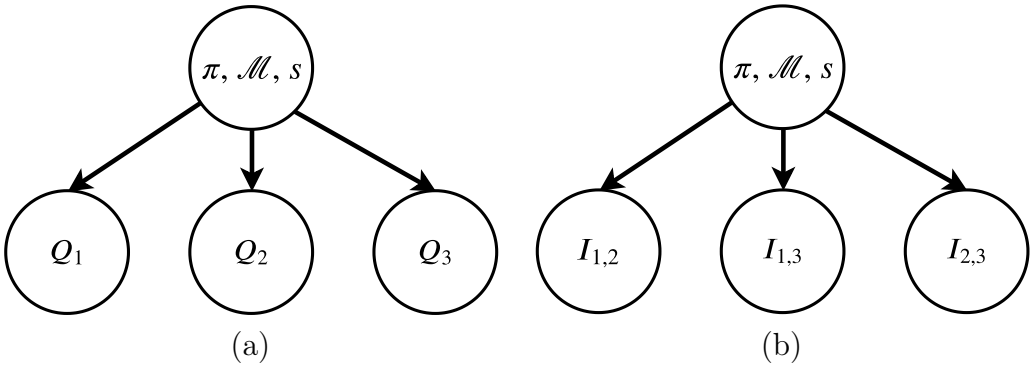


Figure A.2: The directed graph that describes the conditional independence of pairwise relationship of actions, where Q_1 denotes the return of taking action a_1 at state s, following policy π in $\mathcal{M}$, i.e., $Q^{\pi}_{\mathcal{M}}(s, a_1)$. $I_{1,2}$ is a random variable that denotes the pairwise relationship of Q_1 and Q_2, i.e., $I_{1,2} = 1,$ i.i.f. $Q_1 \geq Q_2,$ o.w. $I_{1,2} = 0$.

the deterministic optimal policy is a uniformly optimal policy. This lemma describes one type of MDP in which UOP exists. From the above reasoning, we can see that as long as the system of non-linear equations Eq (A.22 A.23 A.24) has a solution, the uniformly (near)-optimal policy exists. □

Lemma 7 (Hit optimal trajectory)**.** *The probability that a specific optimal trajectory was not encountered given an arbitrary softmax policy π_θ is exponentially decreasing with respect to the number of training episodes. No matter a MDP has deterministic or probabilistic dynamics.*

Proof. Given a specific optimal trajectory $\tau = \{s(\tau, t), a(\tau, t)\}_{t=1}^{T}$, and an arbitrary stationary policy π_θ, the probability that has never encountered at the n-th episode is $[1 - p_\theta(\tau)]^n = \xi^n$, based on lemma 5, we have $p_\theta(\tau) > 0$, therefore we have $\xi \in [0, 1)$. □

Discussions of Assumption 4

Intuitively, given a state and a stationary policy π, the relative relationships among actions can be independent, considering a fixed MDP $\mathcal{M}$. The relative relationship among actions is the relative relationship of actions' return. Starting from the same state, following a

stationary policy, the actions' return is determined by MDP properties such as environment dynamics, reward function, etc.

More concretely, we consider a MDP with three actions (a_1, a_2, a_3) for each state. The action value $Q^{\pi}_{\mathcal{M}}$ satisfies the Bellman equation in Eq (A.29). Notice that in this subsection, we use $Q^{\pi}_{\mathcal{M}}$ to denote the action value that estimates the absolute value of return in $\mathcal{M}$.

$$Q^{\pi}_{\mathcal{M}}(s, a_i) = r(s, a_i) + \max_a \mathbf{E}_{s' \sim p(*|s,a)} Q^{\pi}_{\mathcal{M}}(s', a), \forall i = 1, 2, 3. \tag{A.29}$$

As we can see from Eq (A.29), $Q^{\pi}_{\mathcal{M}}(s, a_i), i = 1, 2, 3$ is only related to s, π, and environment dynamics $\mathbf{P}$. It means if π, $\mathcal{M}$ and s are given, the action values of three actions are determined. Therefore, we can use a directed graph [21] to model the relationship of action values, as shown in Figure A.2 (a). Similarly, if we only consider the ranking of actions, this ranking is consistent with the relationship of actions' return, which is also determined by s, π, and $\mathbf{P}$. Therefore, the pairwise relationship among actions can be described as the directed graph in Figure A.2 (b), which establishes the conditional independence of actions' pairwise relationship. Based on the above reasoning, we conclude that Assumption 4 is realistic.

The proof of Theorem 6

Proof. The proof mainly establishes on the proof for long term performance Theorem 5 and connects the generalization bound in PAC framework to the lower bound of return. We construct a hybrid policy based on pairwise ranking policy Eq (4.5) as follows:

If $\pi_*(s) = \arg\max_a \lambda_\theta(s,a)$,

$$p_h(a|s) = \begin{cases} 1, \pi_*(s) = \arg\max_a \lambda_\theta(s,a) \\ 0, o.w. \end{cases} \tag{A.30}$$

If $\pi_*(s) \neq \arg\max_a \lambda_\theta(s,a)$,

$$p_h(a|s) = \pi_\theta(a|s) = \Pi_{j\neq i, j=1}^{m} p_{ij} \tag{A.31}$$

In plain English, the hybrid policy can be described as follows: for a state s and the policy parameter θ, if the action chosen by UOP has the highest relative action value (i.e., $\pi_*(s) = \arg\max_a \lambda_\theta(s,a)$), we use the deterministic policy as defined in Eq (A.30) for this state. Otherwise, we use the stochastic policy as defined in Eq (A.31). Note that the construction of this policy assume we have access to the UOP π_*, which is feasible in our setting. With TRS 6, we can filter all unique optimal trajectories following UOP. Therefore, when UOP is deterministic, for each state, we have the action that is chosen by the UOP.

We study the generalization performance and sample complexity of the pairwise ranking policy as follows:

$$\begin{gathered} \log(\frac{1}{|\mathcal{T}|}\sum\nolimits_{\tau\in\mathcal{T}} p_\theta(\tau)w(\tau)) \geq \frac{1}{|\mathcal{T}|}\sum_{\tau\in\mathcal{T}} \log p_\theta(\tau)w(\tau) \\ \Leftrightarrow \sum\nolimits_{\tau\in\mathcal{T}} p_\theta(\tau)w(\tau) \geq |\mathcal{T}| \exp(\frac{1}{|\mathcal{T}|}\sum_{\tau\in\mathcal{T}} \log p_\theta(\tau)w(\tau)) \\ \text{denote } F = \sum\nolimits_{\tau} p_\theta(\tau)w(\tau) = \sum\nolimits_{\tau\in\mathcal{T}} p_\theta(\tau)w(\tau) \end{gathered} \tag{A.32}$$

$$\Leftrightarrow F \geq |\mathcal{T}| \exp(\frac{1}{|\mathcal{T}|} \sum\nolimits_{\tau \in \mathcal{T}} \log p_\theta(\tau) w(\tau))$$
$$= |\mathcal{T}| \exp\left(\frac{1}{|\mathcal{T}|} \sum\nolimits_{\tau \in \mathcal{T}} \left(\log p(s_1) + \sum\nolimits_{t=1}^{T} \log p(s_{t+1}|s_t, a_t) + \sum\nolimits_{t=1}^{T} \log p_h(a_t|s_t) + \log w(\tau)\right)\right) \tag{A.33}$$

$$\because w(\tau) = 1,\ \forall \tau \in \mathcal{T}, s_t = s(\tau, t), a_t = a(\tau, t), t = 1, \ldots, T$$
$$= |\mathcal{T}| \exp\left(\frac{1}{|\mathcal{T}|} \sum\nolimits_{\tau \in \mathcal{T}} \left(\log p(s_1) + \sum\nolimits_{t=1}^{T} \log p(s_{t+1}|s_t, a_t) + \sum_{t=1}^{T} \log p_h(a_t|s_t)\right)\right)$$
$$= |\mathcal{T}| \exp\left(\frac{1}{|\mathcal{T}|} \sum_{\tau \in \mathcal{T}} (\log p(s_1) + \sum\nolimits_{t=1}^{T} \log p(s_{t+1}|s_t, a_t))\right) \exp\left(\frac{1}{|\mathcal{T}|} \sum\nolimits_{\tau \in \mathcal{T}} (\sum\nolimits_{t=1}^{T} \log p_h(a_t|s_t))\right) \tag{A.34}$$

Denote the dynamics of a trajectory as $p_d(\tau) = p(s_1) \Pi_{t=1}^{T} p(s_{t+1}|s_t, a_t)$

Notice that $p_d(\tau)$ is environment dynamics, which is fixed given a specific MDP.

$$\Leftrightarrow F \geq |\mathcal{T}| \exp\left(\frac{1}{|\mathcal{T}|} \sum\nolimits_{\tau \in \mathcal{T}} \log p_d(\tau)\right) \exp\left(\frac{1}{|\mathcal{T}|} \sum\nolimits_{\tau \in \mathcal{T}} (\sum\nolimits_{t=1}^{T} \log p_h(a_t|s_t))\right)$$
$$= |\mathcal{T}| \left(\Pi_{\tau \in \mathcal{T}} p_d(\tau)\right)^{\frac{1}{|\mathcal{T}|}} \exp\left(\frac{1}{|\mathcal{T}|T} \sum\nolimits_{\tau \in \mathcal{T}} (\sum\nolimits_{t=1}^{T} \log p_h(a_t|s_t)) T\right)$$

Use the same reasoning from Eq (A.7) to Eq (A.10).

$$= |\mathcal{T}| \left(\Pi_{\tau \in \mathcal{T}} p_d(\tau)\right)^{\frac{1}{|\mathcal{T}|}} \exp\left(T \sum\nolimits_{s,a} p_{\pi_*}(s, a) \log p_h(a|s)\right)$$
$$= |\mathcal{T}| \left(\Pi_{\tau \in \mathcal{T}} p_d(\tau)\right)^{\frac{1}{|\mathcal{T}|}} \exp(TL)$$

We denote $L = \sum\limits_{s,a} p_{\pi_*}(s, a) \log p_h(a|s)$.

L is the only term that is related to the policy parameter θ

Given $h = \pi_\theta$, misclassified state action pairs set $U_w = \{s, a | h(s) \neq a, (s, a) \sim p_*(s, a)\}$

$$L = \sum\nolimits_{s,a \in U_w} p_{\pi_*}(s, a) \log p_h(a|s) + \sum\nolimits_{s,a \notin U_w} p_{\pi_*}(s, a) \log p_h(a|s)$$

By definition of $U_w, \forall s, a \notin U_w, h(s) = a, \therefore p_h(a|s) = 1.$ (A.35)

$$= \sum\nolimits_{s,a \in U_w} p_{\pi_*}(s, a) \log \pi_\theta(a|s)$$

Since we use RPG as our policy parameterization, then with Eq (4.5)

$$= \sum\nolimits_{s,a \in U_w} p_{\pi_*}(s, a) \log(\Pi_{j \neq i, j=1}^{m} p_{ij})$$

$$= \sum\nolimits_{s,a_i \in U_w} p_{\pi_*}(s, a_i) \sum\nolimits_{j \neq i, j=1}^{m} \log \frac{1}{1 + e^{Q_{ji}}}$$

By Condition 1, which can be easily satisfied in practice. Then we have: $Q_{ij} < 2c_q \leq 1$

Apply Lemma 1, the misclassified rate is at most η.

$$\geq \sum\nolimits_{s,a_i \in U_w} p_{\pi_*}(s, a_i)(m - 1) \log(\frac{1}{1 + e})$$

$$\geq -\sum\nolimits_{s,a_i \in U_w} p_{\pi_*}(s, a_i)(m - 1) \log(1 + e)$$

$$\geq -\eta(m - 1) \log(1 + e)$$

$$= \eta(1 - m) \log(1 + e)$$

$$F \geq |\mathcal{T}| \left(\Pi_{\tau \in \mathcal{T}} p_d(\tau)\right)^{\frac{1}{|\mathcal{T}|}} \exp(TL)$$

$$\geq |\mathcal{T}| \left(\Pi_{\tau \in \mathcal{T}} p_d(\tau)\right)^{\frac{1}{|\mathcal{T}|}} \exp(\eta(1 - m)T \log(1 + e))$$

$$\geq |\mathcal{T}| \left(\Pi_{\tau \in \mathcal{T}} p_d(\tau)\right)^{\frac{1}{|\mathcal{T}|}} (1 + e)^{\eta(1-m)T}$$

$$= D(1 + e)^{\eta(1-m)T}$$

From generalization performance to sample complexity:

$$
\begin{aligned}
&\text{Set } 1-\epsilon = D(1+e)^{\eta(1-m)T}, \text{where } D = |\mathcal{T}| \left(\Pi_{\tau\in\mathcal{T}} p_d(\tau)\right)^{\frac{1}{|\mathcal{T}|}} \\
&\eta = \frac{\log_{1+e}\frac{D}{1-\epsilon}}{(m-1)T} \\
&\text{With realizable assumption 11, } \epsilon_{\min} = 0 \\
&\gamma = \frac{\eta - \epsilon_{\min}}{2} = \frac{\eta}{2} \\
n &\geq \frac{1}{2\gamma^2}\log\frac{2|\mathcal{H}|}{\delta} \\
&= \frac{2(m-1)^2T^2}{\left(\log_{1+e}\frac{D}{1-\epsilon}\right)^2}\log\frac{2|\mathcal{H}|}{\delta}
\end{aligned}
$$

Bridge the long-term reward and long-term performance:

$$
\begin{aligned}
&\sum\nolimits_{\tau} p_\theta(\tau) r(\tau) \text{ In Section 4.2.7, } r(\tau)\in[0,1], \forall\tau. \\
\geq &\sum\nolimits_{\tau} p_\theta(\tau) w(\tau) \text{ Since we focus on UOP Def 8, c = 1 in TSR Def 6} \\
= &\sum\nolimits_{\tau\in\mathcal{T}} p_\theta(\tau) w(\tau) \\
\geq &1-\epsilon
\end{aligned}
$$

This thus concludes the proof. □

Assumption 11 (Realizable)**.** *We assume there exists a hypothesis $h_* \in \mathcal{H}$ that obtains zero expected risk, i.e. $\exists h_* \in \mathcal{H} \;\Rightarrow\; \sum_{s,a} p_{\pi_*}(s,a)\mathbf{1}\{h_*(s)\neq a\} = 0$.*

The Assumption 11 is not necessary for the proof of Theorem 6. For the proof of Corollary 4, we introduce this assumption to achieve more concise conclusion. In finite

MDP, the realizable assumption can be satisfied if the policy is parameterized by multi-layer neural network, due to its perfect finite sample expressivity [224]. It is also advocated in our empirical studies since the neural network achieved optimal performance in PONG.

The proof of Lemma 2

Proof. Let $e_{=i}$ denotes the event $n = i|k$, i.e. the number of different optimal trajectories in first k episodes is equal to i. Similarly, $e_{\geq i}$ denotes the event $n \geq i|k$. Since the events $e_{=i}$ and $e_{=j}$ are mutually exclusive when $i \neq j$. Therefore, $p(e_{\geq i}) = p(e_{=i}, e_{=i+1}, ..., e_{=|\mathcal{T}|}) = \sum_{j=i}^{|\mathcal{T}|} p(e_{=j})$. Further more, we know that $\sum_{i=0}^{\mathcal{T}} p(e_{=i}) = 1$ since $\{e_{=i}, i = 0, ..., |\mathcal{T}|\}$ constructs an universal set. For example, $p(e_{\geq 1}) = p_{\pi_r,\mathcal{M}}(n \geq 1|k) = 1 - p_{\pi_r,\mathcal{M}}(n = 0|k) = 1 - (\frac{N-|\mathcal{T}|}{N})^k$.

$$
\begin{aligned}
p_{\pi_r,\mathcal{M}}(n \geq i|k) &= 1 - \sum_{i'=0}^{i-1} p_{\pi,\mathcal{M}}(n = i'|k) \\
&= 1 - \sum_{i'=0}^{i-1} C_{|\mathcal{T}|}^{i'} \frac{\sum_{j=0}^{i'} (-1)^j C_{i'}^j (N - |\mathcal{T}| + i' - j)^k}{N^k}
\end{aligned} \tag{A.36}
$$

In Eq (A.36), we use the inclusion-exclusion principle [93] to have the following equality.

$$
\begin{aligned}
p_{\pi_r,\mathcal{M}}(n = i'|k) &= C_{|\mathcal{T}|}^{i'} p(e_{\tau_1,\tau_2,...,\tau_{i'}}) \\
&= C_{|\mathcal{T}|}^{i'} \frac{\sum_{j=0}^{i'} (-1)^j C_{i'}^j (N - |\mathcal{T}| + i' - j)^k}{N^k}
\end{aligned}
$$

$e_{\tau_1,\tau_2,...,\tau_{i'}}$ denotes the event: in first k episodes, a certain set of i' optimal trajectories $\tau_1, \tau_2, ..., \tau_{i'}, i' \leq |\mathcal{T}|$ is sampled. □

Table A.2: Hyperparameters of RPG network

Hyperparameters	Value
Architecture	Conv(32-8×8-4) -Conv(64-4×4-2) -Conv(64-3×3-2) -FC(512)
Learning rate	0.0000625
Batch size	32
Replay buffer size	1000000
Update period	4
Margin in Eq (4.14)	1

The proof of Corollary 5

Proof. The Corollary 5 is a direct application of Lemma 2 and Theorem 6. First, we reformat Theorem 6 as follows:

$$p(A|B) \geq 1 - \delta$$

where event A denotes $\sum_\tau p_\theta(\tau) r(\tau) \geq D(1+e)^{\eta(1-m)T}$, event B denotes the number of state-action pairs n' from UOP (Def 8) satisfying $n' \geq n$, given fixed δ. With Lemma 2, we have $p(B) \geq p_{\pi_r,\mathcal{M}}(n' \geq n|k)$. Then, $P(A) = P(A|B)P(B) \geq (1-\delta)p_{\pi_r,\mathcal{M}}(n' \geq n|k)$.

$$\begin{aligned}
&\text{Set } (1-\delta)p_{\pi_r,\mathcal{M}}(n' \geq n|k) = 1 - \delta' \\
&\text{we have } P(A) \geq 1 - \delta' \\
&\delta = 1 - \frac{1-\delta'}{p_{\pi_r,\mathcal{M}}(n' \geq n|k)} \\
\eta &= 2\sqrt{\frac{1}{2n}\log\frac{2|\mathcal{H}|}{\delta}} \\
&= 2\sqrt{\frac{1}{2n}\log\frac{2|\mathcal{H}|p_{\pi_r,\mathcal{M}}(n' \geq n|k)}{p_{\pi_r,\mathcal{M}}(n' \geq n|k) - 1 + \delta'}}
\end{aligned}$$

□

Hyperparameters

We present the training details of ranking policy gradient in Table A.2. The network architecture is the same as the convolution neural network used in DQN [132]. We update the RPG network every four timesteps with a minibatch of size 32. The replay ratio is equal to eight for all baselines and RPG (except for ACER we use the default setting in openai baselines [48] for better performance).

Appendix B

Federated Learning

Additional Notations

In this section, we introduce additional notations that are used throughout the proof. Following common practice [178, 110], we define two virtual sequences $\overline{\mathbf{v}}_t$ and $\overline{\mathbf{w}}_t$. For full device participation and $t \notin \mathcal{I}_E$, $\overline{\mathbf{v}}_t = \overline{\mathbf{w}}_t = \sum_{k=1}^N p_k \mathbf{v}_t^k$. For partial participation, $t \in \mathcal{I}_E$, $\overline{\mathbf{w}}_t \neq \overline{\mathbf{v}}_t$ since $\overline{\mathbf{v}}_t = \sum_{k=1}^N p_k \mathbf{v}_t^k$ while $\overline{\mathbf{w}}_t = \sum_{k \in \mathcal{S}_t} \mathbf{w}_t^k$. However, we can set unbiased sampling strategy such that $\mathbb{E}_{\mathcal{S}_t} \overline{\mathbf{w}}_t = \overline{\mathbf{v}}_t$. $\overline{\mathbf{v}}_{t+1}$ is one-step SGD from $\overline{\mathbf{w}}_t$.

$$\overline{\mathbf{v}}_{t+1} = \overline{\mathbf{w}}_t - \eta_t \mathbf{g}_t, \tag{B.1}$$

where $\mathbf{g}_t = \sum_{k=1}^N p_k \mathbf{g}_{t,k}$ is one-step stochastic gradient, averaged over all devices.

$$\mathbf{g}_{t,k} = \nabla F_k \left(\mathbf{w}_t^k, \xi_t^k \right),$$

Similarly, we denote the expected one-step gradient $\overline{\mathbf{g}}_t = \mathbb{E}_{\xi_t}[\mathbf{g}_t] = \sum_{k=1}^N p_k \mathbb{E}_{\xi_t^k} \mathbf{g}_{t,k}$, where

$$\mathbb{E}_{\xi_t^k} \mathbf{g}_{t,k} = \nabla F_k \left(\mathbf{w}_t^k \right), \tag{B.2}$$

and $\xi_t = \{\xi_t^k\}_{k=1}^N$ denotes random samples at all devices at time step t. Since in this work, we also consider the case of partial participation. The sampling strategy to approximate the system heterogeneity can also affect the convergence. Here we follow the prior arts [75] considering two types of sampling schemes. The sampling scheme I establishes $\mathcal{S}_{t+1}$ by i.i.d. sampling the devices with replacement, in this case the upper bound of expected square norm of $\overline{\mathbf{w}}_{t+1} - \overline{\mathbf{v}}_{t+1}$ is given by [110, Lemma 5]:

$$\mathbb{E}_{\mathcal{S}_{t+1}} \left\| \overline{\mathbf{w}}_{t+1} - \overline{\mathbf{v}}_{t+1} \right\|^2 \leq \frac{4}{K} \eta_t^2 E^2 G^2. \tag{B.3}$$

The sampling scheme II establishes $\mathcal{S}_{t+1}$ by uniformly sampling all devices without replacement, in which we have the

$$\mathbb{E}_{\mathcal{S}_{t+1}} \left\| \overline{\mathbf{w}}_{t+1} - \overline{\mathbf{v}}_{t+1} \right\|^2 \leq \frac{4(N-K)}{K(N-1)} \eta_t^2 E^2 G^2. \tag{B.4}$$

We denote this upper bound as follows for concise presentation.

$$\mathbb{E}_{\mathcal{S}_{t+1}} \left\| \overline{\mathbf{w}}_{t+1} - \overline{\mathbf{v}}_{t+1} \right\|^2 \leq \eta_t^2 C. \tag{B.5}$$

Comparison of Convergence Rates with Related Works

In this section, we compare our convergence rate with the best-known results in the literature (see Table B.1). In [75], the authors provide $\mathcal{O}(1/NT)$ convergence rate of non-convex problems under Polyak-Łojasiewicz (PL) condition, which means their results can directly apply to the strongly convex problems. However, their assumption is based on bounded

gradient diversity, defined as follows:

$$\Lambda(\mathbf{w}) = \frac{\sum_k p_k \|\nabla F_k(\mathbf{w})\|_2^2}{\|\sum_k p_k \nabla F_k(\mathbf{w})\|_2^2} \leq B$$

This is a more restrictive assumption comparing to assuming bounded gradient under the case of target accuracy $\epsilon \to 0$ and PL condition. To see this, consider the gradient diversity at the global optimal $\mathbf{w}^*$, i.e., $\Lambda(\mathbf{w}^*) = \frac{\sum_k p_k \|\nabla F_k(\mathbf{w})\|_2^2}{\|\sum_k p_k \nabla F_k(\mathbf{w})\|_2^2}$. For $\Lambda(\mathbf{w}^*)$ to be bounded, it requires $\|\nabla F_k(\mathbf{w}^*)\|_2^2 = 0, \forall\, k$. This indicates $\mathbf{w}^*$ is also the minimizer of each local objective, which contradicts to the practical setting of heterogeneous data. Therefore, their bound is not effective for arbitrary small ϵ-accuracy under general heterogeneous data while our convergence results still hold in this case.

Table B.1: A high-level summary of the convergence results in this paper compared to prior state-of-the-art FL algorithms. This table only highlights the dependence on T (number of iterations), E (the maximal number of local steps), N (the total number of devices), and $K \leq N$ the number of participated devices. κ is the condition number of the system and $\beta \in (0, 1)$. We denote Nesterov accelerated FedAvg as N-FedAvg in this table.

Reference	Convergence rate	E	NonIID	Participation	Extra Assumptions	Setting
FedAvg[110]	$\mathcal{O}(\frac{E^2}{T})$	$\mathcal{O}(1)$	✓	Partial	Bounded gradient	Strongly convex
FedAvg[75]	$\mathcal{O}(\frac{1}{KT})$	$\mathcal{O}(K^{-1/3}T^{2/3})^\dagger$	✓‡‡	Partial	Bounded gradient diversity	Strongly convex§
FedAvg[104]	$\mathcal{O}(\frac{1}{NT})$	$\mathcal{O}(N^{-1/2}T^{1/2})$	✓	Full	Bounded gradient	Strongly convex
FedAvg/N-FedAvg	$\mathcal{O}(\frac{1}{KT})$	$\mathcal{O}(N^{-1/2}T^{1/2})^\ddagger$	✓	Partial	Bounded gradient	Strongly convex
FedAvg[98]	$\mathcal{O}(\frac{1}{\sqrt{NT}})$	$\mathcal{O}(N^{-3/2}T^{1/2})$	✓	Full	Bounded gradient	Convex
FedAvg[104]	$\mathcal{O}(\frac{1}{\sqrt{NT}})$	$\mathcal{O}(N^{-3/4}T^{1/4})$	✓	Full	Bounded gradient	Convex
FedAvg/N-FedAvg	$\mathcal{O}\left(\frac{1}{\sqrt{KT}}\right)$	$\mathcal{O}(N^{-3/4}T^{1/4})^\ddagger$	✓	Partial	Bounded gradient	Convex
FedAvg	$\mathcal{O}\left(\exp(-\frac{NT}{E\kappa_1})\right)$	$\mathcal{O}(T^\beta)$	✓	Partial	Bounded gradient	Overparameterized LR
FedMass	$\mathcal{O}\left(\exp(-\frac{NT}{E\sqrt{\kappa_1\bar{\kappa}}})\right)$	$\mathcal{O}(T^\beta)$	✓	Partial	Bounded gradient	Overparameterized LR

† This E is obtained under i.i.d. setting.
‡ This E is obtained under full participation setting.
§ In [75], the convergence rate is for non-convex smooth problems with PL condition, which also applies to strongly convex problems. Therefore, we compare it with our strongly convex results here.
‡‡ The bounded gradient diversity assumption is not applicable for general heterogeneous data when converging to arbitrarily small ϵ-accuracy (see discussions in Sec B).

Proof of Convergence Results for FedAvg

Strongly Convex Smooth Objectives

To facilitate reading, theorems from the main paper are restated and numbered identically.

We first summarize some properties of L-smooth and μ-strongly convex functions [161].

Lemma 8. *Let F be a convex L-smooth function. Then we have the following inequalities:*

1. *Quadratic upper bound:* $0 \leq F(\mathbf{w}) - F(\mathbf{w}') - \langle \nabla F(\mathbf{w}'), \mathbf{w} - \mathbf{w}' \rangle \leq \frac{L}{2}\|\mathbf{w} - \mathbf{w}'\|^2$.
2. *Coercivity:* $\frac{1}{L}\|\nabla F(\mathbf{w}) - \nabla F(\mathbf{w}')\|^2 \leq \langle \nabla F(\mathbf{w}) - \nabla F(\mathbf{w}'), \mathbf{w} - \mathbf{w}' \rangle$.
3. *Lower bound:* $F(\mathbf{w}) \geq F(\mathbf{w}') + \langle \nabla F(\mathbf{w}'), \mathbf{w} - \mathbf{w}' \rangle + \frac{1}{2L}\|\nabla F(\mathbf{w}) - \nabla F(\mathbf{w}')\|^2$. *In particular,* $\|\nabla F(\mathbf{w})\|^2 \leq 2L(F(\mathbf{w}) - F(\mathbf{w}^*))$.
4. *Optimality gap:* $F(\mathbf{w}) - F(\mathbf{w}^*) \leq \langle \nabla F(\mathbf{w}), \mathbf{w} - \mathbf{w}^* \rangle$.

Lemma 9. *Let F be a μ-strongly convex function. Then*

$$F(\mathbf{w}) \leq F(\mathbf{w}') + \langle \nabla F(\mathbf{w}'), \mathbf{w} - \mathbf{w}' \rangle + \frac{1}{2\mu}\|\nabla F(\mathbf{w}) - \nabla F(\mathbf{w}')\|^2$$

$$F(\mathbf{w}) - F(\mathbf{w}^*) \leq \frac{1}{2\mu}\|\nabla F(\mathbf{w})\|^2$$

Theorem 14. *Let $\overline{\mathbf{w}}_T = \sum_{k=1}^N p_k \mathbf{w}_T^k$, $\nu_{\max} = \max_k Np_k$, and set decaying learning rates $\alpha_t = \frac{1}{4\mu(\gamma+t)}$ with $\gamma = \max\{32\kappa, E\}$ and $\kappa = \frac{L}{\mu}$. Then under Assumptions 7,8,9,10 with full device participation,*

$$\mathbb{E}F(\overline{\mathbf{w}}_T) - F^* = \mathcal{O}\left(\frac{\kappa \nu_{\max}^2 \sigma^2/\mu}{NT} + \frac{\kappa^2 E^2 G^2/\mu}{T^2}\right)$$

and with partial device participation with at most K sampled devices at each communication

round,

$$\mathbb{E}F(\overline{\mathbf{w}}_T) - F^* = \mathcal{O}\left(\frac{\kappa E^2 G^2/\mu}{KT} + \frac{\kappa \nu_{\max}^2 \sigma^2/\mu}{NT} + \frac{\kappa^2 E^2 G^2/\mu}{T^2}\right)$$

Proof. The proof builds on ideas from [110]. The first step is to observe that the L-smoothness of F provides the upper bound

$$\begin{aligned}\mathbb{E}(F(\overline{\mathbf{w}}_t)) - F^* &= \mathbb{E}(F(\overline{\mathbf{w}}_t) - F(\mathbf{w}^*)) \\ &\leq \frac{L}{2}\mathbb{E}\|\overline{\mathbf{w}}_t - \mathbf{w}^*\|^2\end{aligned}$$

and bound $\mathbb{E}\|\overline{\mathbf{w}}_t - \mathbf{w}^*\|^2$.

Our main step is to prove the bound

$$\mathbb{E}\|\overline{\mathbf{w}}_{t+1} - \mathbf{w}^*\|^2 \leq (1 - \mu\alpha_t)\mathbb{E}\|\overline{\mathbf{w}}_t - \mathbf{w}^*\|^2 + \alpha_t^2 \frac{1}{N}\nu_{max}^2\sigma^2 + 5E^2 L\alpha_t^3 G^2$$

We have

$$\begin{aligned}\|\overline{\mathbf{w}}_{t+1} - \mathbf{w}^*\|^2 &= \|(\overline{\mathbf{w}}_t - \alpha_t \mathbf{g}_t) - \mathbf{w}^*\|^2 \\ &= \|(\overline{\mathbf{w}}_t - \alpha_t \overline{\mathbf{g}}_t - \mathbf{w}^*) - \alpha_t(\mathbf{g}_t - \overline{\mathbf{g}}_t)\|^2 \\ &= A_1 + A_2 + A_3\end{aligned}$$

where

$$A_1 = \|\overline{\mathbf{w}}_t - \mathbf{w}^* - \alpha_t \overline{\mathbf{g}}_t\|^2$$

$$A_2 = 2\alpha_t \langle \overline{\mathbf{w}}_t - \mathbf{w}^* - \alpha_t \overline{\mathbf{g}}_t, \overline{\mathbf{g}}_t - \mathbf{g}_t \rangle$$

$$A_3 = \alpha_t^2 \|\mathbf{g}_t - \overline{\mathbf{g}}_t\|^2$$

By definition of $\mathbf{g}_t$ and $\overline{\mathbf{g}}_t$ (see Eq (B.2)), we have $\mathbb{E}A_2 = 0$. For A_3, we have the follow upper bound:

$$\alpha_t^2 \mathbb{E}\|\mathbf{g}_t - \overline{\mathbf{g}}_t\|^2 = \alpha_t^2 \mathbb{E}\|\mathbf{g}_t - \mathbb{E}\mathbf{g}_t\|^2 = \alpha_t^2 \sum_{k=1}^N p_k^2 \|\mathbf{g}_{t,k} - \mathbb{E}\mathbf{g}_{t,k}\|^2 \leq \alpha_t^2 \sum_{k=1}^N p_k^2 \sigma_k^2$$

again by Jensen's inequality and using the independence of $\mathbf{g}_{t,k}, \mathbf{g}_{t,k'}$ [110, Lemma 2].

Next we bound A_1:

$$\|\overline{\mathbf{w}}_t - \mathbf{w}^* - \alpha_t \overline{\mathbf{g}}_t\|^2 = \|\overline{\mathbf{w}}_t - \mathbf{w}^*\|^2 + 2\langle \overline{\mathbf{w}}_t - \mathbf{w}^*, -\alpha_t \overline{\mathbf{g}}_t \rangle + \|\alpha_t \overline{\mathbf{g}}_t\|^2$$

and we will show that the third term $\|\alpha_t \overline{\mathbf{g}}_t\|^2$ can be canceled by an upper bound of the second term.

Now

$$\begin{aligned}
& - 2\alpha_t \langle \overline{\mathbf{w}}_t - \mathbf{w}^*, \overline{\mathbf{g}}_t \rangle \\
= & - 2\alpha_t \sum_{k=1}^N p_k \langle \overline{\mathbf{w}}_t - \mathbf{w}^*, \nabla F_k(\mathbf{w}_t^k) \rangle \\
= & - 2\alpha_t \sum_{k=1}^N p_k \langle \overline{\mathbf{w}}_t - \mathbf{w}_t^k, \nabla F_k(\mathbf{w}_t^k) \rangle - 2\alpha_t \sum_{k=1}^N p_k \langle \mathbf{w}_t^k - \mathbf{w}^*, \nabla F_k(\mathbf{w}_t^k) \rangle \\
\leq & - 2\alpha_t \sum_{k=1}^N p_k \langle \overline{\mathbf{w}}_t - \mathbf{w}_t^k, \nabla F_k(\mathbf{w}_t^k) \rangle + 2\alpha_t \sum_{k=1}^N p_k (F_k(\mathbf{w}^*) - F_k(\mathbf{w}_t^k)) \\
& - \alpha_t \mu \sum_{k=1}^N p_k \|\mathbf{w}_t^k - \mathbf{w}^*\|^2
\end{aligned}$$

$$
\begin{aligned}
&\leq 2\alpha_t \sum_{k=1}^{N} p_k \left[F_k(\mathbf{w}_t^k) - F_k(\overline{\mathbf{w}}_t) + \frac{L}{2}\|\overline{\mathbf{w}}_t - \mathbf{w}_t^k\|^2 + F_k(\mathbf{w}^*) - F_k(\mathbf{w}_t^k) \right] \\
&\quad - \alpha_t \mu \| \sum_{k=1}^{N} p_k \mathbf{w}_t^k - \mathbf{w}^* \|^2 \\
&= \alpha_t L \sum_{k=1}^{N} p_k \|\overline{\mathbf{w}}_t - \mathbf{w}_t^k\|^2 + 2\alpha_t \sum_{k=1}^{N} p_k \left[F_k(\mathbf{w}^*) - F_k(\overline{\mathbf{w}}_t) \right] - \alpha_t \mu \|\overline{\mathbf{w}}_t - \mathbf{w}^*\|^2
\end{aligned}
$$

For the second term, which is negative, we can ignore it, but this yields a suboptimal bound that fails to provide the desired linear speedup. Instead, we upper bound it using the following derivation:

$$
\begin{aligned}
&2\alpha_t \sum_{k=1}^{N} p_k \left[F_k(\mathbf{w}^*) - F_k(\overline{\mathbf{w}}_t) \right] \\
&\leq 2\alpha_t \left[F(\overline{\mathbf{w}}_{t+1}) - F(\overline{\mathbf{w}}_t) \right] \\
&\leq 2\alpha_t \mathbb{E} \langle \nabla F(\overline{\mathbf{w}}_t), \overline{\mathbf{w}}_{t+1} - \overline{\mathbf{w}}_t \rangle + \alpha_t L \mathbb{E} \|\overline{\mathbf{w}}_{t+1} - \overline{\mathbf{w}}_t\|^2 \\
&= -2\alpha_t^2 \mathbb{E} \langle \nabla F(\overline{\mathbf{w}}_t), \mathbf{g}_t \rangle + \alpha_t^3 L \mathbb{E} \|\mathbf{g}_t\|^2 \\
&= -2\alpha_t^2 \mathbb{E} \langle \nabla F(\overline{\mathbf{w}}_t), \overline{\mathbf{g}}_t \rangle + \alpha_t^3 L \mathbb{E} \|\mathbf{g}_t\|^2 \\
&= -\alpha_t^2 \left[\|\nabla F(\overline{\mathbf{w}}_t)\|^2 + \|\overline{\mathbf{g}}_t\|^2 - \|\nabla F(\overline{\mathbf{w}}_t) - \overline{\mathbf{g}}_t\|^2 \right] + \alpha_t^3 L \mathbb{E} \|\mathbf{g}_t\|^2 \\
&= -\alpha_t^2 \left[\|\nabla F(\overline{\mathbf{w}}_t)\|^2 + \|\overline{\mathbf{g}}_t\|^2 - \|\nabla F(\overline{\mathbf{w}}_t) - \sum_k p_k \nabla F(\mathbf{w}_t^k)\|^2 \right] + \alpha_t^3 L \mathbb{E} \|\mathbf{g}_t\|^2 \\
&\leq -\alpha_t^2 \left[\|\nabla F(\overline{\mathbf{w}}_t)\|^2 + \|\overline{\mathbf{g}}_t\|^2 - \sum_k p_k \|\nabla F(\overline{\mathbf{w}}_t) - \nabla F(\mathbf{w}_t^k)\|^2 \right] + \alpha_t^3 L \mathbb{E} \|\mathbf{g}_t\|^2 \\
&\leq -\alpha_t^2 \left[\|\nabla F(\overline{\mathbf{w}}_t)\|^2 + \|\overline{\mathbf{g}}_t\|^2 - L^2 \sum_k p_k \|\overline{\mathbf{w}}_t - \mathbf{w}_t^k\|^2 \right] + \alpha_t^3 L \mathbb{E} \|\mathbf{g}_t\|^2 \\
&\leq -\alpha_t^2 \|\overline{\mathbf{g}}_t\|^2 + \alpha_t^2 L^2 \sum_k p_k \|\overline{\mathbf{w}}_t - \mathbf{w}_t^k\|^2 + \alpha_t^3 L \mathbb{E} \|\mathbf{g}_t\|^2 - \alpha_t^2 \|\nabla F(\overline{\mathbf{w}}_t)\|^2
\end{aligned}
$$

where we have used the smoothness of F twice.

Note that the term $-\alpha_t^2\|\overline{\mathbf{g}}_t\|^2$ exactly cancels the $\alpha_t^2\|\overline{\mathbf{g}}_t\|^2$ in the bound for A_1, so that plugging in the bound for $-2\alpha_t\langle\overline{\mathbf{w}}_t - \mathbf{w}^*, \overline{\mathbf{g}}_t\rangle$, we have so far proved

$$\mathbb{E}\|\overline{\mathbf{w}}_{t+1} - \mathbf{w}^*\|^2 \leq \mathbb{E}(1-\mu\alpha_t)\|\overline{\mathbf{w}}_t - \mathbf{w}^*\|^2 + \alpha_t L\sum_{k=1}^N p_k\|\overline{\mathbf{w}}_t - \mathbf{w}_t^k\|^2 + \alpha_t^2\sum_{k=1}^N p_k^2\sigma_k^2$$
$$+ \alpha_t^2 L^2\sum_{k=1}^N p_k\|\overline{\mathbf{w}}_t - \mathbf{w}_t^k\|^2 + \alpha_t^3 L\mathbb{E}\|\mathbf{g}_t\|^2 - \alpha_t^2\|\nabla F(\overline{\mathbf{w}}_t)\|^2$$

The term $\mathbb{E}\|\mathbf{g}_t\|^2 \leq G^2$ by assumption.

Now we bound $\mathbb{E}\sum_{k=1}^N p_k\|\overline{\mathbf{w}}_t - \mathbf{w}_t^k\|^2$ following [110]. Since communication is done every E steps, for any $t \geq 0$, we can find a $t_0 \leq t$ such that $t - t_0 \leq E-1$ and $\mathbf{w}_{t_0}^k = \overline{\mathbf{w}}_{t_0}$for all k. Moreover, using α_t is non-increasing and $\alpha_{t_0} \leq 2\alpha_t$ for any $t - t_0 \leq E-1$, we have

$$\begin{aligned}
&\mathbb{E}\sum_{k=1}^N p_k\|\overline{\mathbf{w}}_t - \mathbf{w}_t^k\|^2 \\
&= \mathbb{E}\sum_{k=1}^N p_k\|\mathbf{w}_t^k - \overline{\mathbf{w}}_{t_0} - (\overline{\mathbf{w}}_t - \overline{\mathbf{w}}_{t_0})\|^2 \\
&\leq \mathbb{E}\sum_{k=1}^N p_k\|\mathbf{w}_t^k - \overline{\mathbf{w}}_{t_0}\|^2 \\
&= \mathbb{E}\sum_{k=1}^N p_k\|\mathbf{w}_t^k - \mathbf{w}_{t_0}^k\|^2 \\
&= \mathbb{E}\sum_{k=1}^N p_k\|-\sum_{i=t_0}^{t-1}\alpha_i\mathbf{g}_{i,k}\|^2 \\
&\leq 2\sum_{k=1}^N p_k\mathbb{E}\sum_{i=t_0}^{t-1}E\alpha_i^2\|\mathbf{g}_{i,k}\|^2 \\
&\leq 2\sum_{k=1}^N p_k E^2\alpha_{t_0}^2 G^2 \\
&\leq 4E^2\alpha_t^2 G^2
\end{aligned}$$

Using the bound on $\mathbb{E}\sum_{k=1}^{N} p_k \|\overline{\mathbf{w}}_t - \mathbf{w}_t^k\|^2$, we can conclude that, with $\nu_{max} := N \cdot \max_k p_k$ and $\nu_{min} := N \cdot \min_k p_k$,

$$
\begin{aligned}
&\mathbb{E}\|\overline{\mathbf{w}}_{t+1} - \mathbf{w}^*\|^2 \\
&\leq \mathbb{E}(1-\mu\alpha_t)\|\overline{\mathbf{w}}_t - \mathbf{w}^*\|^2 + 4E^2L\alpha_t^3G^2 \\
&\quad + 4E^2L^2\alpha_t^4G^2 + \alpha_t^2\sum_{k=1}^{N} p_k^2\sigma_k^2 + \alpha_t^3LG^2 \\
&= \mathbb{E}(1-\mu\alpha_t)\|\overline{\mathbf{w}}_t - \mathbf{w}^*\|^2 + 4E^2L\alpha_t^3G^2 \\
&\quad + 4E^2L^2\alpha_t^4G^2 + \alpha_t^2\frac{1}{N^2}\sum_{k=1}^{N} (p_kN)^2\sigma_k^2 + \alpha_t^3LG^2 \\
&\leq \mathbb{E}(1-\mu\alpha_t)\|\overline{\mathbf{w}}_t - \mathbf{w}^*\|^2 + 4E^2L\alpha_t^3G^2 \\
&\quad + 4E^2L^2\alpha_t^4G^2 + \alpha_t^2\frac{1}{N^2}\nu_{max}^2\sum_{k=1}^{N} \sigma_k^2 + \alpha_t^3LG^2 \\
&\leq \mathbb{E}(1-\mu\alpha_t)\|\overline{\mathbf{w}}_t - \mathbf{w}^*\|^2 + 6E^2L\alpha_t^3G^2 + \alpha_t^2\frac{1}{N}\nu_{max}^2\sigma^2
\end{aligned}
$$

where in the last inequality we use $\sigma^2 = \max_k \sigma_k^2$, and assume α_t satisfies $L\alpha_t \leq \frac{1}{8}$. We show next that $\mathbb{E}\|\overline{\mathbf{w}}_t - \mathbf{w}^*\|^2 = O(\frac{1}{tN} + \frac{E^2LG^2}{t^2})$.

Let $C \equiv 6E^2LG^2$ and $D \equiv \frac{1}{N}\nu_{max}^2\sigma^2$. Suppose that we have shown $\mathbb{E}\|\overline{\mathbf{w}}_t - \mathbf{w}^*\|^2 \leq b \cdot (\alpha_t D + \alpha_t^2 C)$ for some constant b and α_t. Then

$$
\begin{aligned}
&\mathbb{E}\|\overline{\mathbf{w}}_{t+1} - \mathbf{w}^*\|^2 \\
\leq& b(1-\mu\alpha_t)(\alpha_t D + \alpha_t^2 C) + \alpha_t^2 D + \alpha_t^3 C \\
=& (b(1-\mu\alpha_t) + \alpha_t)\alpha_t D + (b(1-\mu\alpha_t) + \alpha_t)\alpha_t^2 C
\end{aligned}
$$

and so it remains to choose α_t and b such that $(b(1-\mu\alpha_t)+\alpha_t)\alpha_t \leq b\alpha_{t+1}$ and $(b(1-\mu\alpha_t)+\alpha_t)\alpha_t^2 \leq b\alpha_{t+1}^2$. Recall that we require $\alpha_{t_0} \leq 2\alpha_t$ for any $t - t_0 \leq E-1$, and $L\alpha_t \leq \frac{1}{8}$. If we let $\alpha_t = \frac{4}{\mu(t+\gamma)}$ where $\gamma = \max\{E, 32\kappa\}$, then we may check that α_t satisfies both requirements.

Setting $b = \frac{4}{\mu}$, we have

$$\begin{aligned}
(b(1-\mu\alpha_t)+\alpha_t)\alpha_t &= \left(b(1-\frac{4}{t+\gamma}) + \frac{4}{\mu(t+\gamma)}\right)\frac{4}{\mu(t+\gamma)} \\
&= \left(b\frac{t+\gamma-4}{t+\gamma} + \frac{4}{\mu(t+\gamma)}\right)\frac{4}{\mu(t+\gamma)} \\
&= b(\frac{t+\gamma-3}{t+\gamma})\frac{4}{\mu(t+\gamma)} \\
&\leq b(\frac{t+\gamma-1}{t+\gamma})\frac{4}{\mu(t+\gamma)} \\
&\leq b\frac{4}{\mu(t+\gamma+1)} = b\alpha_{t+1}
\end{aligned}$$

and

$$\begin{aligned}
(b(1-\mu\alpha_t)+\alpha_t)\alpha_t^2 &= \left(b(1-\frac{4}{t+\gamma}) + \frac{4}{\mu(t+\gamma)}\right)\frac{16}{\mu^2(t+\gamma)^2} \\
&= \left(b\frac{t+\gamma-4}{t+\gamma} + \frac{4}{\mu(t+\gamma)}\right)\frac{16}{\mu^2(t+\gamma)^2} \\
&= b(\frac{t+\gamma-2}{t+\gamma})\frac{16}{\mu^2(t+\gamma)^2} \\
&\leq b\frac{16}{\mu^2(t+\gamma+1)^2} = b\alpha_{t+1}^2
\end{aligned}$$

where we have used the facts that

$$\frac{t+\gamma-1}{(t+\gamma)^2} \leq \frac{1}{(t+\gamma+1)}$$

$$\frac{t+\gamma-2}{(t+\gamma)^3} \leq \frac{1}{(t+\gamma+1)^2}$$

for $\gamma \geq 1$.

Thus we have shown

$$\mathbb{E}\|\overline{\mathbf{w}}_{t+1} - \mathbf{w}^*\|^2 \leq b \cdot (\alpha_{t+1} D + \alpha_{t+1}^2 C)$$

for our choice of α_t and b. Now to ensure

$$\begin{aligned}\|\mathbf{w}_0 - \mathbf{w}^*\|^2 &\leq b \cdot (\alpha_0 D + \alpha_0^2 C) \\ &= b \cdot (\frac{4}{\mu\gamma} D + \frac{16}{\mu^2\gamma^2} C)\end{aligned}$$

we can simply scale b by $c\|\mathbf{w}_0 - \mathbf{w}^*\|^2$ for a constant c large enough and the induction step still holds.

It follows that

$$\mathbb{E}\|\overline{\mathbf{w}}_t - \mathbf{w}^*\|^2 \leq c\|\mathbf{w}_0 - \mathbf{w}^*\|^2 \frac{4}{\mu}(D\alpha_t + C\alpha_t^2)$$

for all $t \geq 0$.

Finally, the L-smoothness of F implies

$$\begin{aligned}&\mathbb{E}(F(\overline{\mathbf{w}}_T)) - F^* \\ &= \mathbb{E}(F(\overline{\mathbf{w}}_T) - F(\mathbf{w}^*)) \\ &\leq \frac{L}{2}\mathbb{E}\|\overline{\mathbf{w}}_T - \mathbf{w}^*\|^2 \leq \frac{L}{2} c\|\mathbf{w}_0 - \mathbf{w}^*\|^2 \frac{4}{\mu}(D\alpha_T + C\alpha_T^2)\end{aligned}$$

$$
\begin{aligned}
&= 2c\|\mathbf{w}_0 - \mathbf{w}^*\|^2\kappa(D\alpha_T + C\alpha_T^2) \\
&\leq 2c\|\mathbf{w}_0 - \mathbf{w}^*\|^2\kappa\left[\frac{4}{\mu(T+\gamma)}\cdot\frac{1}{N}\nu_{max}^2\sigma^2 + 6E^2LG^2\cdot(\frac{4}{\mu(T+\gamma)})^2\right] \\
&= O(\frac{\kappa}{\mu}\frac{1}{N}\nu_{max}^2\sigma^2\cdot\frac{1}{T} + \frac{\kappa^2}{\mu}E^2G^2\cdot\frac{1}{T^2})
\end{aligned}
$$

With partial participation, the update at each communication round is now given by averages over a subset of sampled devices. When $t+1 \notin \mathcal{I}_E$, $\overline{\mathbf{v}}_{t+1} = \overline{\mathbf{w}}_{t+1}$, while when $t+1 \notin \mathcal{I}_E$, we have $\mathbb{E}\overline{\mathbf{w}}_{t+1} = \overline{\mathbf{v}}_{t+1}$ by design of the sampling schemes, so that

$$
\begin{aligned}
\mathbb{E}\|\overline{\mathbf{w}}_{t+1} - \mathbf{w}^*\|^2 &= \mathbb{E}\|\overline{\mathbf{w}}_{t+1} - \overline{\mathbf{v}}_{t+1} + \overline{\mathbf{v}}_{t+1} - \mathbf{w}^*\|^2 \\
&= \mathbb{E}\|\overline{\mathbf{w}}_{t+1} - \overline{\mathbf{v}}_{t+1}\|^2 + \mathbb{E}\|\overline{\mathbf{v}}_{t+1} - \mathbf{w}^*\|^2
\end{aligned}
$$

As before, $\mathbb{E}\|\overline{\mathbf{v}}_{t+1} - \mathbf{w}^*\|^2 \leq \mathbb{E}(1-\mu\alpha_t)\|\overline{\mathbf{w}}_t - \mathbf{w}^*\|^2 + 6E^2L\alpha_t^3G^2 + \alpha_t^2\frac{1}{N}\nu_{max}^2\sigma^2$.

The key is to bound $\mathbb{E}\|\overline{\mathbf{w}}_{t+1} - \overline{\mathbf{v}}_{t+1}\|^2$. For sampling scheme I we have

$$
\begin{aligned}
\mathbb{E}\|\overline{\mathbf{w}}_{t+1} - \overline{\mathbf{v}}_{t+1}\|^2 &= \frac{1}{K}\sum_k p_k\mathbb{E}\|\mathbf{w}_{t+1}^k - \overline{\mathbf{w}}_{t+1}\|^2 \\
&\leq \frac{4}{K}\alpha_t^2E^2G^2
\end{aligned}
$$

while for sampling scheme II

$$
\begin{aligned}
\mathbb{E}\|\overline{\mathbf{w}}_{t+1} - \overline{\mathbf{v}}_{t+1}\|^2 &= \frac{N-K}{N-1}\frac{1}{K}\sum_k p_k\mathbb{E}\|\mathbf{w}_{t+1}^k - \overline{\mathbf{w}}_{t+1}\|^2 \\
&\leq \frac{N-K}{N-1}\frac{4}{K}\alpha_t^2E^2G^2
\end{aligned}
$$

The same argument as the full participation case implies

$$\mathbb{E}F(\overline{\mathbf{w}}_T) - F^* = O(\frac{\kappa\nu_{\max}^2\sigma^2/\mu}{NT} + \frac{\kappa E^2G^2/\mu}{KT} + \frac{\kappa^2E^2G^2/\mu}{T^2})$$

□

One may ask whether the dependence on E in the term $\frac{\kappa E^2G^2/\mu}{KT}$ can be removed, or equivalently whether $\sum_k p_k\|\mathbf{w}_t^k - \overline{\mathbf{w}}_t\|^2 = O(1/T^2)$ can be independent of E. We provide a simple counterexample that shows that this is not possible in general.

Lemma 10. *There exists a dataset such that if $E = \mathcal{O}(T^\beta)$ for any $\beta > 0$ then $\sum_k p_k\|\mathbf{w}_t^k - \overline{\mathbf{w}}_t\|^2 = \Omega(\frac{1}{T^{2-2\beta}})$.*

Proof. Suppose that we have an even number of devices and each $F_k(\mathbf{w}) = \frac{1}{n_k}\sum_{j=1}^{n_k}(\mathbf{x}_k^j - \mathbf{w})^2$ contains data points $\mathbf{x}_k^j = \mathbf{w}^{*,k}$, with $n_k \equiv n$. Moreover, the $\mathbf{w}^{*,k}$'s come in pairs around the origin. As a result, the global objective F is minimized at $\mathbf{w}^* = 0$. Moreover, if we start from $\overline{\mathbf{w}}_0 = 0$, then by design of the dataset the updates in local steps exactly cancel each other at each iteration, resulting in $\overline{\mathbf{w}}_t = 0$ for all t. On the other hand, if $E = T^\beta$, then starting from any $t = \mathcal{O}(T)$ with constant step size $\mathcal{O}(\frac{1}{T})$, after E iterations of local steps, the local parameters are updated towards $\mathbf{w}^{*,k}$ with $\|\mathbf{w}_{t+E}^k\|^2 = \Omega((T^\beta \cdot \frac{1}{T})^2) = \Omega(\frac{1}{T^{2-2\beta}})$. This implies that

$$\begin{aligned}\sum_k p_k\|\mathbf{w}_{t+E}^k - \overline{\mathbf{w}}_{t+E}\|^2 &= \sum_k p_k\|\mathbf{w}_{t+E}^k\|^2 \\ &= \Omega(\frac{1}{T^{2-2\beta}})\end{aligned}$$

which is at a slower rate than $\frac{1}{T^2}$ for any $\beta > 0$. Thus the sampling variance $\mathbb{E}\|\overline{\mathbf{w}}_{t+1} -$

$\overline{\mathbf{v}}_{t+1}\|^2 = \Omega(\sum_k p_k \mathbb{E}\|\mathbf{w}_{t+1}^k - \overline{\mathbf{w}}_{t+1}\|^2)$ decays at a slower rate than $\frac{1}{T^2}$, resulting in a convergence rate slower than $O(\frac{1}{T})$ with partial participation. □

Convex Smooth Objectives

Theorem 15. *Under assumptions 7,9,10 and constant learning rate* $\alpha_t = \mathcal{O}(\sqrt{\frac{N}{T}})$,

$$\min_{t \leq T} F(\overline{\mathbf{w}}_t) - F(\mathbf{w}^*) = \mathcal{O}\left(\frac{\nu_{\max}\sigma^2}{\sqrt{NT}} + \frac{NE^2LG^2}{T}\right)$$

with full participation, and with partial device participation with K *sampled devices at each communication round and learning rate* $\alpha_t = \mathcal{O}(\sqrt{\frac{K}{T}})$,

$$\min_{t \leq T} F(\overline{\mathbf{w}}_t) - F(\mathbf{w}^*) = \mathcal{O}\left(\frac{\nu_{\max}\sigma^2}{\sqrt{KT}} + \frac{E^2G^2}{\sqrt{KT}} + \frac{KE^2LG^2}{T}\right)$$

Proof. We again start by bounding the term

$$\begin{aligned}
\|\overline{\mathbf{w}}_{t+1} - \mathbf{w}^*\|^2 &= \|(\overline{\mathbf{w}}_t - \alpha_t \mathbf{g}_t) - \mathbf{w}^*\|^2 \\
&= \|(\overline{\mathbf{w}}_t - \alpha_t \overline{\mathbf{g}}_t - \mathbf{w}^*) - \alpha_t(\mathbf{g}_t - \overline{\mathbf{g}}_t)\|^2 \\
&= A_1 + A_2 + A_3
\end{aligned}$$

where

$$A_1 = \|\overline{\mathbf{w}}_t - \mathbf{w}^* - \alpha_t \overline{\mathbf{g}}_t\|^2$$

$$A_2 = 2\alpha_t \langle \overline{\mathbf{w}}_t - \mathbf{w}^* - \alpha_t \overline{\mathbf{g}}_t, \overline{\mathbf{g}}_t - \mathbf{g}_t \rangle$$

$$A_3 = \alpha_t^2 \|\mathbf{g}_t - \overline{\mathbf{g}}_t\|^2$$

By definition of $\mathbf{g}_t$ and $\overline{\mathbf{g}}_t$ (see Eq (B.2)), we have $\mathbb{E}A_2 = 0$. For A_3, we have the follow upper bound:

$$\alpha_t^2 \mathbb{E}\|\mathbf{g}_t - \overline{\mathbf{g}}_t\|^2 = \alpha_t^2 \mathbb{E}\|\mathbf{g}_t - \mathbb{E}\mathbf{g}_t\|^2 = \alpha_t^2 \sum_{k=1}^{N} p_k^2 \|\mathbf{g}_{t,k} - \mathbb{E}\mathbf{g}_{t,k}\|^2 \leq \alpha_t^2 \sum_{k=1}^{N} p_k^2 \sigma_k^2$$

again by Jensen's inequality and using the independence of $\mathbf{g}_{t,k}, \mathbf{g}_{t,k'}$ [110, Lemma 2].

Next we bound A_1:

$$\|\overline{\mathbf{w}}_t - \mathbf{w}^* - \alpha_t \overline{\mathbf{g}}_t\|^2 = \|\overline{\mathbf{w}}_t - \mathbf{w}^*\|^2 + 2\langle \overline{\mathbf{w}}_t - \mathbf{w}^*, -\alpha_t \overline{\mathbf{g}}_t \rangle + \|\alpha_t \overline{\mathbf{g}}_t\|^2$$

Using the convexity and L-smoothness of F_k,

$$\begin{aligned}
&- 2\alpha_t \langle \overline{\mathbf{w}}_t - \mathbf{w}^*, \overline{\mathbf{g}}_t \rangle \\
&= -2\alpha_t \sum_{k=1}^{N} p_k \langle \overline{\mathbf{w}}_t - \mathbf{w}^*, \nabla F_k(\mathbf{w}_t^k) \rangle \\
&= -2\alpha_t \sum_{k=1}^{N} p_k \langle \overline{\mathbf{w}}_t - \mathbf{w}_t^k, \nabla F_k(\mathbf{w}_t^k) \rangle - 2\alpha_t \sum_{k=1}^{N} p_k \langle \mathbf{w}_t^k - \mathbf{w}^*, \nabla F_k(\mathbf{w}_t^k) \rangle \\
&\leq -2\alpha_t \sum_{k=1}^{N} p_k \langle \overline{\mathbf{w}}_t - \mathbf{w}_t^k, \nabla F_k(\mathbf{w}_t^k) \rangle + 2\alpha_t \sum_{k=1}^{N} p_k (F_k(\mathbf{w}^*) - F_k(\mathbf{w}_t^k)) \\
&\leq 2\alpha_t \sum_{k=1}^{N} p_k \left[F_k(\mathbf{w}_t^k) - F_k(\overline{\mathbf{w}}_t) + \frac{L}{2} \|\overline{\mathbf{w}}_t - \mathbf{w}_t^k\|^2 + F_k(\mathbf{w}^*) - F_k(\mathbf{w}_t^k) \right] \\
&= \alpha_t L \sum_{k=1}^{N} p_k \|\overline{\mathbf{w}}_t - \mathbf{w}_t^k\|^2 + 2\alpha_t \sum_{k=1}^{N} p_k \left[F_k(\mathbf{w}^*) - F_k(\overline{\mathbf{w}}_t) \right]
\end{aligned}$$

which results in

$$\|\overline{\mathbf{w}}_{t+1} - \mathbf{w}^*\|^2 \leq \|\overline{\mathbf{w}}_t - \mathbf{w}^*\|^2 + \alpha_t L \sum_{k=1}^{N} p_k \|\overline{\mathbf{w}}_t - \mathbf{w}_t^k\|^2$$

$$+ 2\alpha_t \sum_{k=1}^{N} p_k \left[F_k(\mathbf{w}^*) - F_k(\overline{\mathbf{w}}_t)\right] + \alpha_t^2 \|\overline{\mathbf{g}}_t\|^2 + \alpha_t^2 \sum_{k=1}^{N} p_k^2 \sigma_k^2$$

The difference of this bound with that in the strongly convex case is that we no longer have a contraction factor in front of $\|\overline{\mathbf{w}}_t - \mathbf{w}^*\|^2$. In the strongly convex case, we were able to cancel $\alpha_t^2 \|\overline{\mathbf{g}}_t\|^2$ with $2\alpha_t \sum_{k=1}^{N} p_k \left[F_k(\mathbf{w}^*) - F_k(\overline{\mathbf{w}}_t)\right]$ and obtain only lower order terms. In the convex case, we use a different strategy and preserve $\sum_{k=1}^{N} p_k \left[F_k(\mathbf{w}^*) - F_k(\overline{\mathbf{w}}_t)\right]$ in order to obtain a telescoping sum.

We have

$$\begin{aligned}
\|\overline{\mathbf{g}}_t\|^2 &= \| \sum_k p_k \nabla F_k(\mathbf{w}_t^k) \|^2 \\
&= \| \sum_k p_k \nabla F_k(\mathbf{w}_t^k) - \sum_k p_k \nabla F_k(\overline{\mathbf{w}}_t) + \sum_k p_k \nabla F_k(\overline{\mathbf{w}}_t) \|^2 \\
&\leq 2\| \sum_k p_k \nabla F_k(\mathbf{w}_t^k) - \sum_k p_k \nabla F_k(\overline{\mathbf{w}}_t) \|^2 + 2\| \sum_k p_k \nabla F_k(\overline{\mathbf{w}}_t) \|^2 \\
&\leq 2L^2 \sum_k p_k \|\mathbf{w}_t^k - \overline{\mathbf{w}}_t\|^2 + 2\| \sum_k p_k \nabla F_k(\overline{\mathbf{w}}_t) \|^2 \\
&= 2L^2 \sum_k p_k \|\mathbf{w}_t^k - \overline{\mathbf{w}}_t\|^2 + 2\|\nabla F(\overline{\mathbf{w}}_t)\|^2
\end{aligned}$$

using $\nabla F(\mathbf{w}^*) = 0$. Now using the L smoothness of F, we have $\|\nabla F(\overline{\mathbf{w}}_t)\|^2 \leq 2L(F(\overline{\mathbf{w}}_t) - F(\mathbf{w}^*))$, so that

$$\begin{aligned}
&\|\overline{\mathbf{w}}_{t+1} - \mathbf{w}^*\|^2 \\
&\leq \|\overline{\mathbf{w}}_t - \mathbf{w}^*\|^2 + \alpha_t L \sum_{k=1}^{N} p_k \|\overline{\mathbf{w}}_t - \mathbf{w}_t^k\|^2 + 2\alpha_t \sum_{k=1}^{N} p_k \left[F_k(\mathbf{w}^*) - F_k(\overline{\mathbf{w}}_t)\right] \\
&+ 2\alpha_t^2 L^2 \sum_k p_k \|\mathbf{w}_t^k - \overline{\mathbf{w}}_t\|^2 + 4\alpha_t^2 L(F(\overline{\mathbf{w}}_t) - F(\mathbf{w}^*)) + \alpha_t^2 \sum_{k=1}^{N} p_k^2 \sigma_k^2
\end{aligned}$$

$$= \|\overline{\mathbf{w}}_t - \mathbf{w}^*\|^2 + (2\alpha_t^2 L^2 + \alpha_t L) \sum_{k=1}^N p_k \|\overline{\mathbf{w}}_t - \mathbf{w}_t^k\|^2 + \alpha_t \sum_{k=1}^N p_k \left[F_k(\mathbf{w}^*) - F_k(\overline{\mathbf{w}}_t)\right]$$
$$+ \alpha_t^2 \sum_{k=1}^N p_k^2 \sigma_k^2 + \alpha_t(1 - 4\alpha_t L)(F(\mathbf{w}^*) - F(\overline{\mathbf{w}}_t))$$

Since $F(\mathbf{w}^*) \leq F(\overline{\mathbf{w}}_t)$, as long as $4\alpha_t L \leq 1$, we can ignore the last term, and rearrange the inequality to obtain

$$\begin{aligned}
&\|\overline{\mathbf{w}}_{t+1} - \mathbf{w}^*\|^2 + \alpha_t(F(\overline{\mathbf{w}}_t) - F(\mathbf{w}^*)) \\
&\leq \|\overline{\mathbf{w}}_t - \mathbf{w}^*\|^2 + (2\alpha_t^2 L^2 + \alpha_t L) \sum_{k=1}^N p_k \|\overline{\mathbf{w}}_t - \mathbf{w}_t^k\|^2 + \alpha_t^2 \sum_{k=1}^N p_k^2 \sigma_k^2 \\
&\leq \|\overline{\mathbf{w}}_t - \mathbf{w}^*\|^2 + \frac{3}{2}\alpha_t L \sum_{k=1}^N p_k \|\overline{\mathbf{w}}_t - \mathbf{w}_t^k\|^2 + \alpha_t^2 \sum_{k=1}^N p_k^2 \sigma_k^2
\end{aligned}$$

The same argument as before yields $\mathbb{E}\sum_{k=1}^N p_k \|\overline{\mathbf{w}}_t - \mathbf{w}_t^k\|^2 \leq 4E^2\alpha_t^2 G^2$ which gives

$$\begin{aligned}
\|\overline{\mathbf{w}}_{t+1} - \mathbf{w}^*\|^2 + \alpha_t(F(\overline{\mathbf{w}}_t) - F(\mathbf{w}^*)) &\leq \|\overline{\mathbf{w}}_t - \mathbf{w}^*\|^2 + \alpha_t^2 \sum_{k=1}^N p_k^2 \sigma_k^2 + 6\alpha_t^3 E^2 L G^2 \\
&\leq \|\overline{\mathbf{w}}_t - \mathbf{w}^*\|^2 + \alpha_t^2 \frac{1}{N} \nu_{\max}^2 \sigma^2 + 6\alpha_t^3 E^2 L G^2
\end{aligned}$$

Summing the inequalities from $t = 0$ to $t = T$, we obtain

$$\sum_{t=0}^T \alpha_t(F(\overline{\mathbf{w}}_t) - F(\mathbf{w}^*)) \leq \|\mathbf{w}_0 - \mathbf{w}^*\|^2 + \sum_{t=0}^T \alpha_t^2 \cdot \frac{1}{N} \nu_{\max}^2 \sigma^2 + \sum_{t=0}^T \alpha_t^3 \cdot 6E^2 L G^2$$

so that

$$\min_{t \leq T} F(\overline{\mathbf{w}}_t) - F(\mathbf{w}^*) \leq \frac{1}{\sum_{t=0}^T \alpha_t} \left(\|\mathbf{w}_0 - \mathbf{w}^*\|^2 + \sum_{t=0}^T \alpha_t^2 \cdot \frac{1}{N} \nu_{\max}^2 \sigma^2 + \sum_{t=0}^T \alpha_t^3 \cdot 6E^2 L G^2 \right)$$

By setting the constant learning rate $\alpha_t \equiv \sqrt{\frac{N}{T}}$, we have

$$
\begin{aligned}
&\min_{t\leq T} F(\overline{\mathbf{w}}_t) - F(\mathbf{w}^*) \\
\leq &\frac{1}{\sqrt{NT}} \cdot \|\mathbf{w}_0 - \mathbf{w}^*\|^2 + \frac{1}{\sqrt{NT}} T \cdot \frac{N}{T} \cdot \frac{1}{N} \nu_{\max}^2 \sigma^2 + \frac{1}{\sqrt{NT}} T (\sqrt{\frac{N}{T}})^3 6E^2 L G^2 \\
\leq &\frac{1}{\sqrt{NT}} \cdot \|\mathbf{w}_0 - \mathbf{w}^*\|^2 + \frac{1}{\sqrt{NT}} T \cdot \frac{N}{T} \cdot \frac{1}{N} \nu_{\max}^2 \sigma^2 + \frac{N}{T} 6E^2 L G^2 \\
= &(\|\mathbf{w}_0 - \mathbf{w}^*\|^2 + \nu_{\max}^2 \sigma^2) \frac{1}{\sqrt{NT}} + \frac{N}{T} 6E^2 L G^2 \\
= &O(\frac{\nu_{\max}^2 \sigma^2}{\sqrt{NT}} + \frac{NE^2 L G^2}{T})
\end{aligned}
$$

Similarly, for partial participation, we have

$$
\begin{aligned}
&\min_{t\leq T} F(\overline{\mathbf{w}}_t) - F(\mathbf{w}^*) \\
\leq &\frac{1}{\sum_{t=0}^{T} \alpha_t} \left(\|\mathbf{w}_0 - \mathbf{w}^*\|^2 + \sum_{t=0}^{T} \alpha_t^2 \cdot (\frac{1}{N} \nu_{\max} \sigma^2 + C) + \sum_{t=0}^{T} \alpha_t^3 \cdot 6E^2 L G^2 \right)
\end{aligned}
$$

where $C = \frac{4}{K} E^2 G^2$ or $\frac{N-K}{N-1} \frac{4}{K} E^2 G^2$, so that with $\alpha_t = \sqrt{\frac{K}{T}}$, we have

$$
\min_{t\leq T} F(\overline{\mathbf{w}}_t) - F(\mathbf{w}^*) = \mathcal{O}(\frac{\nu_{\max} \sigma^2}{\sqrt{KT}} + \frac{E^2 G^2}{\sqrt{KT}} + \frac{K E^2 L G^2}{T})
$$

□

Proof of Convergence Results for Nesterov Accelerated FedAvg

Strongly Convex Smooth Objectives

Theorem 16. *Let* $\overline{\mathbf{v}}_T = \sum_{k=1}^N p_k \mathbf{v}_T^k$ *and set learning rates* $\beta_{t-1} = \frac{3}{14(t+\gamma)(1-\frac{6}{t+\gamma})\max\{\mu,1\}}$, $\alpha_t = \frac{6}{\mu(t+\gamma)}$. *Then under Assumptions 7,8,9,10 with full device participation,*

$$\mathbb{E}F(\overline{\mathbf{v}}_T) - F^* = \mathcal{O}\left(\frac{\kappa\nu_{\max}\sigma^2/\mu}{NT} + \frac{\kappa^2E^2G^2/\mu}{T^2}\right),$$

and with partial device participation with K *sampled devices at each communication round,*

$$\mathbb{E}F(\overline{\mathbf{v}}_T) - F^* = \mathcal{O}\left(\frac{\kappa\nu_{\max}\sigma^2/\mu}{NT} + \frac{\kappa E^2G^2/\mu}{KT} + \frac{\kappa^2E^2G^2/\mu}{T^2}\right).$$

Proof. Define the virtual sequences $\overline{\mathbf{v}}_t = \sum_{k=1}^N p_k \mathbf{v}_t^k$, $\overline{\mathbf{w}}_t = \sum_{k=1}^N p_k w_t^k$, and $\overline{\mathbf{g}}_t = \sum_{k=1}^N p_k \mathbb{E}\mathbf{g}_{t,k}$. We have $\mathbb{E}\mathbf{g}_t = \overline{\mathbf{g}}_t$ and $\overline{\mathbf{v}}_{t+1} = \overline{\mathbf{w}}_t - \alpha_t\mathbf{g}_t$, and $\overline{\mathbf{w}}_{t+1} = \overline{\mathbf{v}}_{t+1}$ for all t. The proof again uses the L-smoothness of F to bound

$$\begin{aligned}\mathbb{E}(F(\overline{\mathbf{v}}_t)) - F^* &= \mathbb{E}(F(\overline{\mathbf{v}}_t) - F(\mathbf{w}^*)) \\ &\leq \frac{L}{2}\mathbb{E}\|\overline{\mathbf{v}}_t - \mathbf{w}^*\|^2\end{aligned}$$

Our main step is to prove the bound

$$\mathbb{E}\|\overline{\mathbf{v}}_{t+1} - \mathbf{w}^*\|^2 \leq (1-\mu\alpha_t)\mathbb{E}\|\overline{\mathbf{v}}_t - \mathbf{w}^*\|^2 + \alpha_t^2\frac{1}{N}\nu_{max}^2\sigma^2 + 20E^2L\alpha_t^3G^2$$

for appropriate step sizes α_t, β_t.

We have

$$\begin{aligned}\|\overline{\mathbf{v}}_{t+1} - \mathbf{w}^*\|^2 &= \|(\overline{\mathbf{w}}_t - \alpha_t \mathbf{g}_t) - \mathbf{w}^*\|^2 \\ &= \|(\overline{\mathbf{w}}_t - \alpha_t \overline{\mathbf{g}}_t - \mathbf{w}^*) - \alpha_t(\mathbf{g}_t - \overline{\mathbf{g}}_t)\|^2 \\ &= A_1 + A_2 + A_3\end{aligned}$$

where

$$\begin{aligned}A_1 &= \|\overline{\mathbf{w}}_t - \mathbf{w}^* - \alpha_t \overline{\mathbf{g}}_t\|^2 \\ A_2 &= 2\alpha_t \langle \overline{\mathbf{w}}_t - \mathbf{w}^* - \alpha_t \overline{\mathbf{g}}_t, \overline{\mathbf{g}}_t - \mathbf{g}_t \rangle \\ A_3 &= \alpha_t^2 \|\mathbf{g}_t - \overline{\mathbf{g}}_t\|^2\end{aligned}$$

By definition of $\mathbf{g}_t$ and $\overline{\mathbf{g}}_t$ (see Eq (B.2)), we have $\mathbb{E}A_2 = 0$. For A_3, we have the follow upper bound:

$$\alpha_t^2 \mathbb{E}\|\mathbf{g}_t - \overline{\mathbf{g}}_t\|^2 = \alpha_t^2 \mathbb{E}\|\mathbf{g}_t - \mathbb{E}\mathbf{g}_t\|^2 = \alpha_t^2 \sum_{k=1}^{N} p_k^2 \|\mathbf{g}_{t,k} - \mathbb{E}\mathbf{g}_{t,k}\|^2 \leq \alpha_t^2 \sum_{k=1}^{N} p_k^2 \sigma_k^2$$

again by Jensen's inequality and using the independence of $\mathbf{g}_{t,k}, \mathbf{g}_{t,k'}$ [110, Lemma 2].

Next we bound A_1:

$$\|\overline{\mathbf{w}}_t - \mathbf{w}^* - \alpha_t \overline{\mathbf{g}}_t\|^2 = \|\overline{\mathbf{w}}_t - \mathbf{w}^*\|^2 + 2\langle \overline{\mathbf{w}}_t - \mathbf{w}^*, -\alpha_t \overline{\mathbf{g}}_t \rangle + \|\alpha_t \overline{\mathbf{g}}_t\|^2$$

Same as the SGD case,

$$
\begin{aligned}
&- 2\alpha_t\langle\overline{\mathbf{w}}_t - \mathbf{w}^*, \overline{\mathbf{g}}_t\rangle + \|\alpha_t\overline{\mathbf{g}}_t\|^2 \\
\leq&\alpha_t L\sum_{k=1}^{N} p_k\|\overline{\mathbf{w}}_t - \mathbf{w}_t^k\|^2 + \alpha_t^2 L^2\sum_k p_k\|\overline{\mathbf{w}}_t - \mathbf{w}_t^k\|^2 + \alpha_t^3 L\mathbb{E}\|\mathbf{g}_t\|^2 - \alpha_t\mu\|\overline{\mathbf{w}}_t - \mathbf{w}^*\|^2
\end{aligned}
$$

so that

$$
\begin{aligned}
&\|\overline{\mathbf{w}}_t - \mathbf{w}^* - \alpha_t\overline{\mathbf{g}}_t\|^2 \\
\leq&(1-\alpha_t\mu)\|\overline{\mathbf{w}}_t - \mathbf{w}^*\|^2 + \alpha_t L\sum_{k=1}^{N} p_k\|\overline{\mathbf{w}}_t - \mathbf{w}_t^k\|^2 + \alpha_t^2 L^2\sum_k p_k\|\overline{\mathbf{w}}_t - \mathbf{w}_t^k\|^2 + \alpha_t^3 L\mathbb{E}\|\mathbf{g}_t\|^2
\end{aligned}
$$

Different from the SGD case, we have

$$
\begin{aligned}
&\|\overline{\mathbf{w}}_t - \mathbf{w}^*\|^2 \\
=&\|\overline{\mathbf{v}}_t + \beta_{t-1}(\overline{\mathbf{v}}_t - \overline{\mathbf{v}}_{t-1}) - \mathbf{w}^*\|^2 \\
=&\|(1+\beta_{t-1})(\overline{\mathbf{v}}_t - \mathbf{w}^*) - \beta_{t-1}(\overline{\mathbf{v}}_{t-1} - \mathbf{w}^*)\|^2 \\
=&(1+\beta_{t-1})^2\|\overline{\mathbf{v}}_t - \mathbf{w}^*\|^2 - 2\beta_{t-1}(1+\beta_{t-1})\langle\overline{\mathbf{v}}_t - \mathbf{w}^*, \overline{\mathbf{v}}_{t-1} - \mathbf{w}^*\rangle + \beta_{t-1}^2\|(\overline{\mathbf{v}}_{t-1} - \mathbf{w}^*)\|^2 \\
\leq&(1+\beta_{t-1})^2\|\overline{\mathbf{v}}_t - \mathbf{w}^*\|^2 + 2\beta_{t-1}(1+\beta_{t-1})\|\overline{\mathbf{v}}_t - \mathbf{w}^*\|\cdot\|\overline{\mathbf{v}}_{t-1} - \mathbf{w}^*\| \\
&\quad + \beta_{t-1}^2\|(\overline{\mathbf{v}}_{t-1} - \mathbf{w}^*)\|^2
\end{aligned}
$$

which gives

$$
\|\overline{\mathbf{v}}_{t+1} - \mathbf{w}^*\|^2
$$

$$
\begin{aligned}
&\leq (1-\alpha_t\mu)(1+\beta_{t-1})^2\|\overline{\mathbf{v}}_t - \mathbf{w}^*\|^2 \\
&+2(1-\alpha_t\mu)\beta_{t-1}(1+\beta_{t-1})\|\overline{\mathbf{v}}_t - \mathbf{w}^*\| \cdot \|\overline{\mathbf{v}}_{t-1} - \mathbf{w}^*\| \\
&+\alpha_t^2 \sum_{k=1}^N p_k^2\sigma_k^2 + \beta_{t-1}^2(1-\alpha_t\mu)\|(\overline{\mathbf{v}}_{t-1} - \mathbf{w}^*)\|^2 \\
&+\alpha_t L \sum_{k=1}^N p_k\|\overline{\mathbf{w}}_t - \mathbf{w}_t^k\|^2 + \alpha_t^2 L^2 \sum_k p_k\|\overline{\mathbf{w}}_t - \mathbf{w}_t^k\|^2 + \alpha_t^3 L G^2
\end{aligned}
$$

and we will using this recursive relation to obtain the desired bound.

First we bound $\mathbb{E}\sum_{k=1}^N p_k\|\overline{\mathbf{w}}_t - \mathbf{w}_t^k\|^2$. Since communication is done every E steps, for any $t \geq 0$, we can find a $t_0 \leq t$ such that $t - t_0 \leq E-1$ and $w_{t_0}^k = \overline{\mathbf{w}}_{t_0}$for all k. Moreover, using α_t is non-increasing, $\alpha_{t_0} \leq 2\alpha_t$, and $\beta_t \leq \alpha_t$ for any $t - t_0 \leq E-1$, we have

$$
\begin{aligned}
&\mathbb{E}\sum_{k=1}^N p_k\|\overline{\mathbf{w}}_t - \mathbf{w}_t^k\|^2 \\
&= \mathbb{E}\sum_{k=1}^N p_k\|\mathbf{w}_t^k - \overline{\mathbf{w}}_{t_0} - (\overline{\mathbf{w}}_t - \overline{\mathbf{w}}_{t_0})\|^2 \\
&\leq \mathbb{E}\sum_{k=1}^N p_k\|\mathbf{w}_t^k - \overline{\mathbf{w}}_{t_0}\|^2 \\
&= \mathbb{E}\sum_{k=1}^N p_k\|\mathbf{w}_t^k - \mathbf{w}_{t_0}^k\|^2 \\
&= \mathbb{E}\sum_{k=1}^N p_k\|\sum_{i=t_0}^{t-1}\beta_i(\mathbf{v}_{i+1}^k - \mathbf{v}_i^k) - \sum_{i=t_0}^{t-1}\alpha_i\mathbf{g}_{i,k}\|^2 \\
&\leq 2\sum_{k=1}^N p_k\mathbb{E}\sum_{i=t_0}^{t-1}(E-1)\alpha_i^2\|\mathbf{g}_{i,k}\|^2 + 2\sum_{k=1}^N p_k\mathbb{E}\sum_{i=t_0}^{t-1}(E-1)\beta_i^2\|(\mathbf{v}_{i+1}^k - \mathbf{v}_i^k)\|^2 \\
&\leq 2\sum_{k=1}^N p_k\mathbb{E}\sum_{i=t_0}^{t-1}(E-1)\alpha_i^2(\|\mathbf{g}_{i,k}\|^2 + \|(\mathbf{v}_{i+1}^k - \mathbf{v}_i^k)\|^2) \\
&\leq 4\sum_{k=1}^N p_k\mathbb{E}\sum_{i=t_0}^{t-1}(E-1)\alpha_i^2 G^2
\end{aligned}
$$

$$\leq 4(E-1)^2\alpha_{t_0}^2 G^2 \leq 16(E-1)^2\alpha_t^2 G^2$$

where we have used $\mathbb{E}\|\mathbf{v}_t^k - \mathbf{v}_{t-1}^k\|^2 \leq G^2$. To see this identity for appropriate α_t, β_t, note the recursion

$$\begin{aligned}\mathbf{v}_{t+1}^k - \mathbf{v}_t^k &= \mathbf{w}_t^k - \mathbf{w}_{t-1}^k - (\alpha_t \mathbf{g}_{t,k} - \alpha_{t-1}\mathbf{g}_{t-1,k}) \\ \mathbf{w}_{t+1}^k - \mathbf{w}_t^k &= -\alpha_t \mathbf{g}_{t,k} + \beta_t(\mathbf{v}_{t+1}^k - \mathbf{v}_t^k)\end{aligned}$$

so that

$$\begin{aligned}\mathbf{v}_{t+1}^k - \mathbf{v}_t^k &= -\alpha_{t-1}\mathbf{g}_{t-1,k} + \beta_{t-1}(\mathbf{v}_t^k - \mathbf{v}_{t-1}^k) - (\alpha_t \mathbf{g}_{t,k} - \alpha_{t-1}\mathbf{g}_{t-1,k}) \\ &= \beta_{t-1}(\mathbf{v}_t^k - \mathbf{v}_{t-1}^k) - \alpha_t \mathbf{g}_{t,k}\end{aligned}$$

Since the identity $\mathbf{v}_{t+1}^k - \mathbf{v}_t^k = \beta_{t-1}(\mathbf{v}_t^k - \mathbf{v}_{t-1}^k) - \alpha_t \mathbf{g}_{t,k}$ implies

$$\mathbb{E}\|\mathbf{v}_{t+1}^k - \mathbf{v}_t^k\|^2 \leq 2\beta_{t-1}^2 \mathbb{E}\|\mathbf{v}_t^k - \mathbf{v}_{t-1}^k\|^2 + 2\alpha_t^2 G^2$$

as long as α_t, β_{t-1} satisfy $2\beta_{t-1}^2 + 2\alpha_t^2 \leq 1/2$, we can guarantee that $\mathbb{E}\|\mathbf{v}_t^k - \mathbf{v}_{t-1}^k\|^2 \leq G^2$ for all k by induction. This together with Jensen's inequality also gives $\mathbb{E}\|\overline{\mathbf{v}}_t - \overline{\mathbf{v}}_{t-1}\|^2 \leq G^2$ for all t.

Using the bound on $\mathbb{E}\sum_{k=1}^N p_k \|\overline{\mathbf{w}}_t - \mathbf{w}_t^k\|^2$, we can conclude that, with $\nu_{\max} := N \cdot \max_k p_k$,

$$\mathbb{E}\|\overline{\mathbf{v}}_{t+1} - \mathbf{w}^*\|^2$$

$$
\begin{aligned}
&\leq \mathbb{E}(1-\mu\alpha_t)(1+\beta_{t-1})^2\|\overline{\mathbf{v}}_t - \mathbf{w}^*\|^2 + 16E^2L\alpha_t^3G^2 + 16E^2L^2\alpha_t^4G^2 + \alpha_t^3LG^2 \\
&+ (1-\alpha_t\mu)\beta_{t-1}^2\|(\overline{\mathbf{v}}_{t-1} - \mathbf{w}^*)\|^2 + \alpha_t^2\sum_{k=1}^{N} p_k^2\sigma_k^2 \\
&+ 2\beta_{t-1}(1+\beta_{t-1})(1-\alpha_t\mu)\|\overline{\mathbf{v}}_t - \mathbf{w}^*\| \cdot \|\overline{\mathbf{v}}_{t-1} - \mathbf{w}^*\| \\
&\leq \mathbb{E}(1-\mu\alpha_t)(1+\beta_{t-1})^2\|\overline{\mathbf{v}}_t - \mathbf{w}^*\|^2 + 20E^2L\alpha_t^3G^2 + (1-\alpha_t\mu)\beta_{t-1}^2\|(\overline{\mathbf{v}}_{t-1} - \mathbf{w}^*)\|^2 \\
&+ \alpha_t^2\frac{1}{N}\nu_{\max}\sigma^2 + 2\beta_{t-1}(1+\beta_{t-1})(1-\alpha_t\mu)\|\overline{\mathbf{v}}_t - \mathbf{w}^*\| \cdot \|\overline{\mathbf{v}}_{t-1} - \mathbf{w}^*\|
\end{aligned}
$$

where $\sigma^2 = \sum_k p_k\sigma_k^2$, and α_t satisfies $L\alpha_t \leq \frac{1}{5}$. We show next that $\mathbb{E}\|\overline{\mathbf{v}}_t - \mathbf{w}^*\|^2 = O(\frac{1}{tN} + \frac{E^2}{t^2})$ by induction.

Assume that we have shown

$$
\mathbb{E}\|\overline{\mathbf{y}}_t - \mathbf{w}^*\|^2 \leq b(C\alpha_t^2 + D\alpha_t)
$$

for all iterations until t, where $C = 20E^2LG^2$, $D = \frac{1}{N}\nu_{max}^2\sigma^2$, and b is to be chosen later. For step sizes we choose $\alpha_t = \frac{6}{\mu}\frac{1}{t+\gamma}$ and $\beta_{t-1} = \frac{3}{14(t+\gamma)(1-\frac{6}{t+\gamma})\max\{\mu,1\}}$ where $\gamma = \max\{32\kappa, E\}$, so that $\beta_{t-1} \leq \alpha_t$ and

$$
\begin{aligned}
&(1-\mu\alpha_t)(1+14\beta_{t-1}) \\
\leq&(1-\frac{6}{t+\gamma})(1+\frac{3}{(t+\gamma)(1-\frac{6}{t+\gamma})}) \\
=&1-\frac{6}{t+\gamma}+\frac{3}{t+\gamma} = 1-\frac{3}{t+\gamma} = 1-\frac{\mu\alpha_t}{2}
\end{aligned}
$$

Recall that we also require $\alpha_{t_0} \leq 2\alpha_t$ for any $t - t_0 \leq E-1$, $L\alpha_t \leq \frac{1}{5}$, and $2\beta_{t-1}^2 + 2\alpha_t^2 \leq 1/2$, which we can also check to hold by definition of α_t and β_t.

Moreover, $\mathbb{E}\|\overline{\mathbf{y}}_t - \mathbf{w}^*\|^2 \leq b(C\alpha_t^2 + D\alpha_t)$ with the chosen step sizes also implies $\|\overline{\mathbf{v}}_{t-1} -$

$\mathbf{w}^*\| \leq 2\|\overline{\mathbf{v}}_t - \mathbf{w}^*\|$. Therefore the bound for $\mathbb{E}\|\overline{\mathbf{v}}_{t+1} - \mathbf{w}^*\|^2$ can be further simplified with

$$2\beta_{t-1}(1+\beta_{t-1})(1-\alpha_t\mu)\|\overline{\mathbf{v}}_t - \mathbf{w}^*\| \cdot \|\overline{\mathbf{v}}_{t-1} - \mathbf{w}^*\| \leq 4\beta_{t-1}(1+\beta_{t-1})(1-\alpha_t\mu)\|\overline{\mathbf{v}}_t - \mathbf{w}^*\|^2$$

and

$$(1-\alpha_t\mu)\beta_{t-1}^2\|(\overline{\mathbf{v}}_{t-1} - \mathbf{w}^*)\|^2 \leq 4(1-\alpha_t\mu)\beta_{t-1}^2\|(\overline{\mathbf{v}}_t - \mathbf{w}^*)\|^2$$

so that

$$\begin{aligned}
\mathbb{E}\|\overline{\mathbf{v}}_{t+1} - \mathbf{w}^*\|^2 &\leq (1-\mu\alpha_t)((1+\beta_{t-1})^2 + 4\beta_{t-1}(1+\beta_{t-1}) + 4\beta_{t-1}^2)\mathbb{E}\|(\overline{\mathbf{v}}_t - \mathbf{w}^*)\|^2 \\
&\quad + 20E^2L\alpha_t^3G^2 + \alpha_t^2\frac{1}{N}\nu_{\max}\sigma^2 \\
&\leq \mathbb{E}(1-\mu\alpha_t)(1+14\beta_{t-1})\|(\overline{\mathbf{v}}_t - \mathbf{w}^*)\|^2 + 20E^2L\alpha_t^3G^2 + \alpha_t^2\frac{1}{N}\nu_{\max}\sigma^2 \\
&\leq b(1-\frac{\mu\alpha_t}{2})(C\alpha_t^2 + D\alpha_t) + C\alpha_t^3 + D\alpha_t^2 \\
&= (b(1-\frac{\mu\alpha_t}{2}) + \alpha_t)\alpha_t^2C + (b(1-\frac{\mu\alpha_t}{2}) + \alpha_t)\alpha_tD
\end{aligned}$$

and so it remains to choose b such that

$$\begin{aligned}
(b(1-\frac{\mu\alpha_t}{2}) + \alpha_t)\alpha_t &\leq b\alpha_{t+1} \\
(b(1-\frac{\mu\alpha_t}{2}) + \alpha_t)\alpha_t^2 &\leq b\alpha_{t+1}^2
\end{aligned}$$

from which we can conclude $\mathbb{E}\|\overline{\mathbf{v}}_{t+1} - \mathbf{w}^*\|^2 \leq \alpha_{t+1}^2C + \alpha_{t+1}D$.

With $b = \frac{6}{\mu}$, we have

$$\begin{aligned}(b(1 - \frac{\mu\alpha_t}{2}) + \alpha_t)\alpha_t &= (b(1 - (\frac{3}{t+\gamma}) + \frac{6}{\mu(t+\gamma)})\frac{6}{\mu(t+\gamma)} \\ &= (b\frac{t+\gamma-3}{t+\gamma} + \frac{6}{\mu(t+\gamma)})\frac{6}{\mu(t+\gamma)} \\ &\leq b(\frac{t+\gamma-1}{t+\gamma})\frac{6}{\mu(t+\gamma)} \\ &\leq b\frac{6}{\mu(t+\gamma+1)} = b\alpha_{t+1}\end{aligned}$$

where we have used $\frac{t+\gamma-1}{(t+\gamma)^2} \leq \frac{1}{t+\gamma+1}$.

Similarly

$$\begin{aligned}(b(1 - \frac{\mu\alpha_t}{2}) + \alpha_t)\alpha_t^2 &= (b(1 - (\frac{3}{t+\gamma}) + \frac{6}{\mu(t+\gamma)})(\frac{6}{\mu(t+\gamma)})^2 \\ &= (b\frac{t+\gamma-3}{t+\gamma} + \frac{6}{\mu(t+\gamma)})(\frac{6}{\mu(t+\gamma)})^2 \\ &= b(\frac{t+\gamma-2}{t+\gamma})(\frac{6}{\mu(t+\gamma)})^2 \\ &\leq b\frac{36}{\mu^2(t+\gamma+1)^2} = b\alpha_{t+1}^2\end{aligned}$$

where we have used $\frac{t+\gamma-2}{(t+\gamma)^3} \leq \frac{1}{(t+\gamma+1)^2}$.

Finally, to ensure $\|\mathbf{v}_0 - \mathbf{w}^*\|^2 \leq b(C\alpha_0^2 + D\alpha_0)$, we can rescale b by $c\|\mathbf{v}_0 - \mathbf{w}^*\|^2$ for some c. It follows that $\mathbb{E}\|\overline{\mathbf{v}}_t - \mathbf{w}^*\|^2 \leq b(C\alpha_t^2 + D\alpha_t)$ for all t. Thus

$$\begin{aligned}\mathbb{E}(F(\overline{\mathbf{w}}_T)) - F^* &= \mathbb{E}(F(\overline{\mathbf{w}}_T) - F(\mathbf{w}^*)) \\ &\leq \frac{L}{2}\mathbb{E}\|\overline{\mathbf{w}}_T - \mathbf{w}^*\|^2 \leq \frac{L}{2}c\|\mathbf{w}_0 - \mathbf{w}^*\|^2\frac{6}{\mu}(D\alpha_T + C\alpha_T^2) \\ &= 3c\|\mathbf{w}_0 - \mathbf{w}^*\|^2\kappa(D\alpha_T + C\alpha_T^2)\end{aligned}$$

$$\leq 3c\|\mathbf{w}_0 - \mathbf{w}^*\|^2\kappa \left[\frac{6}{\mu(T+\gamma)} \cdot \frac{1}{N}\nu_{\max}\sigma^2 + 20E^2LG^2 \cdot (\frac{6}{\mu(T+\gamma)})^2\right]$$
$$= O(\frac{\kappa}{\mu}\frac{1}{N}\nu_{\max}\sigma^2 \cdot \frac{1}{T} + \frac{\kappa^2}{\mu}E^2G^2 \cdot \frac{1}{T^2})$$

With partial participation, the same argument in the SGD case yields

$$\mathbb{E}F(\overline{\mathbf{w}}_T) - F^* = O(\frac{\kappa\nu_{\max}\sigma^2/\mu}{NT} + \frac{\kappa E^2G^2/\mu}{KT} + \frac{\kappa^2E^2G^2/\mu}{T^2})$$

□

Convex Smooth Objectives

Theorem 17. *Set learning rates $\alpha_t = \beta_t = \mathcal{O}(\sqrt{\frac{N}{T}})$. Then under Assumptions 7,9,10 Nesterov accelerated FedAvg with full device participation has rate*

$$\min_{t\leq T} F(\overline{\mathbf{w}}_t) - F^* = \mathcal{O}\left(\frac{\nu_{\max}\sigma^2}{\sqrt{NT}} + \frac{NE^2LG^2}{T}\right),$$

and with partial device participation with K sampled devices at each communication round,

$$\min_{t\leq T} F(\overline{\mathbf{w}}_t) - F^* = \mathcal{O}\left(\frac{\nu_{\max}\sigma^2}{\sqrt{KT}} + \frac{E^2G^2}{\sqrt{KT}} + \frac{KE^2LG^2}{T}\right).$$

Proof. Define $\overline{\mathbf{p}}_t := \frac{\beta_t}{1-\beta_t}\left[\overline{\mathbf{w}}_t - \overline{\mathbf{w}}_{t-1} + \alpha_t\mathbf{g}_{t-1}\right] = \frac{\beta_t^2}{1-\beta_t}(\overline{\mathbf{v}}_t - \overline{\mathbf{v}}_{t-1})$ for $t \geq 1$ and 0 for $t = 0$. We can check that

$$\overline{\mathbf{w}}_{t+1} + \overline{\mathbf{p}}_{t+1} = \overline{\mathbf{w}}_t + \overline{\mathbf{p}}_t - \frac{\alpha_t}{1-\beta_t}\mathbf{g}_t$$

Now we define $\overline{\mathbf{z}}_t := \overline{\mathbf{w}}_t + \overline{\mathbf{p}}_t$ and $\eta_t = \frac{\alpha_t}{1-\beta_t}$ for all t, so that we have the recursive relation

$$\overline{\mathbf{z}}_{t+1} = \overline{\mathbf{z}}_t - \eta_t \mathbf{g}_t$$

Now

$$\begin{aligned}
\|\overline{\mathbf{z}}_{t+1} - \mathbf{w}^*\|^2 &= \|(\overline{\mathbf{z}}_t - \eta_t \mathbf{g}_t) - \mathbf{w}^*\|^2 \\
&= \|(\overline{\mathbf{z}}_t - \eta_t \overline{\mathbf{g}}_t - \mathbf{w}^*) - \eta_t(\mathbf{g}_t - \overline{\mathbf{g}}_t)\|^2 \\
&= A_1 + A_2 + A_3
\end{aligned}$$

where

$$\begin{aligned}
A_1 &= \|\overline{\mathbf{z}}_t - \mathbf{w}^* - \eta_t \overline{\mathbf{g}}_t\|^2 \\
A_2 &= 2\eta_t \langle \overline{\mathbf{z}}_t - \mathbf{w}^* - \eta_t \overline{\mathbf{g}}_t, \overline{\mathbf{g}}_t - \mathbf{g}_t \rangle \\
A_3 &= \eta_t^2 \|\mathbf{g}_t - \overline{\mathbf{g}}_t\|^2
\end{aligned}$$

where again $\mathbb{E}A_2 = 0$ and $\mathbb{E}A_3 \leq \eta_t^2 \sum_k p_k^2 \sigma_k^2$. For A_1 we have

$$\|\overline{\mathbf{z}}_t - \mathbf{w}^* - \eta_t \overline{\mathbf{g}}_t\|^2 = \|\overline{\mathbf{z}}_t - \mathbf{w}^*\|^2 + 2\langle \overline{\mathbf{z}}_t - \mathbf{w}^*, -\eta_t \overline{\mathbf{g}}_t \rangle + \|\eta_t \overline{\mathbf{g}}_t\|^2$$

Using the convexity and L-smoothness of F_k,

$$\begin{aligned}
&- 2\eta_t \langle \overline{\mathbf{z}}_t - \mathbf{w}^*, \overline{\mathbf{g}}_t \rangle \\
&= -2\eta_t \sum_{k=1}^{N} p_k \langle \overline{\mathbf{z}}_t - \mathbf{w}^*, \nabla F_k(\mathbf{w}_t^k) \rangle
\end{aligned}$$

$$
\begin{aligned}
&= -2\eta_t \sum_{k=1}^{N} p_k \langle \overline{\mathbf{z}}_t - \mathbf{w}_t^k, \nabla F_k(\mathbf{w}_t^k) \rangle - 2\eta_t \sum_{k=1}^{N} p_k \langle \mathbf{w}_t^k - \mathbf{w}^*, \nabla F_k(\mathbf{w}_t^k) \rangle \\
&= -2\eta_t \sum_{k=1}^{N} p_k \langle \overline{\mathbf{z}}_t - \overline{\mathbf{w}}_t, \nabla F_k(\mathbf{w}_t^k) \rangle - 2\eta_t \sum_{k=1}^{N} p_k \langle \overline{\mathbf{w}}_t - \mathbf{w}_t^k, \nabla F_k(\mathbf{w}_t^k) \rangle \\
&\quad - 2\eta_t \sum_{k=1}^{N} p_k \langle \mathbf{w}_t^k - \mathbf{w}^*, \nabla F_k(\mathbf{w}_t^k) \rangle \\
&\leq -2\eta_t \sum_{k=1}^{N} p_k \langle \overline{\mathbf{z}}_t - \overline{\mathbf{w}}_t, \nabla F_k(\mathbf{w}_t^k) \rangle - 2\eta_t \sum_{k=1}^{N} p_k \langle \overline{\mathbf{w}}_t - \mathbf{w}_t^k, \nabla F_k(\mathbf{w}_t^k) \rangle \\
&\quad + 2\eta_t \sum_{k=1}^{N} p_k (F_k(\mathbf{w}^*) - F_k(\mathbf{w}_t^k)) \\
&\leq 2\eta_t \sum_{k=1}^{N} p_k \left[F_k(\mathbf{w}_t^k) - F_k(\overline{\mathbf{w}}_t) + \frac{L}{2} \|\overline{\mathbf{w}}_t - \mathbf{w}_t^k\|^2 + F_k(\mathbf{w}^*) - F_k(\mathbf{w}_t^k) \right] \\
&- 2\eta_t \sum_{k=1}^{N} p_k \langle \overline{\mathbf{z}}_t - \overline{\mathbf{w}}_t, \nabla F_k(\mathbf{w}_t^k) \rangle \\
&= \eta_t L \sum_{k=1}^{N} p_k \|\overline{\mathbf{w}}_t - \mathbf{w}_t^k\|^2 + 2\eta_t \sum_{k=1}^{N} p_k \left[F_k(\mathbf{w}^*) - F_k(\overline{\mathbf{w}}_t) \right] - 2\eta_t \sum_{k=1}^{N} p_k \langle \overline{\mathbf{z}}_t - \overline{\mathbf{w}}_t, \nabla F_k(\mathbf{w}_t^k) \rangle
\end{aligned}
$$

which results in

$$
\begin{aligned}
\mathbb{E}\|\overline{\mathbf{w}}_{t+1} - \mathbf{w}^*\|^2 \leq \mathbb{E}\|\overline{\mathbf{w}}_t - \mathbf{w}^*\|^2 &+ \eta_t L \sum_{k=1}^{N} p_k \|\overline{\mathbf{w}}_t - \mathbf{w}_t^k\|^2 + 2\eta_t \sum_{k=1}^{N} p_k \left[F_k(\mathbf{w}^*) - F_k(\overline{\mathbf{w}}_t) \right] \\
&+ \eta_t^2 \|\overline{\mathbf{g}}_t\|^2 + \eta_t^2 \sum_{k=1}^{N} p_k^2 \sigma_k^2 - 2\eta_t \sum_{k=1}^{N} p_k \langle \overline{\mathbf{z}}_t - \overline{\mathbf{w}}_t, \nabla F_k(\mathbf{w}_t^k) \rangle
\end{aligned}
$$

As before, $\|\overline{\mathbf{g}}_t\|^2 \leq 2L^2 \sum_k p_k \|\mathbf{w}_t^k - \overline{\mathbf{w}}_t\|^2 + 4L(F(\overline{\mathbf{w}}_t) - F(\mathbf{w}^*))$, so that

$$
\begin{aligned}
&\eta_t^2 \|\overline{\mathbf{g}}_t\|^2 + \eta_t \sum_{k=1}^{N} p_k \left[F_k(\mathbf{w}^*) - F_k(\overline{\mathbf{w}}_t) \right] \\
\leq &2L^2 \eta_t^2 \sum_k p_k \|\mathbf{w}_t^k - \overline{\mathbf{w}}_t\|^2 + \eta_t (1 - 4\eta_t L)(F(\mathbf{w}^*) - F(\overline{\mathbf{w}}_t))
\end{aligned}
$$

$$\leq 2L^2\eta_t^2 \sum_k p_k \|\mathbf{w}_t^k - \overline{\mathbf{w}}_t\|^2$$

for $\eta_t \leq 1/4L$. Using $\sum_{k=1}^N p_k \|\overline{\mathbf{w}}_t - \mathbf{w}_t^k\|^2 \leq 16E^2\alpha_t^2 G^2$ and $\sum_{k=1}^N p_k^2 \sigma_k^2 \leq \nu_{\max}\frac{1}{N}\sigma^2$, it follows that

$$\begin{aligned}
&\mathbb{E}\|\overline{\mathbf{w}}_{t+1} - \mathbf{w}^*\|^2 + \eta_t(F(\overline{\mathbf{w}}_t) - F(\mathbf{w}^*)) \\
&\leq \mathbb{E}\|\overline{\mathbf{w}}_t - \mathbf{w}^*\|^2 + (\eta_t L + 2L^2\eta_t^2)\sum_{k=1}^N p_k \|\overline{\mathbf{w}}_t - \mathbf{w}_t^k\|^2 + \eta_t^2 \sum_{k=1}^N p_k^2 \sigma_k^2 \\
&- 2\eta_t \sum_{k=1}^N p_k \langle \overline{\mathbf{z}}_t - \overline{\mathbf{w}}_t, \nabla F_k(\mathbf{w}_t^k)\rangle \\
&\leq \mathbb{E}\|\overline{\mathbf{w}}_t - \mathbf{w}^*\|^2 + 32LE^2\alpha_t^2\eta_t G^2 + \eta_t^2 \nu_{\max}\frac{1}{N}\sigma^2 \\
&- 2\eta_t \sum_{k=1}^N p_k \langle \overline{\mathbf{z}}_t - \overline{\mathbf{w}}_t, \nabla F_k(\mathbf{w}_t^k)\rangle
\end{aligned}$$

if $\eta_t \leq \frac{1}{2L}$. It remains to bound $\mathbb{E}\sum_{k=1}^N p_k \langle \overline{\mathbf{z}}_t - \overline{\mathbf{w}}_t, \nabla F_k(\mathbf{w}_t^k)\rangle$. Recall that $\overline{\mathbf{z}}_t - \overline{\mathbf{w}}_t = \frac{\beta_t}{1-\beta_t}[\overline{\mathbf{w}}_t - \overline{\mathbf{w}}_{t-1} + \alpha_t \mathbf{g}_{t-1}] = \frac{\beta_t^2}{1-\beta_t}(\overline{\mathbf{v}}_t - \overline{\mathbf{v}}_{t-1})$ and $\mathbb{E}\|\overline{\mathbf{v}}_t - \overline{\mathbf{v}}_{t-1}\|^2 \leq G^2$, $\mathbb{E}\|\nabla F_k(\mathbf{w}_t^k)\|^2 \leq G^2$.

Cauchy-Schwarz gives

$$\begin{aligned}
\mathbb{E}\sum_{k=1}^N p_k \langle \overline{\mathbf{z}}_t - \overline{\mathbf{w}}_t, \nabla F_k(\mathbf{w}_t^k)\rangle &\leq \sum_{k=1}^N p_k \sqrt{\mathbb{E}\|\overline{\mathbf{z}}_t - \overline{\mathbf{w}}_t\|^2} \cdot \sqrt{\mathbb{E}\|\nabla F_k(\mathbf{w}_t^k)\|^2} \\
&\leq \frac{\beta_t^2}{1-\beta_t} G^2
\end{aligned}$$

Thus

$$\mathbb{E}\|\overline{\mathbf{w}}_{t+1} - \mathbf{w}^*\|^2 + \eta_t(F(\overline{\mathbf{w}}_t) - F(\mathbf{w}^*))$$

$$\leq \mathbb{E}\|\overline{\mathbf{w}}_t - \mathbf{w}^*\|^2 + 32LE^2\alpha_t^2\eta_t G^2 + \eta_t^2\nu_{\max}\frac{1}{N}\sigma^2 + 2\eta_t\frac{\beta_t^2}{1-\beta_t}G^2$$

Summing the inequalities from $t = 0$ to $t = T$, we obtain

$$\begin{aligned}
&\sum_{t=0}^{T}\eta_t(F(\overline{\mathbf{w}}_t) - F(\mathbf{w}^*)) \\
&\leq \|\mathbf{w}_0 - \mathbf{w}^*\|^2 + \sum_{t=0}^{T}\eta_t^2 \cdot \frac{1}{N}\nu_{\max}\sigma^2 + \sum_{t=0}^{T}\eta_t\alpha_t^2 \cdot 32LE^2G^2 + \sum_{t=0}^{T}2\eta_t\frac{\beta_t^2}{1-\beta_t}G^2
\end{aligned}$$

so that

$$\begin{aligned}
&\min_{t\leq T} F(\overline{\mathbf{w}}_t) - F(\mathbf{w}^*) \\
\leq &\frac{1}{\sum_{t=0}^{T}\eta_t}\left(\|\mathbf{w}_0 - \mathbf{w}^*\|^2 + \sum_{t=0}^{T}\eta_t^2 \cdot \frac{1}{N}\nu_{\max}\sigma^2 + \sum_{t=0}^{T}\eta_t\alpha_t^2 \cdot 32LE^2G^2 + \sum_{t=0}^{T}2\eta_t\frac{\beta_t^2}{1-\beta_t}G^2\right)
\end{aligned}$$

By setting the constant learning rates $\alpha_t \equiv \sqrt{\frac{N}{T}}$ and $\beta_t \equiv c\sqrt{\frac{N}{T}}$ so that $\eta_t = \frac{\alpha_t}{1-\beta_t} = \frac{\sqrt{\frac{N}{T}}}{1-c\sqrt{\frac{N}{T}}} \leq 2\sqrt{\frac{N}{T}}$, we have

$$\begin{aligned}
&\min_{t\leq T} F(\overline{\mathbf{w}}_t) - F(\mathbf{w}^*) \\
&\leq \frac{1}{2\sqrt{NT}} \cdot \|\mathbf{w}_0 - \mathbf{w}^*\|^2 + \frac{2}{\sqrt{NT}}T \cdot \frac{N}{T} \cdot \frac{1}{N}\nu_{\max}\sigma^2 \\
&\quad + \frac{1}{\sqrt{NT}}T(\sqrt{\frac{N}{T}})^3 32LE^2G^2 + \frac{2}{\sqrt{NT}}T(\sqrt{\frac{N}{T}})^3 G^2 \\
&= (\frac{1}{2}\|\mathbf{w}_0 - \mathbf{w}^*\|^2 + 2\nu_{\max}\sigma^2)\frac{1}{\sqrt{NT}} + \frac{N}{T}(32LE^2G^2 + 2G^2) \\
&= O(\frac{\nu_{\max}\sigma^2}{\sqrt{NT}} + \frac{NE^2LG^2}{T})
\end{aligned}$$

Similarly, for partial participation, we have

$$\begin{aligned}
&\min_{t\leq T} F(\overline{\mathbf{w}}_t) - F(\mathbf{w}^*) \\
\leq &\frac{1}{\sum_{t=0}^{T}\alpha_t}\left(\|\mathbf{w}_0-\mathbf{w}^*\|^2+\sum_{t=0}^{T}\alpha_t^2\cdot(\frac{1}{N}\nu_{\max}\sigma^2+C)+\sum_{t=0}^{T}\alpha_t^3\cdot 6E^2LG^2\right)
\end{aligned}$$

where $C=\frac{4}{K}E^2G^2$ or $\frac{N-K}{N-1}\frac{4}{K}E^2G^2$, so that with $\alpha_t\equiv\sqrt{\frac{K}{T}}$ and $\beta_t\equiv c\sqrt{\frac{K}{T}}$, we have

$$\min_{t\leq T} F(\overline{\mathbf{w}}_t) - F(\mathbf{w}^*) = \mathcal{O}(\frac{\nu_{\max}\sigma^2}{\sqrt{KT}}+\frac{E^2G^2}{\sqrt{KT}}+\frac{KE^2LG^2}{T})$$

□

Proof of Geometric Convergence Results for Overparameterized Problems

Geometric Convergence of FedAvg for general strongly convex and smooth objectives

Theorem 18. *For the overparameterized setting with general strongly convex and smooth objectives, FedAvg with local SGD updates and communication every E iterations with constant step size $\overline{\alpha}=\frac{1}{2E}\frac{N}{l\nu_{\max}+L(N-\nu_{\min})}$ gives the exponential convergence guarantee*

$$\mathbb{E}F(\overline{\mathbf{w}}_t)\leq\frac{L}{2}(1-\mu\overline{\alpha})^t\|\mathbf{w}_0-\mathbf{w}^*\|^2=O(\exp(-\frac{\mu}{2E}\frac{N}{l\nu_{\max}+L(N-\nu_{\min})}t)\cdot\|\mathbf{w}_0-\mathbf{w}^*\|^2)$$

Proof. To illustrate the main ideas of the proof, we first present the proof for $E=2$. Let

$t-1$ be a communication round, so that $\mathbf{w}_{t-1}^k = \overline{\mathbf{w}}_{t-1}$. We show that

$$\|\overline{\mathbf{w}}_{t+1} - \mathbf{w}^*\|^2 \leq (1-\alpha_t\mu)(1-\alpha_{t-1}\mu)\|\overline{\mathbf{w}}_{t-1} - \mathbf{w}^*\|^2$$

for appropriately chosen constant step sizes α_t, α_{t-1}. We have

$$\begin{aligned}\|\overline{\mathbf{w}}_{t+1} - \mathbf{w}^*\|^2 &= \|(\overline{\mathbf{w}}_t - \alpha_t\mathbf{g}_t) - \mathbf{w}^*\|^2 \\ &= \|\overline{\mathbf{w}}_t - \mathbf{w}^*\|^2 - 2\alpha_t\langle\overline{\mathbf{w}}_t - \mathbf{w}^*, \mathbf{g}_t\rangle + \alpha_t^2\|\mathbf{g}_t\|^2\end{aligned}$$

and the cross term can be bounded as usual using μ-convexity and L-smoothness of F_k:

$$\begin{aligned}&-2\alpha_t\mathbb{E}_t\langle\overline{\mathbf{w}}_t - \mathbf{w}^*, \mathbf{g}_t\rangle \\ &= -2\alpha_t\sum_{k=1}^N p_k\langle\overline{\mathbf{w}}_t - \mathbf{w}^*, \nabla F_k(\mathbf{w}_t^k)\rangle \\ &= -2\alpha_t\sum_{k=1}^N p_k\langle\overline{\mathbf{w}}_t - \mathbf{w}_t^k, \nabla F_k(\mathbf{w}_t^k)\rangle - 2\alpha_t\sum_{k=1}^N p_k\langle\mathbf{w}_t^k - \mathbf{w}^*, \nabla F_k(\mathbf{w}_t^k)\rangle \\ &\leq -2\alpha_t\sum_{k=1}^N p_k\langle\overline{\mathbf{w}}_t - \mathbf{w}_t^k, \nabla F_k(\mathbf{w}_t^k)\rangle + 2\alpha_t\sum_{k=1}^N p_k(F_k(\mathbf{w}^*) - F_k(\mathbf{w}_t^k)) \\ &\quad - \alpha_t\mu\sum_{k=1}^N p_k\|\mathbf{w}_t^k - \mathbf{w}^*\|^2 \\ &\leq 2\alpha_t\sum_{k=1}^N p_k\left[F_k(\mathbf{w}_t^k) - F_k(\overline{\mathbf{w}}_t) + \frac{L}{2}\|\overline{\mathbf{w}}_t - \mathbf{w}_t^k\|^2 + F_k(\mathbf{w}^*) - F_k(\mathbf{w}_t^k)\right] \\ &\quad - \alpha_t\mu\|\sum_{k=1}^N p_k(\mathbf{w}_t^k - \mathbf{w}^*)\|^2\end{aligned}$$

$$= \alpha_t L \sum_{k=1}^{N} p_k \|\overline{\mathbf{w}}_t - \mathbf{w}_t^k\|^2 + 2\alpha_t \sum_{k=1}^{N} p_k \left[F_k(\mathbf{w}^*) - F_k(\overline{\mathbf{w}}_t)\right] - \alpha_t \mu \|\overline{\mathbf{w}}_t - \mathbf{w}^*\|^2$$
$$= \alpha_t L \sum_{k=1}^{N} p_k \|\overline{\mathbf{w}}_t - \mathbf{w}_t^k\|^2 - 2\alpha_t \sum_{k=1}^{N} p_k F_k(\overline{\mathbf{w}}_t) - \alpha_t \mu \|\overline{\mathbf{w}}_t - \mathbf{w}^*\|^2$$

and so

$$\mathbb{E}\|\overline{\mathbf{w}}_{t+1} - \mathbf{w}^*\|^2 \leq \mathbb{E}(1 - \alpha_t \mu)\|\overline{\mathbf{w}}_t - \mathbf{w}^*\|^2 - 2\alpha_t F(\overline{\mathbf{w}}_t) + \alpha_t^2 \|\mathbf{g}_t\|^2 + \alpha_t L \sum_{k=1}^{N} p_k \|\overline{\mathbf{w}}_t - \mathbf{w}_t^k\|^2$$

Applying this recursive relation to $\|\overline{\mathbf{w}}_t - \mathbf{w}^*\|^2$ and using $\|\overline{\mathbf{w}}_{t-1} - \mathbf{w}_{t-1}^k\|^2 \equiv 0$, we further obtain

$$\begin{aligned}
&\mathbb{E}\|\overline{\mathbf{w}}_{t+1} - \mathbf{w}^*\|^2 \\
&\leq \mathbb{E}(1 - \alpha_t \mu) \left((1 - \alpha_{t-1}\mu)\|\overline{\mathbf{w}}_{t-1} - \mathbf{w}^*\|^2 - 2\alpha_{t-1} F(\overline{\mathbf{w}}_{t-1}) + \alpha_{t-1}^2 \|\mathbf{g}_{t-1}\|^2 \right) \\
&- 2\alpha_t F(\overline{\mathbf{w}}_t) + \alpha_t^2 \|\mathbf{g}_t\|^2 + \alpha_t L \sum_{k=1}^{N} p_k \|\overline{\mathbf{w}}_t - \mathbf{w}_t^k\|^2
\end{aligned}$$

Now instead of bounding $\sum_{k=1}^{N} p_k \|\overline{\mathbf{w}}_t - \mathbf{w}_t^k\|^2$ using the arguments in the general convex case, we follow [127] and use the fact that in the overparameterized setting, $\mathbf{w}^*$ is a minimizer of each $\ell(\mathbf{w}, x_k^j)$ and that each ℓ is l-smooth to obtain $\|\nabla F_k(\overline{\mathbf{w}}_{t-1}, \xi_{t-1}^k)\|^2 \leq 2l(F_k(\overline{\mathbf{w}}_{t-1}, \xi_{t-1}^k) - F_k(\mathbf{w}^*, \xi_{t-1}^k))$, where recall $F_k(\mathbf{w}, \xi_{t-1}^k) = \ell(\mathbf{w}, \xi_{t-1}^k)$, so that

$$\begin{aligned}
\sum_{k=1}^{N} p_k \|\overline{\mathbf{w}}_t - \mathbf{w}_t^k\|^2 &= \sum_{k=1}^{N} p_k \|\overline{\mathbf{w}}_{t-1} - \alpha_{t-1}\mathbf{g}_{t-1} - \mathbf{w}_{t-1}^k + \alpha_{t-1}\mathbf{g}_{t-1,k}\|^2 \\
&= \sum_{k=1}^{N} p_k \alpha_{t-1}^2 \|\mathbf{g}_{t-1} - \mathbf{g}_{t-1,k}\|^2 \\
&= \alpha_{t-1}^2 \sum_{k=1}^{N} p_k (\|\mathbf{g}_{t-1,k}\|^2 - \|\mathbf{g}_{t-1}\|^2)
\end{aligned}$$

$$= \alpha_{t-1}^2 \sum_{k=1}^{N} p_k \|\nabla F_k(\overline{\mathbf{w}}_{t-1}, \xi_{t-1}^k)\|^2 - \alpha_{t-1}^2 \|\mathbf{g}_{t-1}\|^2$$
$$\leq \alpha_{t-1}^2 \sum_{k=1}^{N} p_k 2l(F_k(\overline{\mathbf{w}}_{t-1}, \xi_{t-1}^k) - F_k(\mathbf{w}^*, \xi_{t-1}^k)) - \alpha_{t-1}^2 \|\mathbf{g}_{t-1}\|^2$$

again using $\overline{\mathbf{w}}_{t-1} = \mathbf{w}_{t-1}^k$. Taking expectation with respect to ξ_{t-1}^k's and using the fact that $F(\mathbf{w}^*) = 0$, we have

$$\mathbb{E}_{t-1} \sum_{k=1}^{N} p_k \|\overline{\mathbf{w}}_t - \mathbf{w}_t^k\|^2 \leq 2l\alpha_{t-1}^2 \sum_{k=1}^{N} p_k F_k(\overline{\mathbf{w}}_{t-1}) - \alpha_{t-1}^2 \|\mathbf{g}_{t-1}\|^2$$
$$= 2l\alpha_{t-1}^2 F(\overline{\mathbf{w}}_{t-1}) - \alpha_{t-1}^2 \|\mathbf{g}_{t-1}\|^2$$

Note also that

$$\|\mathbf{g}_{t-1}\|^2 = \| \sum_{k=1}^{N} p_k \nabla F_k(\overline{\mathbf{w}}_{t-1}, \xi_{t-1}^k)\|^2$$

while

$$\|\mathbf{g}_t\|^2 = \| \sum_{k=1}^{N} p_k \nabla F_k(\mathbf{w}_t^k, \xi_t^k)\|^2$$
$$\leq 2\| \sum_{k=1}^{N} p_k \nabla F_k(\overline{\mathbf{w}}_t, \xi_t^k)\|^2 + 2\| \sum_{k=1}^{N} p_k (\nabla F_k(\overline{\mathbf{w}}_t, \xi_t^k) - \nabla F_k(\mathbf{w}_t^k, \xi_t^k))\|^2$$
$$\leq 2\| \sum_{k=1}^{N} p_k \nabla F_k(\overline{\mathbf{w}}_t, \xi_t^k)\|^2 + 2 \sum_{k=1}^{N} p_k l^2 \|\overline{\mathbf{w}}_t - \mathbf{w}_t^k\|^2$$

Substituting these into the bound for $\|\overline{\mathbf{w}}_{t+1} - \mathbf{w}^*\|^2$, we have

$$\mathbb{E}\|\overline{\mathbf{w}}_{t+1} - \mathbf{w}^*\|^2$$

$$
\begin{aligned}
&\leq \mathbb{E}(1-\alpha_t\mu)((1-\alpha_{t-1}\mu)\|\overline{\mathbf{w}}_{t-1}-\mathbf{w}^*\|^2 - 2\alpha_{t-1}F(\overline{\mathbf{w}}_{t-1}) + \alpha_{t-1}^2\|\mathbf{g}_{t-1}\|^2) \\
&\quad - 2\alpha_t F(\overline{\mathbf{w}}_t) + 2\alpha_t^2\|\sum_{k=1}^{N} p_k \nabla F_k(\overline{\mathbf{w}}_t, \xi_t^k)\|^2 \\
&\quad + \left(2l^2\alpha_{t-1}^2\alpha_t^2 + \alpha_t\alpha_{t-1}^2 L\right)\left(2lF(\overline{\mathbf{w}}_{t-1}) - \|\mathbf{g}_{t-1}\|^2\right)
\end{aligned}
$$

$$
\begin{aligned}
&\mathbb{E}\|\overline{\mathbf{w}}_{t+1}-\mathbf{w}^*\|^2 \\
&\leq \mathbb{E}(1-\alpha_t\mu)(1-\alpha_{t-1}\mu)\|\overline{\mathbf{w}}_{t-1}-\mathbf{w}^*\|^2 - 2\alpha_t(F(\overline{\mathbf{w}}_t) - \alpha_t\|\sum_{k=1}^{N} p_k \nabla F_k(\overline{\mathbf{w}}_t, \xi_t^k)\|^2) \\
&-2\alpha_{t-1}(1-\alpha_t\mu)\left((1-\frac{l\alpha_{t-1}(2l^2\alpha_t^2+\alpha_t L)}{1-\alpha_t\mu})F(\overline{\mathbf{w}}_{t-1}) - \frac{\alpha_{t-1}}{2}\|\sum_{k=1}^{N} p_k \nabla F_k(\overline{\mathbf{w}}_{t-1}, \xi_{t-1}^k)\|^2\right)
\end{aligned}
$$

from which we can conclude that

$$
\mathbb{E}\|\overline{\mathbf{w}}_{t+1}-\mathbf{w}^*\|^2 \leq (1-\alpha_t\mu)(1-\alpha_{t-1}\mu)\mathbb{E}\|\overline{\mathbf{w}}_{t-1}-\mathbf{w}^*\|^2
$$

if we can choose α_t, α_{t-1} to guarantee

$$
\begin{aligned}
\mathbb{E}(F(\overline{\mathbf{w}}_t) - \alpha_t\|\sum_{k=1}^{N} p_k \nabla F_k(\overline{\mathbf{w}}_t, \xi_t^k)\|^2) \geq 0 \\
\mathbb{E}\left((1-\frac{l\alpha_{t-1}(2l^2\alpha_t^2+\alpha_t L)}{1-\alpha_t\mu})F(\overline{\mathbf{w}}_{t-1}) - \frac{\alpha_{t-1}}{2}\|\sum_{k=1}^{N} p_k \nabla F_k(\overline{\mathbf{w}}_{t-1}, \xi_{t-1}^k)\|^2\right) \geq 0
\end{aligned}
$$

Note that

$$
\begin{aligned}
&\mathbb{E}_t\|\sum_{k=1}^N p_k\nabla F_k(\overline{\mathbf{w}}_t,\xi_t^k)\|^2\\
=&\mathbb{E}_t\langle\sum_{k=1}^N p_k\nabla F_k(\overline{\mathbf{w}}_t,\xi_t^k),\sum_{k=1}^N p_k\nabla F_k(\overline{\mathbf{w}}_t,\xi_t^k)\rangle\\
=&\sum_{k=1}^N p_k^2\mathbb{E}_t\|\nabla F_k(\overline{\mathbf{w}}_t,\xi_t^k)\|^2+\sum_{k=1}\sum_{j\neq k}p_jp_k\mathbb{E}_t\langle\nabla F_k(\overline{\mathbf{w}}_t,\xi_t^k),\nabla F_j(\overline{\mathbf{w}}_t,\xi_t^j)\rangle\\
=&\sum_{k=1}^N p_k^2\mathbb{E}_t\|\nabla F_k(\overline{\mathbf{w}}_t,\xi_t^k)\|^2+\sum_{k=1}\sum_{j\neq k}p_jp_k\langle\nabla F_k(\overline{\mathbf{w}}_t),\nabla F_j(\overline{\mathbf{w}}_t)\rangle\\
=&\sum_{k=1}^N p_k^2\mathbb{E}_t\|\nabla F_k(\overline{\mathbf{w}}_t,\xi_t^k)\|^2+\sum_{k=1}^N\sum_{j=1}^N p_jp_k\langle\nabla F_k(\overline{\mathbf{w}}_t),\nabla F_j(\overline{\mathbf{w}}_t)\rangle\\
&-\sum_{k=1}^N p_k^2\|\nabla F_k(\overline{\mathbf{w}}_t)\|^2
\end{aligned}
$$

$$
\begin{aligned}
&\mathbb{E}_t\|\sum_{k=1}^N p_k\nabla F_k(\overline{\mathbf{w}}_t,\xi_t^k)\|^2\\
\leq&\sum_{k=1}^N p_k^2\mathbb{E}_t\|\nabla F_k(\overline{\mathbf{w}}_t,\xi_t^k)\|^2+\|\sum_k p_k\nabla F_k(\overline{\mathbf{w}}_t)\|^2-\frac{1}{N}\nu_{\min}\|\sum_k p_k\nabla F_k(\overline{\mathbf{w}}_t)\|^2\\
=&\sum_{k=1}^N p_k^2\mathbb{E}_t\|\nabla F_k(\overline{\mathbf{w}}_t,\xi_t^k)\|^2+(1-\frac{1}{N}\nu_{\min})\|\nabla F(\overline{\mathbf{w}}_t)\|^2
\end{aligned}
$$

and so following [127] if we let $\alpha_t=\min\{\frac{qN}{2l\nu_{\max}},\frac{1-q}{2L(1-\frac{1}{N}\nu_{\min})}\}$ for a $q\in[0,1]$ to be optimized

later, we have

$$
\begin{aligned}
&\mathbb{E}_t(F(\overline{\mathbf{w}}_t) - \alpha_t \| \sum_{k=1}^{N} p_k \nabla F_k(\overline{\mathbf{w}}_t, \xi_t^k)\|^2) \\
\geq &\mathbb{E}_t \sum_{k=1}^{N} p_k F_k(\overline{\mathbf{w}}_t) - \alpha_t \left[\sum_{k=1}^{N} p_k^2 \mathbb{E}_t \|\nabla F_k(\overline{\mathbf{w}}_t, \xi_t^k)\|^2 + (1 - \frac{1}{N}\nu_{\min})\|\nabla F(\overline{\mathbf{w}}_t)\|^2 \right] \\
\geq &\mathbb{E}_t \sum_{k=1}^{N} p_k (q F_k(\overline{\mathbf{w}}_t, \xi_t^k) - \alpha_t \frac{1}{N}\nu_{\max}\|\nabla F_k(\overline{\mathbf{w}}_t, \xi_t^k)\|^2) \\
&+ ((1-q)F(\overline{\mathbf{w}}_t) - \alpha_t(1 - \frac{1}{N}\nu_{\min})\|\nabla F(\overline{\mathbf{w}}_t)\|^2) \\
\geq &q\mathbb{E}_t \sum_{k=1}^{N} p_k (F_k(\overline{\mathbf{w}}_t, \xi_t^k) - \frac{1}{2l}\|\nabla F_k(\overline{\mathbf{w}}_t, \xi_t^k)\|^2) + (1-q)(F(\overline{\mathbf{w}}_t) - \frac{1}{2L}\|\nabla F(\overline{\mathbf{w}}_t)\|^2) \\
\geq &0
\end{aligned}
$$

again using $\mathbf{w}^*$ optimizes $F_k(\mathbf{w}, \xi_t^k)$ with $F_k(\mathbf{w}^*, \xi_t^k) = 0$.

Maximizing $\alpha_t = \min\{\frac{qN}{2l\nu_{\max}}, \frac{1-q}{2L(1-\frac{1}{N}\nu_{\min})}\}$ over $q \in [0,1]$, we see that $q = \frac{l\nu_{\max}}{l\nu_{\max}+L(N-\nu_{\min})}$ results in the fastest convergence, and this translates to $\alpha_t = \frac{1}{2}\frac{N}{l\nu_{\max}+L(N-\nu_{\min})}$. Next we claim that $\alpha_{t-1} = c\frac{1}{2}\frac{N}{l\nu_{\max}+L(N-\nu_{\min})}$ also guarantees

$$
\mathbb{E}(1 - \frac{l\alpha_{t-1}(2l^2\alpha_t^2 + \alpha_t L)}{1-\alpha_t\mu})F(\overline{\mathbf{w}}_{t-1}) - \frac{\alpha_{t-1}}{2}\| \sum_{k=1}^{N} p_k \nabla F_k(\overline{\mathbf{w}}_{t-1}, \xi_{t-1}^k)\|^2 \geq 0
$$

Note that by scaling α_{t-1} by a constant $c \leq 1$ if necessary, we can guarantee $\frac{l\alpha_{t-1}(2l^2\alpha_t^2+\alpha_t L)}{1-\alpha_t\mu} \leq \frac{1}{2}$, and so the condition is equivalent to

$$
F(\overline{\mathbf{w}}_{t-1}) - \alpha_{t-1}\| \sum_{k=1}^{N} p_k \nabla F_k(\overline{\mathbf{w}}_{t-1}, \xi_{t-1}^k)\|^2 \geq 0
$$

which was shown to hold with $\alpha_{t-1} \leq \frac{1}{2}\frac{N}{l\nu_{\max}+L(N-\nu_{\min})}$.

For the proof of general $E \geq 2$, we use the following two identities:

$$\|\mathbf{g}_t\|^2 \leq 2\|\sum_{k=1}^{N} p_k \nabla F_k(\overline{\mathbf{w}}_t, \xi_t^k)\|^2 + 2\sum_{k=1}^{N} p_k l^2 \|\overline{\mathbf{w}}_t - \mathbf{w}_t^k\|^2$$

$$\begin{aligned}
&\mathbb{E}\sum_{k=1}^{N} p_k \|\overline{\mathbf{w}}_t - \mathbf{w}_t^k\|^2 \\
\leq &\mathbb{E}2(1 + 2l^2\alpha_{t-1}^2)\sum_{k=1}^{N} p_k \|\overline{\mathbf{w}}_{t-1} - \mathbf{w}_{t-1}^k\|^2 + 8\alpha_{t-1}^2 l F(\overline{\mathbf{w}}_{t-1}) - 2\alpha_{t-1}^2 \|\mathbf{g}_{t-1}\|^2
\end{aligned}$$

where the first inequality has been established before. To establish the second inequality, note that

$$\begin{aligned}
\sum_{k=1}^{N} p_k \|\overline{\mathbf{w}}_t - \mathbf{w}_t^k\|^2 &= \sum_{k=1}^{N} p_k \|\overline{\mathbf{w}}_{t-1} - \alpha_{t-1}\mathbf{g}_{t-1} - \mathbf{w}_{t-1}^k + \alpha_{t-1}\mathbf{g}_{t-1,k}\|^2 \\
&\leq 2\sum_{k=1}^{N} p_k \left(\|\overline{\mathbf{w}}_{t-1} - \mathbf{w}_{t-1}^k\|^2 + \|\alpha_{t-1}\mathbf{g}_{t-1} - \alpha_{t-1}\mathbf{g}_{t-1,k}\|^2\right)
\end{aligned}$$

and

$$\begin{aligned}
&\sum_k p_k \|\mathbf{g}_{t-1,k} - \mathbf{g}_{t-1}\|^2 = \sum_k p_k (\|\mathbf{g}_{t-1,k}\|^2 - \|\mathbf{g}_{t-1}\|^2) \\
&= \sum_k p_k \|\nabla F_k(\overline{\mathbf{w}}_{t-1}, \xi_{t-1}^k) + \nabla F_k(\mathbf{w}_{t-1}^k, \xi_{t-1}^k) - \nabla F_k(\overline{\mathbf{w}}_{t-1}, \xi_{t-1}^k)\|^2 - \|\mathbf{g}_{t-1}\|^2 \\
&\leq 2\sum_k p_k \left(\|\nabla F_k(\overline{\mathbf{w}}_{t-1}, \xi_{t-1}^k)\|^2 + l^2 \|\mathbf{w}_{t-1}^k - \overline{\mathbf{w}}_{t-1}\|^2\right) - \|\mathbf{g}_{t-1}\|^2
\end{aligned}$$

so that using the l-smoothness of ℓ,

$$\begin{aligned}
&\mathbb{E}\sum_{k=1}^{N} p_k\|\overline{\mathbf{w}}_t - \mathbf{w}_t^k\|^2 \\
\leq&\mathbb{E}2(1+2l^2\alpha_{t-1}^2)\sum_{k=1}^{N} p_k\|\overline{\mathbf{w}}_{t-1} - \mathbf{w}_{t-1}^k\|^2 + 4\alpha_{t-1}^2\sum_k p_k\|\nabla F_k(\overline{\mathbf{w}}_{t-1},\xi_{t-1}^k)\|^2 \\
&- 2\alpha_{t-1}^2\|\mathbf{g}_{t-1}\|^2 \\
\leq&\mathbb{E}2(1+2l^2\alpha_{t-1}^2)\sum_{k=1}^{N} p_k\|\overline{\mathbf{w}}_{t-1} - \mathbf{w}_{t-1}^k\|^2 + 4\alpha_{t-1}^2 2l\sum_k p_k(F_k(\overline{\mathbf{w}}_{t-1},\xi_{t-1}^k) \\
&- F_k(\mathbf{w}^*,\xi_{t-1}^k)) - 2\alpha_{t-1}^2\|\mathbf{g}_{t-1}\|^2 \\
=&\mathbb{E}2(1+2l^2\alpha_{t-1}^2)\sum_{k=1}^{N} p_k\|\overline{\mathbf{w}}_{t-1} - \mathbf{w}_{t-1}^k\|^2 + 8\alpha_{t-1}^2 lF(\overline{\mathbf{w}}_{t-1}) - 2\alpha_{t-1}^2\|\mathbf{g}_{t-1}\|^2
\end{aligned}$$

Using the first inequality, we have

$$\begin{aligned}
\mathbb{E}\|\overline{\mathbf{w}}_{t+1} - \mathbf{w}^*\|^2 \leq& \mathbb{E}(1-\alpha_t\mu)\|\overline{\mathbf{w}}_t - \mathbf{w}^*\|^2 \\
&- 2\alpha_t F(\overline{\mathbf{w}}_t) + 2\alpha_t^2\|\sum_{k=1}^{N} p_k\nabla F_k(\overline{\mathbf{w}}_t,\xi_t^k)\|^2 \\
&+ (2\alpha_t^2 l^2 + \alpha_t L)\sum_{k=1}^{N} p_k\|\overline{\mathbf{w}}_t - \mathbf{w}_t^k\|^2
\end{aligned}$$

and we choose α_t and α_{t-1} such that $\mathbb{E}(F(\overline{\mathbf{w}}_t) - \alpha_t\|\sum_{k=1}^{N} p_k\nabla F_k(\overline{\mathbf{w}}_t,\xi_t^k)\|^2) \geq 0$ and $(2\alpha_t^2 l^2 + \alpha_t L) \leq (1-\alpha_t\mu)(2\alpha_{t-1}^2 l^2 + \alpha_{t-1}L)/3$. This gives

$$\begin{aligned}
\mathbb{E}\|\overline{\mathbf{w}}_{t+1} - \mathbf{w}^*\|^2 \leq& \mathbb{E}(1-\alpha_t\mu)[(1-\alpha_{t-1}\mu)\|\overline{\mathbf{w}}_{t-1} - \mathbf{w}^*\|^2 - 2\alpha_{t-1}F(\overline{\mathbf{w}}_{t-1}) \\
&+ 2\alpha_{t-1}^2\|\sum_{k=1}^{N} p_k\nabla F_k(\overline{\mathbf{w}}_{t-1},\xi_{t-1}^k)\|^2
\end{aligned}$$

$$+ (2\alpha_{t-1}^2 l^2 + \alpha_{t-1} L)(\sum_{k=1}^{N} p_k \|\overline{\mathbf{w}}_{t-1} - \mathbf{w}_{t-1}^k\|^2 + \sum_{k=1}^{N} p_k \|\overline{\mathbf{w}}_t - \mathbf{w}_t^k\|^2)/3]$$

Using the second inequality

$$\begin{aligned}
&\sum_{k=1}^{N} p_k \|\overline{\mathbf{w}}_t - \mathbf{w}_t^k\|^2 \\
\leq &\mathbb{E} 2(1 + 2l^2\alpha_{t-1}^2) \sum_{k=1}^{N} p_k \|\overline{\mathbf{w}}_{t-1} - \mathbf{w}_{t-1}^k\|^2 + 8\alpha_{t-1}^2 l F(\overline{\mathbf{w}}_{t-1}) - 2\alpha_{t-1}^2 \|\mathbf{g}_{t-1}\|^2
\end{aligned}$$

and that $2(1 + 2l^2\alpha_{t-1}^2) \leq 3$, $2\alpha_{t-1}^2 l^2 + \alpha_{t-1} L \leq 1$, we have

$$\begin{aligned}
\mathbb{E}\|\overline{\mathbf{w}}_{t+1} - \mathbf{w}^*\|^2 \leq \mathbb{E}(1 - \alpha_t \mu)[&(1 - \alpha_{t-1}\mu)\|\overline{\mathbf{w}}_{t-1} - \mathbf{w}^*\|^2 \\
&- 2\alpha_{t-1} F(\overline{\mathbf{w}}_{t-1}) + 2\alpha_{t-1}^2 \|\sum_{k=1}^{N} p_k \nabla F_k(\overline{\mathbf{w}}_{t-1}, \xi_{t-1}^k)\|^2 + 8\alpha_{t-1}^2 l F(\overline{\mathbf{w}}_{t-1}) \\
&+ (2\alpha_{t-1}^2 l^2 + \alpha_{t-1} L)(2 \sum_{k=1}^{N} p_k \|\overline{\mathbf{w}}_{t-1} - \mathbf{w}_{t-1}^k\|^2)]
\end{aligned}$$

and if α_{t-1} is chosen such that

$$(F(\overline{\mathbf{w}}_{t-1}) - 4\alpha_{t-1} l F(\overline{\mathbf{w}}_{t-1})) - \alpha_{t-1} \|\sum_{k=1}^{N} p_k \nabla F_k(\overline{\mathbf{w}}_{t-1}, \xi_{t-1}^k)\|^2 \geq 0$$

and

$$\begin{aligned}
&(2\alpha_{t-1}^2 l^2 + \alpha_{t-1} L)(1 - \alpha_{t-1}\mu) \\
&\qquad \leq (2\alpha_{t-2}^2 l^2 + \alpha_{t-2} L)/3
\end{aligned}$$

we again have

$$\mathbb{E}\|\overline{\mathbf{w}}_{t+1} - \mathbf{w}^*\|^2 \leq \mathbb{E}(1-\alpha_t\mu)(1-\alpha_{t-1}\mu)[\|\overline{\mathbf{w}}_{t-1} - \mathbf{w}^*\|^2 + (2\alpha_{t-2}^2 l^2 + \alpha_{t-2}L)\cdot(2\sum_{k=1}^{N} p_k\|\overline{\mathbf{w}}_{t-1} - \mathbf{w}_{t-1}^k\|^2)/3]$$

Applying the above derivation iteratively $\tau < E$ times, we have

$$\begin{aligned}\mathbb{E}\|\overline{\mathbf{w}}_{t+1} - \mathbf{w}^*\|^2 \leq\ & \mathbb{E}(1-\alpha_t\mu)\cdots(1-\alpha_{t-\tau+1}\mu)[(1-\alpha_{t-\tau}\mu)\|\overline{\mathbf{w}}_{t-\tau} - \mathbf{w}^*\|^2 \\ & - 2\alpha_{t-\tau}F(\overline{\mathbf{w}}_{t-\tau}) + 2\alpha_{t-\tau}^2\|\sum_{k=1}^{N} p_k\nabla F_k(\overline{\mathbf{w}}_{t-\tau}, \xi_{t-\tau}^k)\|^2 + 8\tau\alpha_{t-\tau}^2 lF(\overline{\mathbf{w}}_{t-\tau}) \\ & + (2\alpha_{t-\tau}^2 l^2 + \alpha_{t-\tau}L)((\tau+1)\sum_{k=1}^{N} p_k\|\overline{\mathbf{w}}_{t-\tau} - \mathbf{w}_{t-\tau}^k\|^2)]\end{aligned}$$

as long as the step sizes $\alpha_{t-\tau}$ are chosen such that the following inequalities hold

$$(2\alpha_{t-\tau}^2 l^2 + \alpha_{t-\tau}L)(1-\alpha_{t-\tau}\mu) \leq (2\alpha_{t-\tau-1}^2 l^2 + \alpha_{t-\tau-1}L)/3$$

$$2(1 + 2l^2\alpha_{t-\tau}^2) \leq 3$$

$$2\alpha_{t-\tau}^2 l^2 + \alpha_{t-\tau}L \leq 1$$

$$(F(\overline{\mathbf{w}}_{t-\tau}) - 4\tau\alpha_{t-\tau}lF(\overline{\mathbf{w}}_{t-\tau})) - \alpha_{t-\tau}\|\sum_{k=1}^{N} p_k\nabla F_k(\overline{\mathbf{w}}_{t-\tau}, \xi_{t-\tau}^k)\|^2 \geq 0$$

We can check that setting $\alpha_{t-\tau} = c\frac{1}{\tau+1}\frac{N}{l\nu_{\max}+L(N-\nu_{\min})}$ for some small constant c satisfies the requirements.

Since communication is done every E iterations, $\overline{\mathbf{w}}_{t_0} = \mathbf{w}_{t_0}^k$ for some $t_0 > t - E$, from

which we can conclude that

$$\begin{aligned}\mathbb{E}\|\overline{\mathbf{w}}_t - \mathbf{w}^*\|^2 &\leq (\prod_{\tau=1}^{t-t_0-1}(1-\mu\alpha_{t-\tau}))\|\mathbf{w}_{t_0} - \mathbf{w}^*\|^2 \\ &\leq (1 - c\frac{\mu}{E}\frac{N}{l\nu_{\max} + L(N-\nu_{\min})})^{t-t_0}\|\mathbf{w}_{t_0} - \mathbf{w}^*\|^2\end{aligned}$$

and applying this inequality to iterations between each communication round,

$$\begin{aligned}\mathbb{E}\|\overline{\mathbf{w}}_t - \mathbf{w}^*\|^2 &\leq (1 - c\frac{\mu}{E}\frac{N}{l\nu_{\max} + L(N-\nu_{\min})})^{t}\|\mathbf{w}_0 - \mathbf{w}^*\|^2 \\ &= O(\exp(\frac{\mu}{E}\frac{N}{l\nu_{\max} + L(N-\nu_{\min})}t))\|\mathbf{w}_0 - \mathbf{w}^*\|^2\end{aligned}$$

With partial participation, we note that

$$\begin{aligned}\mathbb{E}\|\overline{\mathbf{w}}_{t+1} - \mathbf{w}^*\|^2 &= \mathbb{E}\|\overline{\mathbf{w}}_{t+1} - \overline{\mathbf{v}}_{t+1} + \overline{\mathbf{v}}_{t+1} - \mathbf{w}^*\|^2 \\ &= \mathbb{E}\|\overline{\mathbf{w}}_{t+1} - \overline{\mathbf{v}}_{t+1}\|^2 + \mathbb{E}\|\overline{\mathbf{v}}_{t+1} - \mathbf{w}^*\|^2 \\ &= \frac{1}{K}\sum_k p_k \mathbb{E}\|\mathbf{w}_{t+1}^k - \overline{\mathbf{w}}_{t+1}\|^2 + \mathbb{E}\|\overline{\mathbf{v}}_{t+1} - \mathbf{w}^*\|^2\end{aligned}$$

and so the recursive identity becomes

$$\begin{aligned}\mathbb{E}\|\overline{\mathbf{w}}_{t+1} - \mathbf{w}^*\|^2 \leq\ & \mathbb{E}(1-\alpha_t\mu)\cdots(1-\alpha_{t-\tau+1}\mu)[(1-\alpha_{t-\tau}\mu)\|\overline{\mathbf{w}}_{t-\tau} - \mathbf{w}^*\|^2 \\ & - 2\alpha_{t-\tau}F(\overline{\mathbf{w}}_{t-\tau}) + 2\alpha_{t-\tau}^2\|\sum_{k=1}^N p_k \nabla F_k(\overline{\mathbf{w}}_{t-\tau}, \xi_{t-\tau}^k)\|^2 + 8\tau\alpha_{t-\tau}^2 l F(\overline{\mathbf{w}}_{t-\tau}) \\ & + (2\alpha_{t-\tau}^2 l^2 + \alpha_{t-\tau}L + \frac{1}{K})((\tau+1)\sum_{k=1}^N p_k\|\overline{\mathbf{w}}_{t-\tau} - \mathbf{w}_{t-\tau}^k\|^2)]\end{aligned}$$

which requires

$$(2\alpha_{t-\tau}^2 l^2 + \alpha_{t-\tau}L + \frac{1}{K})(1-\alpha_{t-\tau}\mu) \leq (2\alpha_{t-\tau-1}^2 l^2 + \alpha_{t-\tau-1}L + \frac{1}{K})/3$$

$$2(1+2l^2\alpha_{t-\tau}^2) \leq 3$$

$$2\alpha_{t-\tau}^2 l^2 + \alpha_{t-\tau}L + \frac{1}{K} \leq 1$$

$$(F(\overline{\mathbf{w}}_{t-\tau}) - 4\tau\alpha_{t-\tau}lF(\overline{\mathbf{w}}_{t-\tau})) - \alpha_{t-\tau}\|\sum_{k=1}^{N} p_k \nabla F_k(\overline{\mathbf{w}}_{t-\tau}, \xi_{t-\tau}^k)\|^2 \geq 0$$

to hold. Again setting $\alpha_{t-\tau} = c\frac{1}{\tau+1}\frac{N}{l\nu_{\max}+L(N-\nu_{\min})}$ for a possibly different constant from before satisfies the requirements.

Finally, using the L-smoothness of F,

$$F(\overline{\mathbf{w}}_T) - F(\mathbf{w}^*) \leq \frac{L}{2}\mathbb{E}\|\overline{\mathbf{w}}_T - \mathbf{w}^*\|^2 = O(L\exp(-\frac{\mu}{E}\frac{N}{l\nu_{\max}+L(N-\nu_{\min})}T))\|\mathbf{w}_0 - \mathbf{w}^*\|^2$$

□

Geometric Convergence of FedAvg for Overparameterized Linear Regression

We first provide details on quantities used in the proof of results on linear regression in Section 6.5 in the main text. The local device objectives are now given by the sum of squares $F_k(\mathbf{w}) = \frac{1}{2n_k}\sum_{j=1}^{n_k}(\mathbf{w}^T\mathbf{x}_k^j - \mathbf{z}_k^j)^2$, and there exists $\mathbf{w}^*$ such that $F(\mathbf{w}^*) \equiv 0$. Define the local Hessian matrix as $\mathbf{H}^k := \frac{1}{n_k}\sum_{j=1}^{n_k}\mathbf{x}_k^j(\mathbf{x}_k^j)^T$, and the stochastic Hessian matrix as $\tilde{\mathbf{H}}_t^k := \xi_t^k(\xi_t^k)^T$, where ξ_t^k is the stochastic sample on the kth device at time t. Define l to be the smallest positive number such that $\mathbb{E}\|\xi_t^k\|^2\xi_t^k(\xi_t^k)^T \preceq l\mathbf{H}^k$ for all k. Note that

$l \leq \max_{k,j} \|\mathbf{x}_k^j\|^2$. Let L and μ be lower and upper bounds of non-zero eigenvalues of $\mathbf{H}^k$. Define $\kappa_1 := l/\mu$ and $\kappa := L/\mu$.

Following [119, 87], we define the statistical condition number $\tilde{\kappa}$ as the smallest positive real number such that $\mathbb{E}\sum_k p_k \tilde{\mathbf{H}}_t^k \mathbf{H}^{-1} \tilde{\mathbf{H}}_t^k \leq \tilde{\kappa}\mathbf{H}$. The condition numbers κ_1 and $\tilde{\kappa}$ are important in the characterization of convergence rates for FedAvg algorithms. Note that $\kappa_1 > \kappa$ and $\kappa_1 > \tilde{\kappa}$.

Let $\mathbf{H} = \sum_k p_k \mathbf{H}^k$. In general $\mathbf{H}$ has zero eigenvalues. However, because the null space of $\mathbf{H}$ and range of $\mathbf{H}$ are orthogonal, in our subsequence analysis it suffices to project $\overline{\mathbf{w}}_t - \mathbf{w}^*$ onto the range of $\mathbf{H}$, thus we may restrict to the non-zero eigenvalue of $\mathbf{H}$.

A useful observation is that we can use $\mathbf{w}^{*T}\mathbf{x}_k^j - \mathbf{z}_k^j \equiv 0$ to rewrite the local objectives as $F_k(\mathbf{w}) = \frac{1}{2}\langle \mathbf{w} - \mathbf{w}^*, \mathbf{H}^k(\mathbf{w} - \mathbf{w}^*)\rangle \equiv \frac{1}{2}\|\mathbf{w} - \mathbf{w}^*\|_{\mathbf{H}^k}^2$:

$$
\begin{aligned}
F_k(\mathbf{w}) &= \frac{1}{2n_k}\sum_{j=1}^{n_k}(\mathbf{w}^T\mathbf{x}_{k,j} - \mathbf{z}_{k,j} - (\mathbf{w}^{*T}\mathbf{x}_{k,j} - \mathbf{z}_{k,j}))^2 = \frac{1}{2n_k}\sum_{j=1}^{n_k}((\mathbf{w} - \mathbf{w}^*)^T\mathbf{x}_{k,j})^2 \\
&= \frac{1}{2}\langle \mathbf{w} - \mathbf{w}^*, \mathbf{H}^k(\mathbf{w} - \mathbf{w}^*)\rangle = \frac{1}{2}\|\mathbf{w} - \mathbf{w}^*\|_{\mathbf{H}^k}^2
\end{aligned}
$$

so that $F(\mathbf{w}) = \frac{1}{2}\|\mathbf{w} - \mathbf{w}^*\|_H^2$.

Finally, note that $\mathbb{E}\tilde{\mathbf{H}}_t^k = \frac{1}{n_k}\sum_{j=1}^{n_k}\mathbf{x}_k^j(\mathbf{x}_k^j)^T = \mathbf{H}^k$ and $\mathbf{g}_{t,k} = \tilde{\mathbf{H}}_t^k(\mathbf{w}_t^k - \mathbf{w}^*)$ while $\mathbf{g}_t = \sum_{k=1}^N p_k \nabla F_k(\mathbf{w}_t^k, \xi_t^k) = \sum_{k=1}^N p_k \tilde{\mathbf{H}}_t^k(\mathbf{w}_t^k - \mathbf{w}^*)$ and $\overline{\mathbf{g}}_t = \sum_{k=1}^N p_k \mathbf{H}^k(\mathbf{w}_t^k - \mathbf{w}^*)$

Theorem 19. *For the overparamterized linear regression problem, FedAvg with communication every E iterations with constant step size $\overline{\alpha} = \mathcal{O}(\frac{1}{E}\frac{N}{l\nu_{\max} + \mu(N - \nu_{\min})})$ has geometric*

convergence:

$$\mathbb{E}F(\overline{\mathbf{w}}_T) \leq \mathcal{O}\left(L\exp(-\frac{NT}{E(\nu_{\max}\kappa_1 + (N-\nu_{\min}))})\|\mathbf{w}_0 - \mathbf{w}^*\|^2\right).$$

Proof. We again show the result first when $E = 2$ and $t-1$ is a communication round. We have

$$\begin{aligned}\|\overline{\mathbf{w}}_{t+1} - \mathbf{w}^*\|^2 &= \|(\overline{\mathbf{w}}_t - \alpha_t \mathbf{g}_t) - \mathbf{w}^*\|^2 \\ &= \|\overline{\mathbf{w}}_t - \mathbf{w}^*\|^2 - 2\alpha_t\langle \overline{\mathbf{w}}_t - \mathbf{w}^*, \mathbf{g}_t\rangle + \alpha_t^2\|\mathbf{g}_t\|^2\end{aligned}$$

and

$$\begin{aligned}&- 2\alpha_t \mathbb{E}_t\langle \overline{\mathbf{w}}_t - \mathbf{w}^*, \mathbf{g}_t\rangle \\ &= -2\alpha_t \sum_{k=1}^N p_k \langle \overline{\mathbf{w}}_t - \mathbf{w}^*, \nabla F_k(\mathbf{w}_t^k)\rangle \\ &= -2\alpha_t \sum_{k=1}^N p_k \langle \overline{\mathbf{w}}_t - \mathbf{w}_t^k, \nabla F_k(\mathbf{w}_t^k)\rangle - 2\alpha_t \sum_{k=1}^N p_k \langle \mathbf{w}_t^k - \mathbf{w}^*, \nabla F_k(\mathbf{w}_t^k)\rangle \\ &= -2\alpha_t \sum_{k=1}^N p_k \langle \overline{\mathbf{w}}_t - \mathbf{w}_t^k, \nabla F_k(\mathbf{w}_t^k)\rangle - 2\alpha_t \sum_{k=1}^N p_k \langle \mathbf{w}_t^k - \mathbf{w}^*, \mathbf{H}^k(\mathbf{w}_t^k - \mathbf{w}^*)\rangle \\ &= -2\alpha_t \sum_{k=1}^N p_k \langle \overline{\mathbf{w}}_t - \mathbf{w}_t^k, \nabla F_k(\mathbf{w}_t^k)\rangle - 4\alpha_t \sum_{k=1}^N p_k F_k(\mathbf{w}_t^k) \\ &\leq 2\alpha_t \sum_{k=1}^N p_k (F_k(\mathbf{w}_t^k) - F_k(\overline{\mathbf{w}}_t) + \frac{L}{2}\|\overline{\mathbf{w}}_t - \mathbf{w}_t^k\|^2) - 4\alpha_t \sum_{k=1}^N p_k F_k(\mathbf{w}_t^k) \\ &= \alpha_t L \sum_{k=1}^N p_k \|\overline{\mathbf{w}}_t - \mathbf{w}_t^k\|^2 - 2\alpha_t \sum_{k=1}^N p_k F_k(\overline{\mathbf{w}}_t) - 2\alpha_t \sum_{k=1}^N p_k F_k(\mathbf{w}_t^k) \\ &= \alpha_t L \sum_{k=1}^N p_k \|\overline{\mathbf{w}}_t - \mathbf{w}_t^k\|^2 - \alpha_t \sum_{k=1}^N p_k \langle (\overline{\mathbf{w}}_t - \mathbf{w}^*), \mathbf{H}^k(\overline{\mathbf{w}}_t - \mathbf{w}^*)\rangle - 2\alpha_t \sum_{k=1}^N p_k F_k(\mathbf{w}_t^k)\end{aligned}$$

and

$$\begin{aligned}\|\mathbf{g}_t\|^2 &= \|\sum_{k=1}^N p_k \tilde{\mathbf{H}}_t^k(\mathbf{w}_t^k - \mathbf{w}^*)\|^2 \\ &= \|\sum_{k=1}^N p_k \tilde{\mathbf{H}}_t^k(\overline{\mathbf{w}}_t - \mathbf{w}^*) + \sum_{k=1}^N p_k \tilde{\mathbf{H}}_t^k(\mathbf{w}_t^k - \overline{\mathbf{w}}_t)\|^2 \\ &\leq 2\|\sum_{k=1}^N p_k \tilde{\mathbf{H}}_t^k(\overline{\mathbf{w}}_t - \mathbf{w}^*)\|^2 + 2\|\sum_{k=1}^N p_k \tilde{\mathbf{H}}_t^k(\mathbf{w}_t^k - \overline{\mathbf{w}}_t)\|^2\end{aligned}$$

which gives

$$\begin{aligned}&\mathbb{E}\|\overline{\mathbf{w}}_{t+1} - \mathbf{w}^*\|^2 \\ \leq&\mathbb{E}\|\overline{\mathbf{w}}_t - \mathbf{w}^*\|^2 - \alpha_t \sum_{k=1}^N p_k \langle \overline{\mathbf{w}}_t - \mathbf{w}^*, \mathbf{H}^k \overline{\mathbf{w}}_t - \mathbf{w}^* \rangle + 2\alpha_t^2 \|\sum_{k=1}^N p_k \tilde{\mathbf{H}}_t^k(\overline{\mathbf{w}}_t - \mathbf{w}^*)\|^2 \\ &+ \alpha_t L \sum_{k=1}^N p_k \|\overline{\mathbf{w}}_t - \mathbf{w}_t^k\|^2 + 2\alpha_t^2 \|\sum_{k=1}^N p_k \tilde{\mathbf{H}}_t^k(\mathbf{w}_t^k - \overline{\mathbf{w}}_t)\|^2 - 2\alpha_t \sum_{k=1}^N p_k F_k(\mathbf{w}_t^k)\end{aligned}$$

following [127] we first prove that

$$\begin{aligned}\mathbb{E}\|\overline{\mathbf{w}}_t - \mathbf{w}^*\|^2 - \alpha_t \sum_{k=1}^N p_k \langle (\overline{\mathbf{w}}_t - \mathbf{w}^*), \mathbf{H}^k(\overline{\mathbf{w}}_t - \mathbf{w}^*) \rangle + 2\alpha_t^2 \|\sum_{k=1}^N p_k \tilde{\mathbf{H}}_t^k(\overline{\mathbf{w}}_t - \mathbf{w}^*)\|^2 \\ \leq (1 - \frac{N}{8(\nu_{\max}\kappa_1 + (N - \nu_{\min}))})\mathbb{E}\|\overline{\mathbf{w}}_t - \mathbf{w}^*\|^2\end{aligned}$$

with appropriately chosen α_t. Compared to the rate $\mathcal{O}(\frac{N}{\nu_{\max}\kappa_1 + (N - \nu_{\min})\kappa})$ for general strongly convex and smooth objectives, this is an improvement as linear speedup is now available for a larger range of N.

We have

$$
\begin{aligned}
&\mathbb{E}_t\|\sum_{k=1}^{N} p_k\tilde{\mathbf{H}}_t^k(\overline{\mathbf{w}}_t-\mathbf{w}^*)\|^2\\
&=\mathbb{E}_t\langle\sum_{k=1}^{N} p_k\tilde{\mathbf{H}}_t^k(\overline{\mathbf{w}}_t-\mathbf{w}^*),\sum_{k=1}^{N} p_k\tilde{\mathbf{H}}_t^k(\overline{\mathbf{w}}_t-\mathbf{w}^*)\rangle\\
&=\sum_{k=1}^{N} p_k^2\mathbb{E}_t\|\tilde{\mathbf{H}}_t^k(\overline{\mathbf{w}}_t-\mathbf{w}^*)\|^2+\sum_{k=1}^{N}\sum_{j\neq k} p_jp_k\mathbb{E}_t\langle\tilde{\mathbf{H}}_t^k(\overline{\mathbf{w}}_t-\mathbf{w}^*),\tilde{\mathbf{H}}_t^j(\overline{\mathbf{w}}_t-\mathbf{w}^*)\rangle\\
&=\sum_{k=1}^{N} p_k^2\mathbb{E}_t\|\tilde{\mathbf{H}}_t^k(\overline{\mathbf{w}}_t-\mathbf{w}^*)\|^2+\sum_{k=1}^{N}\sum_{j\neq k} p_jp_k\mathbb{E}_t\langle\mathbf{H}^k(\overline{\mathbf{w}}_t-\mathbf{w}^*),\mathbf{H}^j(\overline{\mathbf{w}}_t-\mathbf{w}^*)\rangle\\
&=\sum_{k=1}^{N} p_k^2\mathbb{E}_t\|\tilde{\mathbf{H}}_t^k(\overline{\mathbf{w}}_t-\mathbf{w}^*)\|^2+\sum_{k=1}^{N}\sum_{j=1}^{N} p_jp_k\mathbb{E}_t\langle\mathbf{H}^k(\overline{\mathbf{w}}_t-\mathbf{w}^*),\mathbf{H}^j(\overline{\mathbf{w}}_t-\mathbf{w}^*)\rangle\\
&\quad-\sum_{k=1}^{N} p_k^2\|\mathbf{H}^k(\overline{\mathbf{w}}_t-\mathbf{w}^*)\|^2\\
&=\sum_{k=1}^{N} p_k^2\mathbb{E}_t\|\tilde{\mathbf{H}}_t^k(\overline{\mathbf{w}}_t-\mathbf{w}^*)\|^2+\|\sum_{k} p_k\mathbf{H}^k(\overline{\mathbf{w}}_t-\mathbf{w}^*)\|^2-\sum_{k=1}^{N} p_k^2\|\mathbf{H}^k(\overline{\mathbf{w}}_t-\mathbf{w}^*)\|^2\\
&\leq\sum_{k=1}^{N} p_k^2\mathbb{E}_t\|\tilde{\mathbf{H}}_t^k(\overline{\mathbf{w}}_t-\mathbf{w}^*)\|^2+\|\sum_{k} p_k\mathbf{H}^k(\overline{\mathbf{w}}_t-\mathbf{w}^*)\|^2-\frac{1}{N}\nu_{\min}\|\sum_{k} p_k\mathbf{H}^k(\overline{\mathbf{w}}_t-\mathbf{w}^*)\|^2\\
&\leq\frac{1}{N}\nu_{\max}\sum_{k=1}^{N} p_k\mathbb{E}_t\|\tilde{\mathbf{H}}_t^k(\overline{\mathbf{w}}_t-\mathbf{w}^*)\|^2+(1-\frac{1}{N}\nu_{\min})\|\sum_{k} p_k\mathbf{H}^k(\overline{\mathbf{w}}_t-\mathbf{w}^*)\|^2\\
&\leq\frac{1}{N}\nu_{\max}l\sum_{k=1}^{N} p_k\langle(\overline{\mathbf{w}}_t-\mathbf{w}^*),\mathbf{H}^k(\overline{\mathbf{w}}_t-\mathbf{w}^*)\rangle+(1-\frac{1}{N}\nu_{\min})\|\sum_{k} p_k\mathbf{H}^k(\overline{\mathbf{w}}_t-\mathbf{w}^*)\|^2\\
&=\frac{1}{N}\nu_{\max}l\langle(\overline{\mathbf{w}}_t-\mathbf{w}^*),\mathbf{H}(\overline{\mathbf{w}}_t-\mathbf{w}^*)\rangle+(1-\frac{1}{N}\nu_{\min})\langle\overline{\mathbf{w}}_t-\mathbf{w}^*,\mathbf{H}^2(\overline{\mathbf{w}}_t-\mathbf{w}^*)\rangle
\end{aligned}
$$

using $\|\tilde{\mathbf{H}}_t^k\|\leq l$.

Now we have

$$
\mathbb{E}\|\overline{\mathbf{w}}_t-\mathbf{w}^*\|^2-\alpha_t\sum_{k=1}^{N} p_k\langle(\overline{\mathbf{w}}_t-\mathbf{w}^*),\mathbf{H}^k(\overline{\mathbf{w}}_t-\mathbf{w}^*)\rangle+2\alpha_t^2\|\sum_{k=1}^{N} p_k\tilde{\mathbf{H}}_t^k(\overline{\mathbf{w}}_t-\mathbf{w}^*)\|^2=
$$

$$\langle \overline{\mathbf{w}}_t - \mathbf{w}^*, (I - \alpha_t \mathbf{H} + 2\alpha_t^2 (\frac{\nu_{\max} l}{N} \mathbf{H} + \frac{N - \nu_{\min}}{N} \mathbf{H}^2))(\overline{\mathbf{w}}_t - \mathbf{w}^*) \rangle$$

and it remains to bound the maximum eigenvalue of

$$(I - \alpha_t \mathbf{H} + 2\alpha_t^2 (\frac{\nu_{\max} l}{N} \mathbf{H} + \frac{N - \nu_{\min}}{N} \mathbf{H}^2))$$

and we bound this following [127]. If we choose $\alpha_t < \frac{N}{2(\nu_{\max} l + (N - \nu_{\min})L)}$, then

$$-\alpha_t \mathbf{H} + 2\alpha_t^2 (\frac{\nu_{\max} l}{N} \mathbf{H} + \frac{N - \nu_{\min}}{N} \mathbf{H}^2) \prec 0$$

and the convergence rate is given by the maximum of $1 - \alpha_t \lambda + 2\alpha_t^2 (\frac{\nu_{\max} l}{N} \lambda + \frac{N - \nu_{\min}}{N} \lambda^2)$ maximized over the non-zero eigenvalues λ of $\mathbf{H}$. To select the step size α_t that gives the smallest upper bound, we then minimize over α_t, resulting in

$$\min_{\alpha_t < \frac{N}{2(\nu_{\max} l + (N - \nu_{\min})L)}} \max_{\lambda > 0 : \exists v, \mathbf{H} v = \lambda v} \left\{ 1 - \alpha_t \lambda + 2\alpha_t^2 (\frac{\nu_{\max} l}{N} \lambda + \frac{N - \nu_{\min}}{N} \lambda^2) \right\}$$

Since the objective is quadratic in λ, the maximum is achieved at either the largest eigenvalue $\lambda_{\max}$ of $\mathbf{H}$ or the smallest non-zero eigenvalue $\lambda_{\min}$ of $\mathbf{H}$.

When $N \leq \frac{4\nu_{\max} l}{L - \lambda_{\min}} + 4\nu_{\min}$, i.e. when $N = O(l/\lambda_{\min}) = O(\kappa_1)$, the optimal objective value is achieved at $\lambda_{\min}$ and the optimal step size is given by $\alpha_t = \frac{N}{4(\nu_{\max} l + (N - \nu_{\min})\lambda_{\min})}$. The optimal convergence rate (i.e. the optimal objective value) is equal to

$$1 - \frac{1}{8} \frac{N \lambda_{\min}}{(\nu_{\max} l + (N - \nu_{\min})\lambda_{\min})} = 1 - \frac{1}{8} \frac{N}{(\nu_{\max} \kappa_1 + (N - \nu_{\min}))}.$$

This implies that when $N = O(\kappa_1)$, the optimal convergence rate has a linear speedup in N.

When N is larger, this step size is no longer optimal, but we still have $1-\frac{1}{8}\frac{N}{(\nu_{\max}\kappa_1+(N-\nu_{\min}))}$ as an upper bound on the convergence rate.

Now we have proved

$$\begin{aligned}
&\mathbb{E}\|\overline{\mathbf{w}}_{t+1}-\mathbf{w}^*\|^2\\
\leq&(1-\frac{1}{8}\frac{N}{(\nu_{\max}\kappa_1+(N-\nu_{\min}))})\mathbb{E}\|\overline{\mathbf{w}}_t-\mathbf{w}^*\|^2\\
&+\alpha_t L\sum_{k=1}^N p_k\|\overline{\mathbf{w}}_t-\mathbf{w}_t^k\|^2+2\alpha_t^2\|\sum_{k=1}^N p_k\tilde{\mathbf{H}}_t^k(\mathbf{w}_t^k-\overline{\mathbf{w}}_t)\|^2-2\alpha_t\sum_{k=1}^N p_k F_k(\mathbf{w}_t^k)
\end{aligned}$$

Next we bound terms in the second line using a similar argument as the general case. We have

$$2\alpha_t^2\|\sum_{k=1}^N p_k\tilde{\mathbf{H}}_t^k(\mathbf{w}_t^k-\overline{\mathbf{w}}_t)\|^2\leq 2\alpha_t^2 l^2\sum_{k=1}^N p_k\|\overline{\mathbf{w}}_t-\mathbf{w}_t^k\|^2$$

and

$$\begin{aligned}
\mathbb{E}\sum_{k=1}^N p_k\|\overline{\mathbf{w}}_t-\mathbf{w}_t^k\|^2&\leq\mathbb{E}2(1+2l^2\alpha_{t-1}^2)\sum_{k=1}^N p_k\|\overline{\mathbf{w}}_{t-1}-\mathbf{w}_{t-1}^k\|^2+8\alpha_{t-1}^2 lF(\overline{\mathbf{w}}_{t-1})\\
&=4\alpha_{t-1}^2 l\langle\overline{\mathbf{w}}_{t-1}-\mathbf{w}^*,\mathbf{H}(\overline{\mathbf{w}}_{t-1}-\mathbf{w}^*)\rangle
\end{aligned}$$

and if α_t,α_{t-1} satisfy

$$\begin{aligned}
\alpha_t L+2\alpha_t^2&\leq(1-\frac{1}{8}\frac{N}{(\nu_{\max}\kappa_1+(N-\nu_{\min}))})(\alpha_{t-1}L+2\alpha_{t-1}^2)/3\\
2(1+2l^2\alpha_{t-1}^2)&\leq 3\\
\alpha_t L+2\alpha_t^2&\leq 1
\end{aligned}$$

we have

$$
\begin{aligned}
&\mathbb{E}\|\overline{\mathbf{w}}_{t+1}-\mathbf{w}^*\|^2\\
&\leq (1-\frac{1}{8}\frac{N}{(\nu_{\max}\kappa_1+(N-\nu_{\min}))})[\mathbb{E}\|\overline{\mathbf{w}}_{t-1}-\mathbf{w}^*\|^2-\alpha_t\langle\overline{\mathbf{w}}_{t-1}-\mathbf{w}^*,\mathbf{H}\overline{\mathbf{w}}_{t-1}-\mathbf{w}^*\rangle\\
&+2\alpha_t^2\|\sum_{k=1}^{N}p_k\tilde{\mathbf{H}}_t^k(\overline{\mathbf{w}}_t-\mathbf{w}^*)\|^2\\
&+(\alpha_{t-1}L+2\alpha_{t-1}^2)\cdot 2\sum_{k=1}^{N}p_k\|\overline{\mathbf{w}}_{t-1}-\mathbf{w}_{t-1}^k\|^2+4\alpha_{t-1}^2 l\langle\overline{\mathbf{w}}_{t-1}-\mathbf{w}^*,\mathbf{H}(\overline{\mathbf{w}}_{t-1}-\mathbf{w}^*)\rangle]
\end{aligned}
$$

and again by choosing $\alpha_{t-1}=c\frac{N}{8(\nu_{\max}l+(N-\nu_{\min})\lambda_{\min})}$ for a small constant c, we can guarantee that

$$
\begin{aligned}
&\mathbb{E}\|\overline{\mathbf{w}}_{t-1}-\mathbf{w}^*\|^2-\alpha_{t-1}\langle\overline{\mathbf{w}}_{t-1}-\mathbf{w}^*,\mathbf{H}\overline{\mathbf{w}}_{t-1}-\mathbf{w}^*\rangle\\
+2\alpha_{t-1}^2\|\sum_{k=1}^{N}p_k\tilde{\mathbf{H}}_{t-1}^k(\overline{\mathbf{w}}_{t-1}-\mathbf{w}^*)\|^2&+4\alpha_{t-1}^2 l\langle\overline{\mathbf{w}}_{t-1}-\mathbf{w}^*,\mathbf{H}(\overline{\mathbf{w}}_{t-1}-\mathbf{w}^*)\rangle\\
&\leq(1-c\frac{N}{16(\nu_{\max}l+(N-\nu_{\min})\lambda_{\min})})\mathbb{E}\|\overline{\mathbf{w}}_{t-1}-\mathbf{w}^*\|^2
\end{aligned}
$$

For general E, we have the recursive relation

$$
\begin{aligned}
&\mathbb{E}\|\overline{\mathbf{w}}_{t+1}-\mathbf{w}^*\|^2\\
\leq&\mathbb{E}(1-c\frac{1}{8}\frac{N}{(\nu_{\max}\kappa_1+(N-\nu_{\min}))})\cdots(1-c\frac{1}{8\tau}\frac{N}{(\nu_{\max}\kappa_1+(N-\nu_{\min}))})[\|\overline{\mathbf{w}}_{t-\tau}-\mathbf{w}^*\|^2\\
&-\alpha_{t-\tau}\langle\overline{\mathbf{w}}_{t-\tau}-\mathbf{w}^*,\mathbf{H}\overline{\mathbf{w}}_{t-\tau}-\mathbf{w}^*\rangle+2\alpha_{t-\tau}^2\|\sum_{k=1}^{N}p_k\tilde{\mathbf{H}}_{t-\tau}^k(\overline{\mathbf{w}}_{t-\tau}-\mathbf{w}^*)\|^2\\
&+4\tau\alpha_{t-1}^2 l\langle\overline{\mathbf{w}}_{t-1}-\mathbf{w}^*,\mathbf{H}(\overline{\mathbf{w}}_{t-1}-\mathbf{w}^*)\rangle\\
&+(2\alpha_{t-\tau}^2 l^2+\alpha_{t-\tau}L)((\tau+1)\sum_{k=1}^{N}p_k\|\overline{\mathbf{w}}_{t-\tau}-\mathbf{w}_{t-\tau}^k\|^2)]
\end{aligned}
$$

as long as the step sizes are chosen $\alpha_{t-\tau} = c\frac{N}{4\tau(\nu_{\max}l+(N-\nu_{\min})\lambda_{\min})}$ such that the following inequalities hold

$$
\begin{aligned}
(2\alpha_{t-\tau}^2 l^2 + \alpha_{t-\tau}L) &\leq (1-\alpha_{t-\tau}\mu)(2\alpha_{t-\tau-1}^2 l^2 + \alpha_{t-\tau-1}L)/3 \\
2(1+2l^2\alpha_{t-\tau}^2) &\leq 3 \\
2\alpha_{t-\tau}^2 l^2 + \alpha_{t-\tau}L &\leq 1
\end{aligned}
$$

and

$$
\begin{aligned}
&\|\overline{\mathbf{w}}_{t-\tau} - \mathbf{w}^*\|^2 - \alpha_{t-\tau}\langle \overline{\mathbf{w}}_{t-\tau} - \mathbf{w}^*, \mathbf{H}\overline{\mathbf{w}}_{t-\tau} - \mathbf{w}^*\rangle \\
&+ 2\alpha_{t-\tau}^2 \|\sum_{k=1}^{N} p_k \tilde{\mathbf{H}}_{t-\tau}^k (\overline{\mathbf{w}}_{t-\tau} - \mathbf{w}^*)\|^2 + 4\tau\alpha_{t-1}^2 l\langle \overline{\mathbf{w}}_{t-1} - \mathbf{w}^*, \mathbf{H}(\overline{\mathbf{w}}_{t-1} - \mathbf{w}^*)\rangle \\
&\leq (1 - c\frac{N}{8(\tau+1)(\nu_{\max}\kappa_1 + (N-\nu_{\min}))})\mathbb{E}\|\overline{\mathbf{w}}_{t-\tau} - \mathbf{w}^*\|^2
\end{aligned}
$$

which gives

$$
\begin{aligned}
\mathbb{E}\|\overline{\mathbf{w}}_t - \mathbf{w}^*\|^2 &\leq (1 - c\frac{1}{8E}\frac{N}{(\nu_{\max}\kappa_1 + (N-\nu_{\min}))})^t \|\mathbf{w}_0 - \mathbf{w}^*\|^2 \\
&= O(\exp(-\frac{1}{E}\frac{N}{(\nu_{\max}\kappa_1 + (N-\nu_{\min}))}t))\|\mathbf{w}_0 - \mathbf{w}^*\|^2
\end{aligned}
$$

and with partial participation, the same bound holds with a possibly different choice of c. $\square$

Geometric Convergence of FedMaSS for Overparameterized Linear Regression

Theorem 20. *For the overparamterized linear regression problem, FedMaSS with communication every E iterations and constant step sizes $\overline{\eta}_1 = \mathcal{O}(\frac{1}{E}\frac{N}{l\nu_{\max}+\mu(N-\nu_{\min})}), \overline{\eta}_2 = \frac{\overline{\eta}_1(1-\frac{1}{\tilde{\kappa}})}{1+\frac{1}{\sqrt{\kappa_1\tilde{\kappa}}}}, \overline{\gamma} = \frac{1-\frac{1}{\sqrt{\kappa_1\tilde{\kappa}}}}{1+\frac{1}{\sqrt{\kappa_1\tilde{\kappa}}}}$ has geometric convergence:*

$$\mathbb{E}F(\overline{\mathbf{w}}_T) \leq \mathcal{O}\left(L\exp(-\frac{NT}{E(\nu_{\max}\sqrt{\kappa_1\tilde{\kappa}}+(N-\nu_{\min}))})\|\mathbf{w}_0-\mathbf{w}^*\|^2\right).$$

Proof. The proof is based on results in [119] which originally proposed the MaSS algorithm. Note that the update can equivalently be written as

$$\begin{aligned}
\mathbf{v}_{t+1}^k &= (1-\alpha^k)\mathbf{v}_t^k + \alpha^k\mathbf{u}_t^k - \delta^k\mathbf{g}_{t,k} \\
\mathbf{w}_{t+1}^k &= \begin{cases} \mathbf{u}_t^k - \eta^k\mathbf{g}_{t,k} & \text{if } t+1 \notin \mathcal{I}_E \\ \sum_{k=1}^N p_k\left[\mathbf{u}_t^k - \eta^k\mathbf{g}_{t,k}\right] & \text{if } t+1 \in \mathcal{I}_E \end{cases} \\
\mathbf{u}_{t+1}^k &= \frac{\alpha^k}{1+\alpha^k}\mathbf{v}_{t+1}^k + \frac{1}{1+\alpha^k}\mathbf{w}_{t+1}^k
\end{aligned}$$

where there is a bijection between the parameters $\frac{1-\alpha^k}{1+\alpha^k} = \gamma^k, \eta^k = \eta_1^k, \frac{\eta^k-\alpha^k\delta^k}{1+\alpha^k} = \eta_2^k$, and we further introduce an auxiliary parameter $\mathbf{v}_t^k$, which is initialized at $\mathbf{v}_0^k$. We also note that when $\delta^k = \frac{\eta^k}{\alpha^k}$, the update reduces to the Nesterov accelerated SGD. This version of the FedAvg algorithm with local MaSS updates is used for analyzing the geometric convergence.

As before, define the virtual sequences $\overline{\mathbf{w}}_t = \sum_{k=1}^N p_k\mathbf{w}_t^k$, $\overline{\mathbf{v}}_t = \sum_{k=1}^N p_k\mathbf{v}_t^k$, $\overline{\mathbf{u}}_t = \sum_{k=1}^N p_k\mathbf{u}_t^k$, and $\overline{\mathbf{g}}_t = \sum_{k=1}^N p_k\mathbb{E}\mathbf{g}_{t,k}$. We have $\mathbb{E}\mathbf{g}_t = \overline{\mathbf{g}}_t$ and $\overline{\mathbf{w}}_{t+1} = \overline{\mathbf{u}}_t - \eta_t\mathbf{g}_t$, $\overline{\mathbf{v}}_{t+1} =$

$(1-\alpha^k)\overline{\mathbf{v}}_t + \alpha^k\overline{\mathbf{w}}_t - \delta^k\mathbf{g}_t$, and $\overline{\mathbf{u}}_{t+1} = \frac{\alpha^k}{1+\alpha^k}\overline{\mathbf{v}}_{t+1} + \frac{1}{1+\alpha^k}\overline{\mathbf{w}}_{t+1}$.

We first prove the theorem with $E = 2$ and $t-1$ being a communication round. We have

$$
\begin{aligned}
&\|\overline{\mathbf{v}}_{t+1} - \mathbf{w}^*\|^2_{\mathbf{H}-1} \\
&= \|(1-\alpha)\overline{\mathbf{v}}_t + \alpha\overline{\mathbf{u}}_t - \delta\sum_k p_k\tilde{\mathbf{H}}_t^k(\mathbf{u}_t^k - \mathbf{w}^*) - \mathbf{w}^*\|^2_{\mathbf{H}-1} \\
&= \|(1-\alpha)\overline{\mathbf{v}}_t + \alpha\overline{\mathbf{u}}_t - \mathbf{w}^*\|^2_{\mathbf{H}-1} + \delta^2\|\sum_k p_k\tilde{\mathbf{H}}_t^k(\mathbf{u}_t^k - \mathbf{w}^*)\|^2_{\mathbf{H}-1} \\
&- 2\delta\langle\sum_k p_k\tilde{\mathbf{H}}_t^k(\mathbf{u}_t^k - \mathbf{w}^*), (1-\alpha)\overline{\mathbf{v}}_t + \alpha\overline{\mathbf{u}}_t - \mathbf{w}^*\rangle_{\mathbf{H}-1} \\
&\leq \underbrace{\|(1-\alpha)\overline{\mathbf{v}}_t + \alpha\overline{\mathbf{u}}_t - \mathbf{w}^*\|^2_{\mathbf{H}-1}}_{A} + \underbrace{2\delta^2\|\sum_k p_k\tilde{\mathbf{H}}_t^k(\overline{\mathbf{u}}_t - \mathbf{w}^*)\|^2_{\mathbf{H}-1}}_{B} \\
&+ 2\delta^2\|\sum_k p_k\tilde{\mathbf{H}}_t^k(\overline{\mathbf{u}}_t - \mathbf{u}_t^k)\|^2_{\mathbf{H}-1} \underbrace{-2\delta\langle\sum_k p_k\tilde{\mathbf{H}}_t^k(\overline{\mathbf{u}}_t - \mathbf{w}^*), (1-\alpha)\overline{\mathbf{v}}_t + \alpha\overline{\mathbf{u}}_t - \mathbf{w}^*\rangle_{\mathbf{H}-1}}_{C} \\
&- 2\delta\langle\sum_k p_k\tilde{\mathbf{H}}_t^k(\mathbf{u}_t^k - \overline{\mathbf{u}}_t), (1-\alpha)\overline{\mathbf{v}}_t + \alpha\overline{\mathbf{u}}_t - \mathbf{w}^*\rangle_{\mathbf{H}-1}
\end{aligned}
$$

Following the proof in [119],

$$
\begin{aligned}
\mathbb{E}A &\leq \mathbb{E}(1-\alpha)\|\overline{\mathbf{v}}_t - \mathbf{w}^*\|^2_{\mathbf{H}-1} + \alpha\|\overline{\mathbf{u}}_t - \mathbf{w}^*\|^2_{\mathbf{H}-1} \\
&\leq \mathbb{E}(1-\alpha)\|\overline{\mathbf{v}}_t - \mathbf{w}^*\|^2_{\mathbf{H}-1} + \frac{\alpha}{\mu}\|\overline{\mathbf{u}}_t - \mathbf{w}^*\|^2
\end{aligned}
$$

using the convexity of the norm $\|\cdot\|_{\mathbf{H}-1}$ and that μ is the smallest non-zero eigenvalue of H.

Now

$$
\mathbb{E}B \leq 2\delta^2(\nu_{\max}\frac{1}{N}\tilde{\kappa} + \frac{N-\nu_{\min}}{N})\|(\overline{\mathbf{u}}_t - \mathbf{w}^*)\|^2_H
$$

using the folowing bound:

$$
\begin{aligned}
&\mathbb{E}\left(\sum_k p_k\tilde{\mathbf{H}}_t^k\right)\mathbf{H}^{-1}\left(\sum_k p_k\tilde{\mathbf{H}}_t^k\right)\\
&=\mathbb{E}\sum_k p_k^2\tilde{\mathbf{H}}_t^k\mathbf{H}^{-1}\tilde{\mathbf{H}}_t^k+\sum_{k\neq j}p_kp_j\tilde{\mathbf{H}}_t^k\mathbf{H}^{-1}\tilde{\mathbf{H}}_t^j\\
&\preceq \nu_{\max}\frac{1}{N}\mathbb{E}\sum_k p_k\tilde{\mathbf{H}}_t^k\mathbf{H}^{-1}\tilde{\mathbf{H}}_t^k+\sum_{k\neq j}p_kp_j\mathbf{H}^k\mathbf{H}^{-1}\mathbf{H}^j\\
&=\nu_{\max}\frac{1}{N}\mathbb{E}\sum_k p_k\tilde{\mathbf{H}}_t^k\mathbf{H}^{-1}\tilde{\mathbf{H}}_t^k+\sum_{k,j}p_kp_j\mathbf{H}^k\mathbf{H}^{-1}\mathbf{H}^j-\sum_k p_k^2\mathbf{H}^k\mathbf{H}^{-1}\mathbf{H}^k\\
&\preceq \nu_{\max}\frac{1}{N}\mathbb{E}\sum_k p_k\tilde{\mathbf{H}}_t^k\mathbf{H}^{-1}\tilde{\mathbf{H}}_t^k+\mathbf{H}-\frac{1}{N}\nu_{\min}\sum_k p_k\mathbf{H}^k\mathbf{H}^{-1}\mathbf{H}^k\\
&\preceq \nu_{\max}\frac{1}{N}\mathbb{E}\sum_k p_k\tilde{\mathbf{H}}_t^k\mathbf{H}^{-1}\tilde{\mathbf{H}}_t^k+\mathbf{H}-\frac{1}{N}\nu_{\min}(\sum_k p_k\mathbf{H}^k)\mathbf{H}^{-1}(\sum_k p_k\mathbf{H}^k)\\
&=\nu_{\max}\frac{1}{N}\mathbb{E}\sum_k p_k\tilde{\mathbf{H}}_t^k\mathbf{H}^{-1}\tilde{\mathbf{H}}_t^k+\frac{N-\nu_{\min}}{N}\mathbf{H}\\
&\preceq \nu_{\max}\frac{1}{N}\tilde{\kappa}\mathbf{H}+\frac{N-\nu_{\min}}{N}\mathbf{H}
\end{aligned}
$$

where we have used $\mathbb{E}\sum_k p_k\tilde{\mathbf{H}}_t^k\mathbf{H}^{-1}\tilde{\mathbf{H}}_t^k\leq\tilde{\kappa}\mathbf{H}$ by definition of $\tilde{\kappa}$ and the operator convexity of the mapping $W\to W\mathbf{H}^{-1}W$.

Finally,

$$
\begin{aligned}
\mathbb{E}C&=-\mathbb{E}2\delta\langle\sum_k p_k\tilde{\mathbf{H}}_t^k(\overline{\mathbf{u}}_t-\mathbf{w}^*),(1-\alpha)\overline{\mathbf{v}}_t+\alpha\overline{\mathbf{u}}_t-\mathbf{w}^*\rangle_{\mathbf{H}^{-1}}\\
&=-2\delta\langle\sum_k p_k\mathbf{H}^k(\overline{\mathbf{u}}_t-\mathbf{w}^*),(1-\alpha)\overline{\mathbf{v}}_t+\alpha\overline{\mathbf{u}}_t-\mathbf{w}^*\rangle_{\mathbf{H}^{-1}}\\
&=-2\delta\langle(\overline{\mathbf{u}}_t-\mathbf{w}^*),(1-\alpha)\overline{\mathbf{v}}_t+\alpha\overline{\mathbf{u}}_t-\mathbf{w}^*\rangle\\
&=-2\delta\langle(\overline{\mathbf{u}}_t-\mathbf{w}^*),\overline{\mathbf{u}}_t-\mathbf{w}^*+\frac{1-\alpha}{\alpha}(\overline{\mathbf{u}}_t-\overline{\mathbf{w}}_t)\rangle\\
&=-2\delta\|\overline{\mathbf{u}}_t-\mathbf{w}^*\|^2+\frac{1-\alpha}{\alpha}\delta(\|\overline{\mathbf{w}}_t-\mathbf{w}^*\|^2-\|\overline{\mathbf{u}}_t-\mathbf{w}^*\|^2-\|\overline{\mathbf{w}}_t-\overline{\mathbf{u}}_t\|^2)
\end{aligned}
$$

$$\leq \frac{1-\alpha}{\alpha}\delta\|\overline{\mathbf{w}}_t - \mathbf{w}^*\|^2 - \frac{1-\alpha}{\alpha}\delta\|\overline{\mathbf{u}}_t - \mathbf{w}^*\|^2$$

where we have used

$$\begin{aligned}
&(1-\alpha)\overline{\mathbf{v}}_t + \alpha\overline{\mathbf{u}}_t \\
&= (1-\alpha)((1+\alpha)\overline{\mathbf{u}}_t - \overline{\mathbf{w}}_t)/\alpha + \alpha\overline{\mathbf{u}}_t \\
&= \frac{1}{\alpha}\overline{\mathbf{u}}_t - \frac{1-\alpha}{\alpha}\overline{\mathbf{w}}_t
\end{aligned}$$

and the identity that $-2\langle \mathbf{a}, \mathbf{b}\rangle = \|\mathbf{a}\|^2 + \|\mathbf{b}\|^2 - \|\mathbf{a}+\mathbf{b}\|^2$.

It follows that

$$\begin{aligned}
&\mathbb{E}\|\overline{\mathbf{v}}_{t+1} - \mathbf{w}^*\|^2_{\mathbf{H}^{-1}} \\
&\leq (1-\alpha)\|\overline{\mathbf{v}}_t - \mathbf{w}^*\|^2_{\mathbf{H}^{-1}} + \frac{1-\alpha}{\alpha}\delta\|\overline{\mathbf{w}}_t - \mathbf{w}^*\|^2 \\
&+ (\frac{\alpha}{\mu} - \frac{1-\alpha}{\alpha}\delta)\|\overline{\mathbf{u}}_t - \mathbf{w}^*\|^2 + 2\delta^2(\nu_{\max}\frac{1}{N}\tilde{\kappa} + \frac{N-\nu_{\min}}{N})\|(\overline{\mathbf{u}}_t - \mathbf{w}^*)\|^2_H \\
&+ 2\delta^2\|\sum_k p_k\tilde{\mathbf{H}}^k_t(\overline{\mathbf{u}}_t - \mathbf{u}^k_t)\|^2_{\mathbf{H}^{-1}} \\
&- 2\delta\langle\sum_k p_k\tilde{\mathbf{H}}^k_t(\mathbf{u}^k_t - \overline{\mathbf{u}}_t), (1-\alpha)\overline{\mathbf{v}}_t + \alpha\overline{\mathbf{u}}_t - \mathbf{w}^*\rangle_{\mathbf{H}^{-1}}
\end{aligned}$$

On the other hand,

$$\begin{aligned}
\mathbb{E}\|\overline{\mathbf{w}}_{t+1} - \mathbf{w}^*\|^2 &= \mathbb{E}\|\overline{\mathbf{u}}_t - \mathbf{w}^* - \eta\sum_k p_k\tilde{\mathbf{H}}^k_t(\overline{\mathbf{u}}_t - \mathbf{w}^*)\|^2 \\
&= \mathbb{E}\|\overline{\mathbf{u}}_t - \mathbf{w}^*\|^2 - 2\eta\|\overline{\mathbf{u}}_t - \mathbf{w}^*\|^2_H + \eta^2\|\sum_k p_k\tilde{\mathbf{H}}^k_t(\overline{\mathbf{u}}_t - \mathbf{w}^*)\|^2 \\
&\leq \mathbb{E}\|\overline{\mathbf{u}}_t - \mathbf{w}^*\|^2 - 2\eta\|\overline{\mathbf{u}}_t - \mathbf{w}^*\|^2_H + \eta^2(\nu_{\max}\frac{1}{N}\ell + L\frac{N-\nu_{\min}}{N})\|\overline{\mathbf{u}}_t - \mathbf{w}^*\|^2
\end{aligned}$$

where we use the following bound:

$$
\begin{aligned}
&\mathbb{E}\left(\sum_k p_k \tilde{\mathbf{H}}_t^k\right)\left(\sum_k p_k \tilde{\mathbf{H}}_t^k\right) \\
&= \mathbb{E}\sum_k p_k^2 \tilde{\mathbf{H}}_t^k \tilde{\mathbf{H}}_t^k + \sum_{k\neq j} p_k p_j \tilde{\mathbf{H}}_t^k \tilde{\mathbf{H}}_t^j \\
&\preceq \nu_{\max}\frac{1}{N}\mathbb{E}\sum_k p_k \tilde{\mathbf{H}}_t^k \tilde{\mathbf{H}}_t^k + \sum_{k\neq j} p_k p_j \mathbf{H}^k \mathbf{H}^j \\
&= \nu_{\max}\frac{1}{N}\mathbb{E}\sum_k p_k \tilde{\mathbf{H}}_t^k \tilde{\mathbf{H}}_t^k + \sum_{k,j} p_k p_j \mathbf{H}^k \mathbf{H}^j - \sum_k p_k^2 \mathbf{H}^k \mathbf{H}^k \\
&\preceq \nu_{\max}\frac{1}{N}\mathbb{E}\sum_k p_k \tilde{\mathbf{H}}_t^k \tilde{\mathbf{H}}_t^k + \mathbf{H}^2 - \frac{1}{N}\nu_{\min}\sum_k p_k \mathbf{H}^k \mathbf{H}^k \\
&\preceq \nu_{\max}\frac{1}{N}\mathbb{E}\sum_k p_k \tilde{\mathbf{H}}_t^k \tilde{\mathbf{H}}_t^k + \mathbf{H}^2 - \frac{1}{N}\nu_{\min}(\sum_k p_k \mathbf{H}^k)(\sum_k p_k \mathbf{H}^k) \\
&= \nu_{\max}\frac{1}{N}\mathbb{E}\sum_k p_k \tilde{\mathbf{H}}_t^k \tilde{\mathbf{H}}_t^k + \frac{N-\nu_{\min}}{N}\mathbf{H}^2 \\
&\preceq \nu_{\max}\frac{1}{N} l\mathbf{H} + L\frac{N-\nu_{\min}}{N}\mathbf{H}
\end{aligned}
$$

again using that $W \to W^2$ is operator convex and that $\mathbb{E}\tilde{\mathbf{H}}_t^k \tilde{\mathbf{H}}_t^k \preceq l\mathbf{H}^k$ by definition of l.

Combining the bounds for $\mathbb{E}\|\overline{\mathbf{w}}_{t+1} - \mathbf{w}^*\|^2$ and $\mathbb{E}\|\overline{\mathbf{v}}_{t+1} - \mathbf{w}^*\|^2_{\mathbf{H}^{-1}}$,

$$
\begin{aligned}
&\mathbb{E}\frac{\delta}{\alpha}\|\overline{\mathbf{w}}_{t+1} - \mathbf{w}^*\|^2 + \|\overline{\mathbf{v}}_{t+1} - \mathbf{w}^*\|^2_{\mathbf{H}^{-1}} \\
&\leq (1-\alpha)\|\overline{\mathbf{v}}_t - \mathbf{w}^*\|^2_{\mathbf{H}^{-1}} + \frac{1-\alpha}{\alpha}\delta\|\overline{\mathbf{w}}_t - \mathbf{w}^*\|^2 + (\frac{\alpha}{\mu} - \delta)\|\overline{\mathbf{u}}_t - \mathbf{w}^*\|^2 \\
&+ (2\delta^2(\nu_{\max}\frac{1}{N}\tilde{\kappa} + \frac{N-\nu_{\min}}{N}) - 2\eta\delta/\alpha + \eta^2\delta(\nu_{\max}\frac{1}{N}l + L\frac{N-\nu_{\min}}{N})/\alpha)\|\overline{\mathbf{u}}_t - \mathbf{w}^*\|^2 \\
&+ 2\delta^2\|\sum_k p_k \tilde{\mathbf{H}}_t^k(\overline{\mathbf{u}}_t - \mathbf{u}_t^k)\|^2_{\mathbf{H}^{-1}} \\
&+ \delta L\sum_k p_k\|(\overline{\mathbf{u}}_t - \mathbf{u}_t^k)\|^2_{\mathbf{H}^{-1}}
\end{aligned}
$$

Following [119] if we choose step sizes so that

$$\frac{\alpha}{\mu} - \delta \leq 0$$
$$2\delta^2(\nu_{\max}\frac{1}{N}\tilde{\kappa} + \frac{N-\nu_{\min}}{N}) - 2\eta\delta/\alpha + \eta^2\delta(\nu_{\max}\frac{1}{N}l + L\frac{N-\nu_{\min}}{N})/\alpha \leq 0$$

or equivalently

$$\alpha/\delta \leq \mu$$
$$2\alpha\delta(\nu_{\max}\frac{1}{N}\tilde{\kappa} + \frac{N-\nu_{\min}}{N}) + \eta(\eta(\nu_{\max}\frac{1}{N}l + L\frac{N-\nu_{\min}}{N}) - 2) \leq 0$$

the second and third terms are negative. To optimize the step sizes, note that the two inequalities imply

$$\alpha^2 \leq \eta(2 - \eta(\nu_{\max}\frac{1}{N}l + L\frac{N-\nu_{\min}}{N}))\mu/2(\nu_{\max}\frac{1}{N}\tilde{\kappa} + \frac{N-\nu_{\min}}{N})$$

and maximizing the right hand side with respect to η, which is quadratic, we see that $\eta \equiv 1/(\nu_{\max}\frac{1}{N}l + L\frac{N-\nu_{\min}}{N})$ maximizes the right hand side, with

$$\alpha \equiv \frac{1}{\sqrt{2(\nu_{\max}\frac{1}{N}\kappa_1 + \kappa\frac{N-\nu_{\min}}{N})(\nu_{\max}\frac{1}{N}\tilde{\kappa} + \frac{N-\nu_{\min}}{N})}}$$
$$\delta \equiv \frac{\alpha}{\mu} = \frac{\eta}{\alpha(\nu_{\max}\frac{1}{N}\tilde{\kappa} + \frac{N-\nu_{\min}}{N})}$$

Note that $\alpha = \frac{1}{\sqrt{2(\nu_{\max}\frac{1}{N}\kappa_1+\kappa\frac{N-\nu_{\min}}{N})(\nu_{\max}\frac{1}{N}\tilde{\kappa}+\frac{N-\nu_{\min}}{N})}} = O(\frac{N}{\sqrt{\kappa_1\tilde{\kappa}}})$ when $N = O(\min\{\tilde{\kappa}, \kappa_1/\kappa\})$.

Finally, to deal with the terms $2\delta^2\|\sum_k p_k\tilde{\mathbf{H}}_t^k(\overline{\mathbf{u}}_t - \mathbf{u}_t^k)\|^2_{\mathbf{H}^{-1}} + \delta L\sum_k p_k\|(\overline{\mathbf{u}}_t - \mathbf{u}_t^k)\|^2_{\mathbf{H}^{-1}}$, we

can use Jensen

$$
\begin{aligned}
&2\delta^2\|\sum_k p_k\tilde{\mathbf{H}}_t^k(\overline{\mathbf{u}}_t-\mathbf{u}_t^k)\|_{\mathbf{H}^{-1}}^2+\delta L\sum_k p_k\|(\overline{\mathbf{u}}_t-\mathbf{u}_t^k)\|_{\mathbf{H}^{-1}}^2\\
&\leq(2\delta^2l^2+\delta L)\sum_k p_k\|\overline{\mathbf{u}}_t-\mathbf{u}_t^k\|_{\mathbf{H}^{-1}}^2\\
&=(2\delta^2l^2+\delta L)\sum_k p_k\|\frac{\alpha}{1+\alpha}\overline{\mathbf{v}}_t+\frac{1}{1+\alpha}\overline{\mathbf{w}}_t-(\frac{\alpha}{1+\alpha}v_t^k+\frac{1}{1+\alpha}w_t^k)\|_{\mathbf{H}^{-1}}^2\\
&\leq(2\delta^2l^2+\delta L)(2(\frac{\alpha}{1+\alpha})^2\delta^2+2(\frac{1}{1+\alpha})^2\eta^2)\sum_k p_k\|\tilde{\mathbf{H}}_{t-1}^k(\overline{\mathbf{u}}_{t-1}-\mathbf{w}^*)\|^2\\
&\leq(2\delta^2l^2+\delta L)(2(\frac{\alpha}{1+\alpha})^2\delta^2+2(\frac{1}{1+\alpha})^2\eta^2)l^2\|(\overline{\mathbf{u}}_{t-1}-\mathbf{w}^*)\|^2
\end{aligned}
$$

which can be combined with the terms with $\|(\overline{\mathbf{u}}_{t-1}-\mathbf{w}^*)\|^2$ in the recursive expansion of $\mathbb{E}\frac{\delta}{\alpha}\|\overline{\mathbf{w}}_t-\mathbf{w}^*\|^2+\|\overline{\mathbf{v}}_t-\mathbf{w}^*\|_{\mathbf{H}^{-1}}^2$:

$$
\begin{aligned}
&\mathbb{E}\frac{\delta}{\alpha}\|\overline{\mathbf{w}}_t-\mathbf{w}^*\|^2+\|\overline{\mathbf{v}}_t-\mathbf{w}^*\|_{\mathbf{H}^{-1}}^2\\
&\leq(1-\alpha)\|\overline{\mathbf{v}}_{t-1}-\mathbf{w}^*\|_{\mathbf{H}^{-1}}^2+\frac{1-\alpha}{\alpha}\delta\|\overline{\mathbf{w}}_{t-1}-\mathbf{w}^*\|^2+(\frac{\alpha}{\mu}-\delta)\|\overline{\mathbf{u}}_{t-1}-\mathbf{w}^*\|^2\\
&+(2\delta^2(\nu_{\max}\frac{1}{N}\tilde{\kappa}+\frac{N-\nu_{\min}}{N})-2\eta\delta/\alpha+\eta^2\delta(\nu_{\max}\frac{1}{N}l+L\frac{N-\nu_{\min}}{N})/\alpha)\|\overline{\mathbf{u}}_{t-1}-\mathbf{w}^*\|^2
\end{aligned}
$$

and the step sizes can be chosen so that the resulting coefficients are negative. Therefore, we have shown that

$$
\mathbb{E}\|\overline{\mathbf{w}}_{t+1}-\mathbf{w}^*\|^2\leq(1-\alpha)^2\|\overline{\mathbf{w}}_{t-1}-\mathbf{w}^*\|^2
$$

where $\alpha=\frac{1}{\sqrt{2(\nu_{\max}\frac{1}{N}\kappa_1+\kappa\frac{N-\nu_{\min}}{N})(\nu_{\max}\frac{1}{N}\tilde{\kappa}+\frac{N-\nu_{\min}}{N})}}=O(\frac{N}{\nu_{\max}\sqrt{\kappa_1\tilde{\kappa}}+N-\nu_{\min}})$ when $N=O(\min\{\tilde{\kappa},\kappa_1/\kappa\})$.

For general $E > 1$, choosing $\eta = c/E(\nu_{\max}\frac{1}{N}l + L\frac{N-\nu_{\min}}{N})$ for some small constant c results in $\alpha = O(\frac{1}{E\sqrt{(\nu_{\max}\frac{1}{N}\kappa_1+\kappa\frac{N-\nu_{\min}}{N})(\nu_{\max}\frac{1}{N}\tilde{\kappa}+\frac{N-\nu_{\min}}{N})}})$ and this guarantees that

$$\mathbb{E}\|\overline{\mathbf{w}}_t - \mathbf{w}^*\|^2 \leq (1-\alpha)^t\|\mathbf{w}_0 - \mathbf{w}^*\|^2$$

for all t.

□

Details on Experiments and Additional Results

We described the precise procedure to reproduce the results in this chapter. As we mentioned in Section 6.6, we empirically verified the linear speed up on various convex settings for both FedAvg and its accelerated variants. For all the results, we set random seeds as $0, 1, 2$ and report the best convergence rate across the three folds. For each run, we initialize $\mathbf{w}_0 = \mathbf{0}$ and measure the number of iteration to reach the target accuracy ϵ. We use the small-scale dataset w8a [155], which consists of $n = 49749$ samples with feature dimension $d = 300$. The label is either positive one or negative one. The dataset has sparse binary features in $\{0, 1\}$. Each sample has 11.15 non-zero feature values out of 300 features on average. We set the batch size equal to four across all experiments. In the next following subsections, we introduce parameter searching in each objective separately.

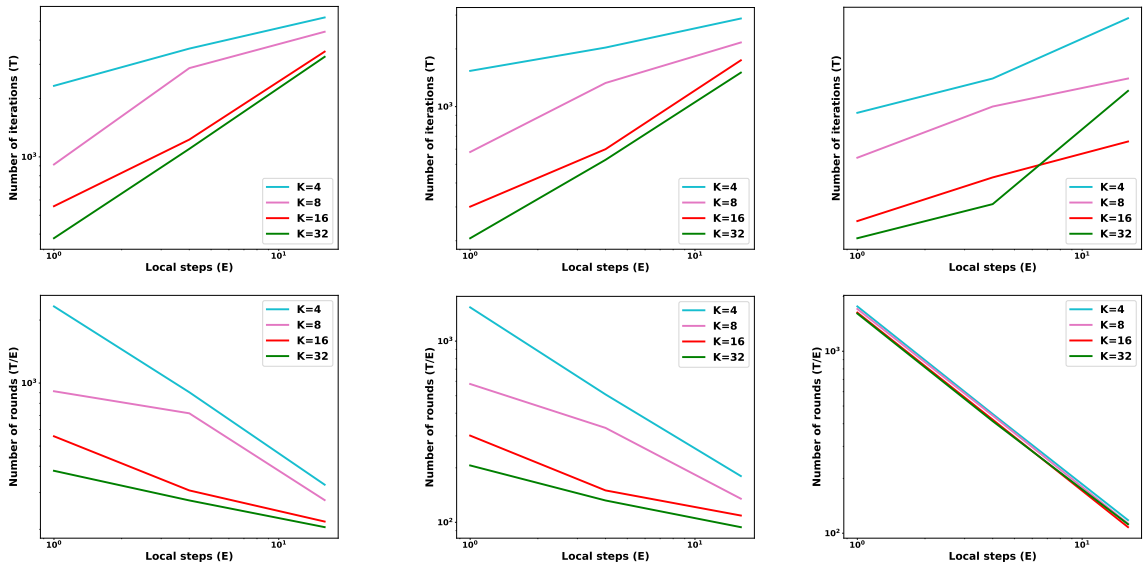
Strongly Convex Objectives We first consider the strongly convex objective function, where we use a regularized binary logistic regression with regularization $\lambda = 1/n \approx 2e-5$. We evenly distributed on $1, 2, 4, 8, 16, 32$ devices and report the number of iterations/rounds needed to converge to $\epsilon-$accuracy, where $\epsilon = 0.005$. The optimal objective function value f^*

is set as $f^* = 0.126433176216545$. This is determined numerically and we follow the setting in [178]. The learning rate is decayed as the $\eta_t = \min(\eta_0, \frac{nc}{1+t})$, where we extensively search the best learning rate $c \in \{2^{-1}c_0, 2^{-2}c_0, c_0, 2c_0, 2^2c_0\}$. In this case, we search the initial learning rate $\eta_0 \in \{1, 32\}$ and $c_0 = 1/8$.

Convex Smooth Objectives We also use binary logistic regression without regularization. The setting is almost same as its regularized counter part. We also evenly distributed all the samples on $1, 2, 4, 8, 16, 32$ devices. The figure shows the number of iterations needed to converge to $\epsilon-$accuracy, where $\epsilon = 0.02$. The optiaml objective function value is set as $f^* = 0.11379089057514849$, determined numerically. The learning rate is decayed as the $\eta_t = \min(\eta_0, \frac{nc}{1+t})$, where we extensively search the best learning rate $c \in \{2^{-1}c_0, 2^{-2}c_0, c_0, 2c_0, 2^2c_0\}$. In this case, we search the initial learning rate $\eta_0 \in \{1, 32\}$ and $c_0 = 1/8$.

Linear regression For linear regression, we use the same feature vectors from w8a dataset and generate ground truth $[\mathbf{w}^*, b^*]$ from a multivariate normal distribution with zero mean and standard deviation one. Then we generate label based on $y_i = \mathbf{x}_i^t\mathbf{w}^* + b^*$. This procedure will ensure we satisfy the over-parameterized setting as required in our theorems. We also evenly distributed all the samples on $1, 2, 4, 8, 16, 32$ devices. The figure shows the number of iterations needed to converge to $\epsilon-$accuracy, where $\epsilon = 0.02$. The optiaml objective function value is $f^* = 0$. The learning rate is decayed as the $\eta_t = \min(\eta_0, \frac{nc}{1+t})$, where we extensively search the best learning rate $c \in \{2^{-1}c_0, 2^{-2}c_0, c_0, 2c_0, 2^2c_0\}$. In this case, we search the initial learning rate $\eta_0 \in \{0.1, 0.12\}$ and $c_0 = 1/256$.

Partial Participation To examine the linear speedup of FedAvg in partial participation setting, we evenly distributed data on $4, 8, 16, 32, 64, 128$ devices and uniformly sample 50% devices without replacement. All other hyperparameters are the same as previous sections.



(a) Strongly convex objective (b) Convex smooth objective (c) Linear regression

Figure B.1: The convergence of FedAvg w.r.t the number of local steps E.

Nesterov accelerated FedAvg The experiments of Nesterov accelerated FedAvg (see update formula below) uses the same setting as previous three sections for vanilia FedAvg.

$$\mathbf{y}_{t+1}^k = \mathbf{w}_t^k - \alpha_t \mathbf{g}_{t,k}$$

$$\mathbf{w}_{t+1}^k = \begin{cases} \mathbf{y}_{t+1}^k + \beta_t(\mathbf{y}_{t+1}^k - \mathbf{y}_t^k) & \text{if } t+1 \notin \mathcal{I}_E \\ \sum_{k \in \mathcal{S}_{t+1}} \left(\mathbf{y}_{t+1}^k + \beta_t(\mathbf{y}_{t+1}^k - \mathbf{y}_t^k) \right) & \text{if } t+1 \in \mathcal{I}_E \end{cases}$$

We set $\beta_t = 0.1$ and search α_t in the same way as η_t in FedAvg.

The impact of E. In this subsection, we further examine how does the number of local steps (E) affect convergence. As shown in Figure B.1, the number of iterations increases as E increase, which slow down the convergence in terms of gradient computation. However, it can save communication costs as the number of rounds decreased when the E increases. This showcase that we need a proper choice of E to trade-off the communication cost and convergence speed.

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] Abbas Abdolmaleki, Jost Tobias Springenberg, Yuval Tassa, Remi Munos, Nicolas Heess, and Martin Riedmiller. Maximum a posteriori policy optimisation. *arXiv preprint arXiv:1806.06920*, 2018.

[2] Naoki Abe, Prem Melville, Cezar Pendus, Chandan K Reddy, David L Jensen, Vince P Thomas, James J Bennett, Gary F Anderson, Brent R Cooley, Melissa Kowalczyk, et al. Optimizing debt collections using constrained reinforcement learning. In *SIGKDD*, pages 75–84. ACM, 2010.

[3] Naoki Abe, Naval Verma, Chid Apte, and Robert Schroko. Cross channel optimized marketing by reinforcement learning. In *SIGKDD*, pages 767–772. ACM, 2004.

[4] YuXuan Liu Pieter Abbeel†‡ Sergey Levine Abhishek Gupta†, Coline Devin†. Learning invariant feature spaces to transfer skills with reinforcement learning. In *Under review as a conference paper at ICLR 2017*, 2017.

[5] Zeyuan Allen-Zhu, Yuanzhi Li, and Zhao Song. A convergence theory for deep learning via over-parameterization. *arXiv preprint arXiv:1811.03962*, 2018.

[6] Saleema Amershi, Maya Cakmak, W Bradley Knox, and Todd Kulesza. Power to the people: The role of humans in interactive machine learning. AAAI, 2014.

[7] Saleema Amershi, James Fogarty, and Daniel Weld. Regroup: Interactive machine learning for on-demand group creation in social networks. In *Proceedings of the SIGCHI*, pages 21–30. ACM, 2012.

[8] Saleema Amershi, Bongshin Lee, Ashish Kapoor, Ratul Mahajan, and Blaine Christian. Cuet: human-guided fast and accurate network alarm triage. In *Proceedings of the SIGCHI*, pages 157–166. ACM, 2011.

[9] Mihael Ankerst, Christian Elsen, Martin Ester, and Hans-Peter Kriegel. Visual classification: an interactive approach to decision tree construction. In *SIGKDD*, pages 392–396. ACM, 1999.

[10] A. Argyriou, T. Evgeniou, and M. Pontil. Convex multi-task feature learning. *Machine Learning*, 73(3):243–272, 2008.

[11] Kai Arulkumaran, Marc Peter Deisenroth, Miles Brundage, and Anil Anthony Bharath. A brief survey of deep reinforcement learning. *arXiv preprint arXiv:1708.05866*, 2017.

[12] Dzmitry Bahdanau, Philemon Brakel, Kelvin Xu, Anirudh Goyal, Ryan Lowe, Joelle Pineau, Aaron Courville, and Yoshua Bengio. An actor-critic algorithm for sequence prediction. *arXiv preprint arXiv:1607.07086*, 2016.

[13] Bart Bakker and Tom Heskes. Task clustering and gating for bayesian multitask learning. *The Journal of Machine Learning Research*, 4:83–99, 2003.

[14] Bram Bakker, Shimon Whiteson, Leon Kester, and Frans CA Groen. Traffic light control by multiagent reinforcement learning systems. In *Interactive Collaborative Information Systems*, pages 475–510. Springer, 2010.

[15] Peter L Bartlett and Shahar Mendelson. Rademacher and gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3(Nov):463–482, 2002.

[16] Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM journal on imaging sciences*, 2(1):183–202, 2009.

[17] Marc G Bellemare, Will Dabney, and Rémi Munos. A distributional perspective on reinforcement learning. *arXiv preprint arXiv:1707.06887*, 2017.

[18] Marc G Bellemare, Yavar Naddaf, Joel Veness, and Michael Bowling. The arcade learning environment: An evaluation platform for general agents. *J. Artif. Intell. Res.(JAIR)*, 47:253–279, 2013.

[19] Shai Ben-David and Reba Schuller. Exploiting task relatedness for multiple task learning. In *Learning Theory and Kernel Machines*, pages 567–580. Springer, 2003.

[20] Jacob Bien, Jonathan Taylor, and Robert Tibshirani. A lasso for hierarchical interactions. *Annals of statistics*, 41(3):1111, 2013.

[21] Christopher M Bishop. *Pattern recognition and machine learning*. springer, 2006.

[22] Edwin V Bonilla, Kian M Chai, and Christopher Williams. Multi-task gaussian process prediction. In *NIPS*, pages 153–160, 2007.

[23] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning*, 3(1):1–122, 2011.

[24] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym. *arXiv preprint arXiv:1606.01540*, 2016.

[25] Cristian Bucilu, Rich Caruana, and Alexandru Niculescu-Mizil. Model compression. In *SIGKDD*, pages 535–541. ACM, 2006.

[26] Chris Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Greg Hullender. Learning to rank using gradient descent. In *Proceedings of the 22nd international conference on Machine learning*, pages 89–96. ACM, 2005.

[27] Lucian Busoniu, Robert Babuska, and Bart De Schutter. A comprehensive survey of multiagent reinforcement learning. *IEEE Trans. Systems, Man, and Cybernetics, Part C*, 38(2):156–172, 2008.

[28] Lucian Buşoniu, Robert Babuška, and Bart De Schutter. Multi-agent reinforcement learning: An overview. In *Innovations in multi-agent systems and applications-1*, pages 183–221. Springer, 2010.

[29] Remi J Cadoret, William R Yates, George Woodworth, and Mark A Stewart. Genetic-environmental interaction in the genesis of aggressivity and conduct disorders. *Archives of General Psychiatry*, 52(11):916–924, 1995.

[30] Alfredo Canziani, Adam Paszke, and Eugenio Culurciello. An analysis of deep neural network models for practical applications. *arXiv preprint arXiv:1605.07678*, 2016.

[31] Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li. Learning to rank: from pairwise approach to listwise approach. In *ICML*, pages 129–136. ACM, 2007.

[32] Marc Carreras, Junku Yuh, Joan Batlle, and Pere Ridao. A behavior-based scheme using reinforcement learning for autonomous underwater vehicles. *IEEE Journal of Oceanic Engineering*, 30(2):416–427, 2005.

[33] Rich Caruana. Multitask learning. *Machine learning*, 28(1):41–75, 1997.

[34] Pablo Samuel Castro, Subhodeep Moitra, Carles Gelada, Saurabh Kumar, and Marc G. Bellemare. Dopamine: A research framework for deep reinforcement learning. *CoRR*, abs/1812.06110, 2018.

[35] Shiyu Chang, Guo-Jun Qi, Charu C Aggarwal, Jiayu Zhou, Meng Wang, and Thomas S Huang. Factorized similarity learning in networks. In *ICDM*, pages 60–69. IEEE, 2014.

[36] Fei Chen, Zhenhua Dong, Zhenguo Li, and Xiuqiang He. Federated meta-learning for recommendation. *arXiv preprint arXiv:1802.07876*, 2018.

[37] Jianhui Chen, Jiayu Zhou, and Jieping Ye. Integrating low-rank and group-sparse structures for robust multi-task learning. In *SIGKDD*, pages 42–50. ACM, 2011.

[38] Xiaohui Chen, Xinghua Shi, Xing Xu, Zhiyong Wang, Ryan Mills, Charles Lee, and Jinbo Xu. A two-graph guided multi-task lasso approach for eqtl mapping. In *AISTATS*, pages 208–217, 2012.

[39] Nam Hee Choi, William Li, and Ji Zhu. Variable selection with the strong heredity constraint and its oracle property. *JASA*, 105(489):354–364, 2010.

[40] Didi Chuxing.

[41] Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. Fast and accurate deep network learning by exponential linear units (elus). *arXiv preprint arXiv:1511.07289*, 2015.

[42] Will Dabney, Georg Ostrovski, David Silver, and Rémi Munos. Implicit quantile networks for distributional reinforcement learning. *arXiv preprint arXiv:1806.06923*, 2018.

[43] Bo Dai, Albert Shaw, Lihong Li, Lin Xiao, Niao He, Zhen Liu, Jianshu Chen, and Le Song. Sbeed: Convergent reinforcement learning with nonlinear function approximation. *arXiv preprint arXiv:1712.10285*, 2017.

[44] Hal Daumé, John Langford, and Daniel Marcu. Search-based structured prediction. *Machine learning*, 75(3):297–325, 2009.

[45] Peter Dayan and Geoffrey E Hinton. Using expectation-maximization for reinforcement learning. *Neural Computation*, 9(2):271–278, 1997.

[46] Thomas Degris, Martha White, and Richard S Sutton. Off-policy actor-critic. *arXiv preprint arXiv:1205.4839*, 2012.

[47] Pierre J Dejax and Teodor Gabriel Crainic. Survey paper—a review of empty flows and fleet management models in freight transportation. *Transportation science*, 21(4):227–248, 1987.

[48] Prafulla Dhariwal, Christopher Hesse, Oleg Klimov, Alex Nichol, Matthias Plappert, Alec Radford, John Schulman, Szymon Sidor, Yuhuai Wu, and Peter Zhokhov. Openai baselines. `https://github.com/openai/baselines`, 2017.

[49] Jilles Steeve Dibangoye and Olivier Buffet. *Learning to Act in Decentralized Partially Observable MDPs*. PhD thesis, INRIA Grenoble-Rhone-Alpes-CHROMA Team; INRIA Nancy, équipe LARSEN, 2018.

[50] Pierre Dillenbourg. *Collaborative Learning: Cognitive and Computational Approaches. Advances in Learning and Instruction Series*. ERIC, 1999.

[51] Pierre Dutilleul. The mle algorithm for the matrix normal distribution. *J STAT COMPUT SIM*, 64(2):105–123, 1999.

[52] Thalia C Eley, Karen Sugden, Alejandro Corsico, Alice M Gregory, Pak Sham, Peter McGuffin, Robert Plomin, and Ian W Craig. Gene–environment interaction analysis of serotonin system markers with adolescent depression. *Molecular psychiatry*, 9(10):908–915, 2004.

[53] A Evgeniou and Massimiliano Pontil. Multi-task feature learning. *NIPS*, 19:41, 2007.

[54] Theodoros Evgeniou, Charles A Micchelli, and Massimiliano Pontil. Learning multiple tasks with kernel methods. In *JMLR*, pages 615–637, 2005.

[55] Theodoros Evgeniou and Massimiliano Pontil. Regularized multi–task learning. In *SIGKDD*, pages 109–117. ACM, 2004.

[56] Benjamin Eysenbach and Sergey Levine. If maxent rl is the answer, what is the question? *arXiv preprint arXiv:1910.01913*, 2019.

[57] Alireza Fallah, Aryan Mokhtari, and Asuman Ozdaglar. Personalized federated learning: A meta-learning approach. *arXiv preprint arXiv:2002.07948*, 2020.

[58] Hongliang Fei and Jun Huan. Structured feature selection and task relationship inference for multi-task learning. *Knowledge and information systems*, 35(2):345–364, 2013.

[59] Chelsea Finn. *Learning to learn with gradients*. PhD thesis, UC Berkeley, 2018.

[60] Jakob Foerster, Gregory Farquhar, Triantafyllos Afouras, Nantas Nardelli, and Shimon Whiteson. Counterfactual multi-agent policy gradients. *arXiv preprint arXiv:1705.08926*, 2017.

[61] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. A note on the group lasso and a sparse group lasso. *arXiv preprint arXiv:1001.0736*, 2010.

[62] Silvia Gandy, Benjamin Recht, and Isao Yamada. Tensor completion and low-n-rank tensor recovery via convex optimization. *Inverse Problems*, 27(2):025010, 2011.

[63] JM Gatt, CB Nemeroff, C Dobson-Stone, RH Paul, RA Bryant, PR Schofield, E Gordon, AH Kemp, and LM Williams. Interactions between bdnf val66met polymorphism and early life stress predict brain and arousal pathways to syndromal depression and anxiety. *Molecular psychiatry*, 14(7):681–695, 2009.

[64] Gregory A Godfrey and Warren B Powell. An adaptive dynamic programming algorithm for dynamic fleet management, i: Single period travel times. *Transportation Science*, 36(1):21–39, 2002.

[65] Gregory A Godfrey and Warren B Powell. An adaptive dynamic programming algorithm for dynamic fleet management, ii: Multiperiod travel times. *Transportation Science*, 36(1):40–54, 2002.

[66] Pinghua Gong, Jieping Ye, and Changshui Zhang. Robust multi-task feature learning. In *SIGKDD*, pages 895–903. ACM, 2012.

[67] Pinghua Gong, Changshui Zhang, Zhaosong Lu, Jianhua Z Huang, and Jieping Ye. A general iterative shrinkage and thresholding algorithm for non-convex regularized optimization problems. In *ICML*, volume 28, page 37, 2013.

[68] Pinghua Gong, Jiayu Zhou, Wei Fan, and Jieping Ye. Efficient multi-task feature learning with calibration. In *SIGKDD*, pages 761–770. ACM, 2014.

[69] Michael Grant, Stephen Boyd, and Yinyu Ye. Cvx: Matlab software for disciplined convex programming, 2008.

[70] Nizar Grira, Michel Crucianu, and Nozha Boujemaa. Active semi-supervised fuzzy clustering for image database categorization. In *Proceedings of the 7th ACM SIGMM international workshop on Multimedia information retrieval*, pages 9–16. ACM, 2005.

[71] Audrunas Gruslys, Will Dabney, Mohammad Gheshlaghi Azar, Bilal Piot, Marc Bellemare, and Remi Munos. The reactor: A fast and sample-efficient actor-critic agent for reinforcement learning. 2018.

[72] Shixiang Gu, Timothy Lillicrap, Zoubin Ghahramani, Richard E Turner, and Sergey Levine. Q-prop: Sample-efficient policy gradient with an off-policy critic. *arXiv preprint arXiv:1611.02247*, 2016.

[73] Carlos Guestrin, Michail Lagoudakis, and Ronald Parr. Coordinated reinforcement learning. In *ICML*, volume 2, pages 227–234, 2002.

[74] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International Conference on Machine Learning*, pages 1856–1865, 2018.

[75] Farzin Haddadpour and Mehrdad Mahdavi. On the convergence of local descent methods in federated learning. *arXiv preprint arXiv:1910.14425*, 2019.

[76] Shih-Ping Han. A successive projection method. *Mathematical Programming*, 40(1-3):1–14, 1988.

[77] Andrew Hard, Chloé M Kiddon, Daniel Ramage, Francoise Beaufays, Hubert Eichner, Kanishka Rao, Rajiv Mathews, and Sean Augenstein. Federated learning for mobile keyboard prediction, 2018.

[78] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *ECCV*, pages 630–645. Springer, 2016.

[79] Kari Hemminki, Justo Lorenzo Bermejo, and Asta Försti. The balance between heritable and environmental aetiology of human disease. *Nature Reviews Genetics*, 7(12):958–965, 2006.

[80] Matteo Hessel, Joseph Modayil, Hado Van Hasselt, Tom Schaul, Georg Ostrovski, Will Dabney, Dan Horgan, Bilal Piot, Mohammad Azar, and David Silver. Rainbow: Combining improvements in deep reinforcement learning. *arXiv preprint arXiv:1710.02298*, 2017.

[81] Todd Hester, Matej Vecerik, Olivier Pietquin, Marc Lanctot, Tom Schaul, Bilal Piot, Dan Horgan, John Quan, Andrew Sendonaris, Ian Osband, et al. Deep q-learning from demonstrations. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[82] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[83] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989.

[84] Zhouyuan Huo, Qian Yang, Bin Gu, Lawrence Carin Huang, et al. Faster on-device training using new federated momentum algorithm. *arXiv preprint arXiv:2002.02090*, 2020.

[85] Andrew Ilyas, Logan Engstrom, Shibani Santurkar, Dimitris Tsipras, Firdaus Janoos, Larry Rudolph, and Aleksander Madry. Are deep policy gradient algorithms truly policy gradient algorithms? *arXiv preprint arXiv:1811.02553*, 2018.

[86] Max Jaderberg, Volodymyr Mnih, Wojciech Marian Czarnecki, Tom Schaul, Joel Z Leibo, David Silver, and Koray Kavukcuoglu. Reinforcement learning with unsupervised auxiliary tasks. *arXiv preprint arXiv:1611.05397*, 2016.

[87] Prateek Jain, Sham M Kakade, Rahul Kidambi, Praneeth Netrapalli, and Aaron Sidford. Accelerating stochastic gradient descent. In *Proc. STAT*, volume 1050, page 26, 2017.

[88] Ali Jalali, Sujay Sanghavi, Chao Ruan, and Pradeep K Ravikumar. A dirty model for multi-task learning. In *NIPS*.

[89] Shuiwang Ji and Jieping Ye. An accelerated gradient method for trace norm minimization. In *ICML*, pages 457–464. ACM, 2009.

[90] Nan Jiang and Alekh Agarwal. Open problem: The dependence of sample complexity lower bounds on planning horizon. In *Conference On Learning Theory*, pages 3395–3398, 2018.

[91] Nan Jiang, Akshay Krishnamurthy, Alekh Agarwal, John Langford, and Robert E Schapire. Contextual decision processes with low bellman rank are pac-learnable. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1704–1713. JMLR. org, 2017.

[92] Peng Jiang and Gagan Agrawal. A linear speedup analysis of distributed deep learning with sparse and quantized communication. In *Advances in Neural Information Processing Systems*, pages 2525–2536, 2018.

[93] Jeff Kahn, Nathan Linial, and Alex Samorodnitsky. Inclusion-exclusion: Exact and approximate. *Combinatorica*, 16(4):465–477, 1996.

[94] Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Keith Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. Advances and open problems in federated learning. *arXiv preprint arXiv:1912.04977*, 2019.

[95] Sham Machandranath Kakade et al. *On the sample complexity of reinforcement learning*. PhD thesis, University of London London, England, 2003.

[96] Tsuyoshi Kato, Hisashi Kashima, Masashi Sugiyama, and Kiyoshi Asai. Multi-task learning via conic programming. In *NIPS*, pages 737–744, 2008.

[97] Michael J Kearns, Yishay Mansour, and Andrew Y Ng. Approximate planning in large pomdps via reusable trajectories. In *Advances in Neural Information Processing Systems*, pages 1001–1007, 2000.

[98] A Khaled, K Mishchenko, and P Richtárik. Tighter theory for local sgd on identical and heterogeneous data. In *The 23rd International Conference on Artificial Intelligence and Statistics (AISTATS 2020)*, 2020.

[99] Ahmed Khaled, Konstantin Mishchenko, and Peter Richtárik. First analysis of local gd on heterogeneous data. *NeurIPS Workshop on Federated Learning for Data Privacy and Confidentiality*, 2019.

[100] Rahul Kidambi, Praneeth Netrapalli, Prateek Jain, and Sham Kakade. On the insufficiency of existing momentum schemes for stochastic optimization. In *2018 Information Theory and Applications Workshop (ITA)*, pages 1–9. IEEE, 2018.

[101] Seyoung Kim and Eric P Xing. Tree-guided group lasso for multi-task regression with structured sparsity. *ICML*, 2010.

[102] Jens Kober and Jan R Peters. Policy search for motor primitives in robotics. In *Advances in neural information processing systems*, pages 849–856, 2009.

[103] Jelle R Kok and Nikos Vlassis. Collaborative multiagent reinforcement learning by payoff propagation. *JMLR*, 7(Sep):1789–1828, 2006.

[104] Anastasia Koloskova, Nicolas Loizou, Sadra Boreiri, Martin Jaggi, and Sebastian U Stich. A unified theory of decentralized sgd with changing topology and local updates. *arXiv preprint arXiv:2003.10422*, 2020.

[105] Akshay Krishnamurthy, Alekh Agarwal, and John Langford. Pac reinforcement learning with rich observations. In *Advances in Neural Information Processing Systems*, pages 1840–1848, 2016.

[106] Monica S Lam. Autonomy and privacy with open federated virtual assistants.

[107] Guillaume Lample and Devendra Singh Chaplot. Playing fps games with deep reinforcement learning. *arXiv preprint arXiv:1609.05521*, 2016.

[108] Jason Lee, Yuekai Sun, and Michael Saunders. Proximal newton-type methods for convex optimization. In *NIPS*, pages 836–844, 2012.

[109] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. *MLSys*, 2020.

[110] Xiang Li, Kaixuan Huang, Wenhao Yang, Shusen Wang, and Zhihua Zhang. On the convergence of fedavg on non-iid data. *ICLR*, 2020.

[111] Xiujun Li, Yun-Nung Chen, Lihong Li, Jianfeng Gao, and Asli Celikyilmaz. End-to-end task-completion neural dialogue systems. *arXiv preprint arXiv:1703.01008*, 2017.

[112] Z Li, Y Hong, and Z Zhang. Do on-demand ride-sharing services affect traffic congestion? evidence from uber entry. Technical report, Working paper, available at SSRN: https://ssrn. com/abstract= 2838043, 2016.

[113] Xianfeng Liang, Shuheng Shen, Jingchang Liu, Zhen Pan, Enhong Chen, and Yifei Cheng. Variance reduced local sgd with lower communication complexity. *arXiv preprint arXiv:1912.12844*, 2019.

[114] Kaixiang Lin, Shu Wang, and Jiayu Zhou. Collaborative deep reinforcement learning. *arXiv preprint arXiv:1702.05796*, 2017.

[115] Kaixiang Lin, Jianpeng Xu, Inci M Baytas, Shuiwang Ji, and Jiayu Zhou. Multi-task feature interaction learning. In *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1735–1744, 2016.

[116] Kaixiang Lin, Renyu Zhao, Zhe Xu, and Jiayu Zhou. Efficient large-scale fleet management via multi-agent deep reinforcement learning. *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge*, 2018.

[117] Kaixiang Lin and Jiayu Zhou. Interactive multi-task relationship learning. In *2016 IEEE 16th International Conference on Data Mining (ICDM)*, pages 241–250. IEEE, 2016.

[118] Kaixiang Lin and Jiayu Zhou. Ranking policy gradient. In *International Conference on Learning Representations*, 2020.

[119] Chaoyue Liu and Mikhail Belkin. Accelerating sgd with momentum for over-parameterized learning. *ICLR*, 2020.

[120] Jun Liu, Shuiwang Ji, and Jieping Ye. Multi-task feature learning via efficient $\ell_{2,1}$-norm minimization. In *Proceedings of the 25th conference on UAI*, pages 339–348. AUAI Press, 2009.

[121] Jun Liu and Jieping Ye. Efficient l1/lq norm regularization. *arXiv:1009.4766*, 2010.

[122] Tianyi Liu, Zhehui Chen, Enlu Zhou, and Tuo Zhao. Toward deeper understanding of nonconvex stochastic optimization with momentum using diffusion approximations. *arXiv preprint arXiv:1802.05155*, 2018.

[123] Wei Liu, Li Chen, Yunfei Chen, and Wenyi Zhang. Accelerating federated learning via momentum gradient descent. *IEEE Transactions on Parallel and Distributed Systems*, 2020.

[124] Yashu Liu, Jie Wang, and Jieping Ye. An efficient algorithm for weak hierarchical lasso. In *SIGKDD*, pages 283–292. ACM, 2014.

[125] Ryan Lowe, Yi Wu, Aviv Tamar, Jean Harb, Pieter Abbeel, and Igor Mordatch. Multi-agent actor-critic for mixed cooperative-competitive environments. *arXiv preprint arXiv:1706.02275*, 2017.

[126] Lyft.

[127] Siyuan Ma, Raef Bassily, and Mikhail Belkin. The power of interpolation: Understanding the effectiveness of sgd in modern over-parametrized learning. *ICML*, 2018.

[128] Michał Maciejewski and Kai Nagel. The influence of multi-agent cooperation on the efficiency of taxi dispatching. In *PPAM*, pages 751–760. Springer, 2013.

[129] H Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, et al. Communication-efficient learning of deep networks from decentralized data. *Proceedings of the 20 th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2017.

[130] Prem Melville and Vikas Sindhwani. Recommender systems. In *Encyclopedia of machine learning*, pages 829–838. Springer, 2011.

[131] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy P Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. *arXiv preprint arXiv:1602.01783*, 2016.

[132] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

[133] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of machine learning*. MIT press, 2018.

[134] Eric Moulines and Francis R Bach. Non-asymptotic analysis of stochastic approximation algorithms for machine learning. In *Advances in Neural Information Processing Systems*, pages 451–459, 2011.

[135] Rémi Munos, Tom Stepleton, Anna Harutyunyan, and Marc Bellemare. Safe and efficient off-policy reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 1054–1062, 2016.

[136] Ofir Nachum, Mohammad Norouzi, Kelvin Xu, and Dale Schuurmans. Bridging the gap between value and policy based reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 2775–2785, 2017.

[137] Arun Nair, Praveen Srinivasan, Sam Blackwell, Cagdas Alcicek, Rory Fearon, Alessandro De Maria, Vedavyas Panneershelvam, Mustafa Suleyman, Charles Beattie, Stig Petersen, et al. Massively parallel methods for deep reinforcement learning. *arXiv preprint arXiv:1507.04296*, 2015.

[138] Deanna Needell, Rachel Ward, and Nati Srebro. Stochastic gradient descent, weighted sampling, and the randomized kaczmarz algorithm. In *Advances in neural information processing systems*, pages 1017–1025, 2014.

[139] Andrew Y Ng, Daishi Harada, and Stuart Russell. Policy invariance under reward transformations: Theory and application to reward shaping. In *ICML*, volume 99, pages 278–287, 1999.

[140] Duc Thien Nguyen, Akshat Kumar, and Hoong Chuin Lau. Collective multiagent sequential decision making under uncertainty. *AAAI*, 2017.

[141] Duc Thien Nguyen, Akshat Kumar, and Hoong Chuin Lau. Policy gradient with value function approximation for collective multiagent planning. *NIPS*, 2017.

[142] Guillaume Obozinski, Ben Taskar, and Michael I Jordan. Joint covariate selection and joint subspace selection for multiple classification problems. *Statistics and Computing*, 20(2):231–252, 2010.

[143] Brendan O'Donoghue. Variational bayesian reinforcement learning with regret bounds. *arXiv preprint arXiv:1807.09647*, 2018.

[144] Brendan O'Donoghue, Remi Munos, Koray Kavukcuoglu, and Volodymyr Mnih. Combining policy gradient and q-learning. *arXiv preprint arXiv:1611.01626*, 2016.

[145] Junhyuk Oh, Yijie Guo, Satinder Singh, and Honglak Lee. Self-imitation learning. *arXiv preprint arXiv:1806.05635*, 2018.

[146] Junhyuk Oh, Matteo Hessel, Wojciech M Czarnecki, Zhongwen Xu, Hado van Hasselt, Satinder Singh, and David Silver. Discovering reinforcement learning algorithms. *arXiv preprint arXiv:2007.08794*, 2020.

[147] OpenAI. Openai universe-starter-agent. `https://github.com/openai/universe-starter-agent`, 2017. Accessed: 2017-0201.

[148] Takayuki Osa, Joni Pajarinen, Gerhard Neumann, J Andrew Bagnell, Pieter Abbeel, Jan Peters, et al. An algorithmic perspective on imitation learning. *Foundations and Trends® in Robotics*, 7(1-2):1–179, 2018.

[149] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *TKDE*, 22(10):1345–1359, 2010.

[150] Neal Parikh and Stephen P Boyd. Proximal algorithms. *Foundations and Trends in optimization*, 1(3):127–239, 2014.

[151] Emilio Parisotto, Jimmy Lei Ba, and Ruslan Salakhutdinov. Actor-mimic: Deep multitask and transfer reinforcement learning. *arXiv preprint arXiv:1511.06342*, 2015.

[152] Jan Peters and Stefan Schaal. Reinforcement learning by reward-weighted regression for operational space control. In *Proceedings of the 24th international conference on Machine learning*, pages 745–750. ACM, 2007.

[153] Jan Peters and Stefan Schaal. Reinforcement learning of motor skills with policy gradients. *Neural networks*, 21(4):682–697, 2008.

[154] Tu-Hoa Pham, Giovanni De Magistris, and Ryuki Tachibana. Optlayer-practical constrained optimization for deep reinforcement learning in the real world. *arXiv preprint arXiv:1709.07643*, 2017.

[155] J Platt. Fast training of support vector machines using sequential minimal optimization, in, b. scholkopf, c. burges, a. smola,(eds.): Advances in kernel methods-support vector learning, 1998.

[156] Martin L Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

[157] Zhaonan Qu*, Kaixiang Lin*, Jayant Kalagnanam, Zhaojian Li, Jiayu Zhou, and Zhengyuan Zhou. Federated learning's blessing: Fedavg has linear speedup. *arXiv preprint arXiv:2007.05690*, 2020, * denotes equal contribution.

[158] Peter Radchenko and Gareth M James. Variable selection using adaptive nonlinear interaction structures in high dimensions. *Journal of the American Statistical Association*, 105(492):1541–1553, 2010.

[159] Janarthanan Rajendran, Aravind Lakshminarayanan, Mitesh M Khapra, Balaraman Ravindran, et al. A2t: Attend, adapt and transfer: Attentive deep architecture for adaptive transfer from multiple sources. *arXiv preprint arXiv:1510.02879*, 2015.

[160] Tabish Rashid, Mikayel Samvelyan, Christian Schroeder de Witt, Gregory Farquhar, Jakob Foerster, and Shimon Whiteson. Qmix: Monotonic value function factorisation for deep multi-agent reinforcement learning. *arXiv preprint arXiv:1803.11485*, 2018.

[161] R Tyrrell Rockafellar. *Convex analysis*. Number 28. Princeton university press, 1970.

[162] Bernardino Romera-Paredes, Hane Aung, Nadia Bianchi-Berthouze, and Massimiliano Pontil. Multilinear multitask learning. In *ICML*, pages 1444–1452, 2013.

[163] Stéphane Ross and Drew Bagnell. Efficient reductions for imitation learning. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 661–668, 2010.

[164] Stephane Ross and J Andrew Bagnell. Reinforcement and imitation learning via interactive no-regret learning. *arXiv preprint arXiv:1406.5979*, 2014.

[165] Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 627–635, 2011.

[166] Andrei A Rusu, Sergio Gomez Colmenarejo, Caglar Gulcehre, Guillaume Desjardins, James Kirkpatrick, Razvan Pascanu, Volodymyr Mnih, Koray Kavukcuoglu, and Raia Hadsell. Policy distillation. *arXiv preprint arXiv:1511.06295*, 2015.

[167] Tom Schaul, John Quan, Ioannis Antonoglou, and David Silver. Prioritized experience replay. *arXiv preprint arXiv:1511.05952*, 2015.

[168] Mark Schmidt and Nicolas Le Roux. Fast convergence of stochastic gradient descent under a strong growth condition. *arXiv preprint arXiv:1308.6370*, 2013.

[169] John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438*, 2015.

[170] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[171] Anton Schwaighofer, Volker Tresp, and Kai Yu. Learning gaussian process kernels via hierarchical bayes. In *NIPS*, pages 1209–1216, 2004.

[172] Kiam Tian Seow, Nam Hai Dang, and Der-Horng Lee. A collaborative multiagent taxi-dispatch system. *IEEE T-ASE*, 7(3):607–616, 2010.

[173] Burr Settles. Active learning literature survey. *University of Wisconsin, Madison*, 52(55-66):11, 2010.

[174] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *Nature*, 529(7587):484–489, 2016.

[175] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. Mastering the game of go without human knowledge. *Nature*, 550(7676):354, 2017.

[176] Virginia Smith, Chao-Kai Chiang, Maziar Sanjabi, and Ameet S Talwalkar. Federated multi-task learning. In *Advances in Neural Information Processing Systems*, pages 4424–4434, 2017.

[177] Mark John Somers. Organizational commitment, turnover and absenteeism: An examination of direct and interaction effects. *Journal of Organizational Behavior*, 16(1):49–58, 1995.

[178] Sebastian U Stich. Local sgd converges fast and communicates little. *ICLR*, 2019.

[179] Alexander L Strehl, Lihong Li, and Michael L Littman. Reinforcement learning in finite mdps: Pac analysis. *Journal of Machine Learning Research*, 10(Nov):2413–2444, 2009.

[180] Alexander L Strehl, Lihong Li, Eric Wiewiora, John Langford, and Michael L Littman. Pac model-free reinforcement learning. In *Proceedings of the 23rd international conference on Machine learning*, pages 881–888. ACM, 2006.

[181] Thomas Strohmer and Roman Vershynin. A randomized kaczmarz algorithm with exponential convergence. *Journal of Fourier Analysis and Applications*, 15(2):262, 2009.

[182] Rukhsana Sultana, Debra Boyd-Kimball, H Fai Poon, Jian Cai, William M Pierce, Jon B Klein, Michael Merchant, William R Markesbery, and D Allan Butterfield. Redox proteomics identification of oxidized proteins in alzheimer's disease hippocampus and cerebellum: an approach to understand pathological and biochemical alterations in ad. *Neurobiology of aging*, 27(11):1564–1576, 2006.

[183] Qian Sun, Rita Chattopadhyay, Sethuraman Panchanathan, and Jieping Ye. A two-stage weighting framework for multi-source domain adaptation. In *NIPS*, pages 505–513, 2011.

[184] Wen Sun, Arun Venkatraman, Geoffrey J Gordon, Byron Boots, and J Andrew Bagnell. Deeply aggrevated: Differentiable imitation learning for sequential prediction. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 3309–3318. JMLR. org, 2017.

[185] Peter Sunehag, Guy Lever, Audrunas Gruslys, Wojciech Marian Czarnecki, Vinicius Zambaldi, Max Jaderberg, Marc Lanctot, Nicolas Sonnerat, Joel Z Leibo, Karl Tuyls, et al. Value-decomposition networks for cooperative multi-agent learning. *arXiv preprint arXiv:1706.05296*, 2017.

[186] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge, 1998.

[187] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[188] Umar Syed and Robert E Schapire. A reduction from apprenticeship learning to classification. In *Advances in Neural Information Processing Systems*, pages 2253–2261, 2010.

[189] Ardi Tampuu, Tambet Matiisen, Dorian Kodelja, Ilya Kuzovkin, Kristjan Korjus, Juhan Aru, Jaan Aru, and Raul Vicente. Multiagent cooperation and competition with deep reinforcement learning. *PloS one*, 12(4):e0172395, 2017.

[190] Ming Tan. Multi-agent reinforcement learning: Independent vs. cooperative agents. In *ICML*, pages 330–337, 1993.

[191] Matthew E Taylor and Peter Stone. Transfer learning for reinforcement learning domains: A survey. *JMLR*, 10(Jul):1633–1685, 2009.

[192] Stefan J Teipel, Wolfram Bayer, Gene E Alexander, York Zebuhr, Diane Teichberg, Luka Kulic, Marc B Schapiro, Hans-Jürgen Möller, Stanley I Rapoport, and Harald

Hampel. Progression of corpus callosum atrophy in alzheimer disease. *Archives of Neurology*, 59(2):243–248, 2002.

[193] Devinder Thapa, In-Sung Jung, and Gi-Nam Wang. Agent based decision support system using reinforcement learning under emergency circumstances. In *International Conference on Natural Computation*, pages 888–892. Springer, 2005.

[194] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996.

[195] Ryota Tomioka, Kohei Hayashi, and Hisashi Kashima. Estimation of low-rank tensors via convex optimization. *arXiv preprint arXiv:1010.0789*, 2010.

[196] Ryota Tomioka and Taiji Suzuki. Convex tensor decomposition via structured schatten norm regularization. In *NIPS*, pages 1331–1339, 2013.

[197] Ahmed Touati, Pierre-Luc Bacon, Doina Precup, and Pascal Vincent. Convergent tree-backup and retrace with function approximation. *arXiv preprint arXiv:1705.09322*, 2017.

[198] Paul Tseng. Convergence of a block coordinate descent method for nondifferentiable minimization. *Journal of optimization theory and applications*, 109(3):475–494, 2001.

[199] Berwin A Turlach, William N Venables, and Stephen J Wright. Simultaneous variable selection. *Technometrics*, 47(3):349–363, 2005.

[200] Eric Tzeng, Judy Hoffman, Trevor Darrell, and Kate Saenko. Simultaneous deep transfer across domains and tasks. In *ICCV*, pages 4068–4076, 2015.

[201] Uber.

[202] Leslie G Valiant. A theory of the learnable. In *Proceedings of the sixteenth annual ACM symposium on Theory of computing*, pages 436–445. ACM, 1984.

[203] Hado Van Hasselt, Arthur Guez, and David Silver. Deep reinforcement learning with double q-learning. In *AAAI*, volume 2, page 5. Phoenix, AZ, 2016.

[204] Vladimir Vapnik. *Estimation of dependences based on empirical data*. Springer Science & Business Media, 2006.

[205] Jianyu Wang and Gauri Joshi. Cooperative sgd: A unified framework for the design and analysis of communication-efficient sgd algorithms. *arXiv preprint arXiv:1808.07576*, 2018.

[206] Shiqiang Wang, Tiffany Tuor, Theodoros Salonidis, Kin K Leung, Christian Makaya, Ting He, and Kevin Chan. Adaptive federated learning in resource constrained edge computing systems. *IEEE Journal on Selected Areas in Communications*, 37(6):1205–1221, 2019.

[207] Zirui Wang, Zihang Dai, Barnabás Póczos, and Jaime Carbonell. Characterizing and avoiding negative transfer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 11293–11302, 2019.

[208] Ziyu Wang, Victor Bapst, Nicolas Heess, Volodymyr Mnih, Remi Munos, Koray Kavukcuoglu, and Nando de Freitas. Sample efficient actor-critic with experience replay. *arXiv preprint arXiv:1611.01224*, 2016.

[209] Ziyu Wang, Tom Schaul, Matteo Hessel, Hado Van Hasselt, Marc Lanctot, and Nando De Freitas. Dueling network architectures for deep reinforcement learning. *arXiv preprint arXiv:1511.06581*, 2015.

[210] Malcolm Ware, Eibe Frank, Geoffrey Holmes, Mark Hall, and Ian H Witten. Interactive machine learning: letting users build classifiers. *INT J HUM-COMPUT ST*, 55(3):281–292, 2001.

[211] Chong Wei, Yinhu Wang, Xuedong Yan, and Chunfu Shao. Look-ahead insertion policy for a shared-taxi system based on reinforcement learning. *IEEE Access*, 2017.

[212] Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256, 1992.

[213] Blake E Woodworth, Jialei Wang, Adam Smith, Brendan McMahan, and Nati Srebro. Graph oracle models, lower bounds, and gaps for parallel stochastic optimization. In *Advances in neural information processing systems*, pages 8496–8506, 2018.

[214] Stephen J Wright, Robert D Nowak, and Mário AT Figueiredo. Sparse reconstruction by separable approximation. *Signal Processing, IEEE Transactions on*, 57(7):2479–2493, 2009.

[215] Sicheng Xiong, Javad Azimi, and Xiaoli Z Fern. Active learning of constraints for semi-supervised clustering. *TKDE*, 26(1):43–54, 2014.

[216] Jianpeng Xu, Pang-Ning Tan, and Lifeng Luo. Orion: Online regularized multi-task regression and its application to ensemble forecasting. In *ICDM*, pages 1061–1066. IEEE, 2014.

[217] Yaodong Yang, Rui Luo, Minne Li, Ming Zhou, Weinan Zhang, and Jun Wang. Mean field multi-agent reinforcement learning. *ICML*, 2018.

[218] Hao Yu, Rong Jin, and Sen Yang. On the linear speedup analysis of communication efficient momentum sgd for distributed non-convex optimization. *ICML*, 2019.

[219] Hao Yu, Sen Yang, and Shenghuo Zhu. Parallel restarted sgd with faster convergence and less communication: Demystifying why model averaging works for deep learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 5693–5700, 2019.

[220] Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. *arXiv preprint arXiv:2001.06782*, 2020.

[221] Kun Yuan, Bicheng Ying, and Ali H Sayed. On the influence of momentum acceleration on online learning. *The Journal of Machine Learning Research*, 17(1):6602–6667, 2016.

[222] Lei Yuan, Jun Liu, and Jieping Ye. Efficient methods for overlapping group lasso. In *NIPS*, pages 352–360, 2011.

[223] Andrea Zanette and Emma Brunskill. Tighter problem-dependent regret bounds in reinforcement learning without domain knowledge using value function bounds. *arXiv preprint arXiv:1901.00210*, 2019.

[224] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. *arXiv preprint arXiv:1611.03530*, 2016.

[225] Yi Zhang and Jeff G Schneider. Learning multiple tasks with a sparse matrix-normal penalty. In *NIPS*, pages 2550–2558, 2010.

[226] Yu Zhang and Qiang Yang. A survey on multi-task learning. *arXiv preprint arXiv:1707.08114*, 2017.

[227] Yu Zhang and Dit-Yan Yeung. A convex formulation for learning task relationships in multi-task learning. *arXiv preprint arXiv:1203.3536*, 2012.

[228] Yu Zhang, Dit-Yan Yeung, and Qian Xu. Probabilistic multi-task feature selection. In *NIPS*, pages 2559–2567, 2010.

[229] Zhanpeng Zhang, Ping Luo, Chen Change Loy, and Xiaoou Tang. Facial landmark detection by deep multi-task learning. In *ECCV*, pages 94–108. Springer, 2014.

[230] Lianmin Zheng, Jiacheng Yang, Han Cai, Weinan Zhang, Jun Wang, and Yong Yu. Magent: A many-agent reinforcement learning platform for artificial collective intelligence. *arXiv preprint arXiv:1712.00600*, 2017.

[231] Fan Zhou and Guojing Cong. On the convergence properties of a k-step averaging stochastic gradient descent algorithm for nonconvex optimization. *IJCAI*, 2018.

[232] Jiayu Zhou, Jianhui Chen, and Jieping Ye. Clustered multi-task learning via alternating structure optimization. In *NIPS*, pages 702–710, 2011.

[233] Jiayu Zhou, Jianhui Chen, and Jieping Ye. Malsar: Multi-task learning via structural regularization. *Arizona State University*, 2011.

[234] Jiayu Zhou, Jun Liu, Vaibhav A Narayan, Jieping Ye, Alzheimer's Disease Neuroimaging Initiative, et al. Modeling disease progression via multi-task learning. *NeuroImage*, 78:233–248, 2013.

[235] Jiayu Zhou, Lei Yuan, Jun Liu, and Jieping Ye. A multi-task learning formulation for predicting disease progression. In *SIGKDD*, pages 814–822. ACM, 2011.

[236] Yuke Zhu, Roozbeh Mottaghi, Eric Kolve, Joseph J Lim, Abhinav Gupta, Li Fei-Fei, and Ali Farhadi. Target-driven visual navigation in indoor scenes using deep reinforcement learning. *arXiv preprint arXiv:1609.05143*, 2016.