TOWARD OPEN WORLD VISUAL UNDERSTANDING

By

Wentao Bao

A DISSERTATION

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

Computer Science—Doctor of Philosophy

2024

ABSTRACT

Visual data such as images and videos are the most prominent media to record, transmit, and exchange information in this era. Though we have witnessed waves of success in visual intelligence, teaching machines to understand visual content at the level of human intelligence remains a fundamental challenge. In past decades, visual understanding has been extensively explored through computer vision tasks such as object (or activity) recognition, segmentation, and detection. However, existing methods can hardly be deployed in real open-world applications where *unseen* environments, objects, and activities inevitably appear in testing. Such a limitation is attributed to the closed-world assumption that ignores the unknown in model design, learning, and evaluation.

In this dissertation, I will introduce my works that go beyond the traditional closed-world visual understanding and tackle several challenging open-world problems. The ultimate goal is to endow machines with visual perception capabilities in an open world, where unseen environments, image objects, and video activities can be handled. First, I will begin the dissertation by investigating **open-world visual forecasting** problems in an *unseen perception environment*. Specifically, I primarily explore how the early observed videos can be leveraged to promptly forecast the traffic accident risk for safe self-driving (in Chapter 2 and Chapter 3), and forecast the 3D hand motion trajectory in an unseen first-person view (in Chapter 4). Second, I will cover the **open-world visual recognition** problems that aim to *identify the unseen visual concepts*. In this part, we are especially interested in identifying and localizing unseen human actions in general videos (in Chapter 5 and Chapter 6). Lastly, I will delve into **open-world visual language understanding** problems that further *recognize unseen visual concepts* from language queries, including the recognition of unseen compositional objects in images (in Chapter 7) and spatiotemporally detecting unseen human actions (in Chapter 8).

In Chapter 9, I summarize the main contributions of this dissertation and discuss unsolved challenges in real-world practices. Based on the line of the dissertation research, some future directions for open-world visual understanding are briefly discussed.

ACKNOWLEDGEMENTS

I would like to extend many thanks to my advisor Yu Kong, for all his efforts in making my five years of Ph.D. fruitful and enjoyable. I am fortunate to have received his hands-on guidance to become an independent researcher. The guidance such as critical thinking, paper writing and presentation, and professional communication, are really helpful to my growth. Particularly, his durable passion for research impresses me a lot, with countless brainstorming discussion hours to exchange cutting-edge ideas. These discussions have motivated most of my top-tier publications with valuable research questions and greatly improved my research taste in the past years. Besides, I want to extend my sincere gratitude to my co-advisor Qi Yu when I was in RIT for the first three years. Though Prof. Qi Yu is a senior professor with a large group of Ph.D. students, he is always supportive of my research, with many insightful comments on my projects. I am impressed by his expertise and emphasis on machine learning, which equipped me with a solid ML background for doing computer vision research in the early years of my Ph.D. journey.

Besides, I want to sincerely thank my dissertation committee members, Xiaoming Liu, Vishnu Boddeti, and Daniel Morris. Their thoughtful advice and cooperative spirit make my dissertation and defense presentation of high quality at a smooth pace. I particularly want to thank Xiaoming and Vishnu for their recommendations for my career search and grant proposal collaboration, and thank Daniel for his prompt and insightful feedback on my defense. I also want to thank many other excellent faculties and staff from MSU such as Sandeep Kulkarni and Vincent Mattison, and those from RIT such as Pengcheng Shi, Linwei Wang, Rui Li, and Min-Hong Fu, who collectively contributed to excellent doctoral programs in MSU and RIT.

Over the past five years, I would like to thank many researchers, engineers, and collaborators who shaped my research interest and contributed to my fruitful doctoral outcomes and career development. Particularly, I want to thank Terrance Boult and Walter Scheirer for their decades of research in open-set recognition as well as their kind and insightful in-person discussions, that inspired my dedication to this dissertation. I want to thank Jason Corso for his interesting and inspiring advice when he was my doctoral consortium advisor and thank Junsong Yuan and

Heng Huang for paper collaboration and career search recommendations. I also want to thank many scholars with helpful discussions of my dissertation research: Guangyao Chen, Zhen Fang, Bolei Zhou, Xiang Pan, Tianyu Luan, Yuanhao Zhai, Shuai Li. I want to thank Weishi Shi for this countless help in my career search preparation. Also, I want to extend many thanks to my excellent internship mentors and managers, Kai Li, Martin Renqiang Min, Deep Patel from NEC Lab America, Lele Chen, Zhong Li, Yi Xu from OPPO US Research Center, and Weiyu Zhang, Yongxi Lu from Apple. I believe that high-quality research outcomes cannot be achieved without excellent collaborators. I would like to thank my collaborators and those smart Master students that I have honorably mentored, Junwen Chen, Yifan Li, Yuxiao Chen, Lichang Chen, Libing Zeng, Yuansheng Zhu, Xinmiao Lin, Xiwen Dengxiong, Hanbin Hong, Suhan Park.

Apart from academics, I value the memorable friendships that have been supporting me to survive and refresh along the Ph.D. journey. I would like to extend my particular thanks to my senior lab mate Junwen Chen, who not only helped my research a lot but also comradely walked me through those difficult early times since my first visit to the U.S. Besides, I want to thank Yifan Li, Yujiang Pu, Zhanbo Huang, Suhan Park, Kwonyeong Cho, Xinmiao Lin, Xiwen Dengxiong, Haiting Hao, Anna Starynska, Chuanqi Zang, Hanbin Hong, and many other ActionLab students, who made the lives of ActionLab enjoyable and wonderful. I also want to thank many many of my friends including but not limited to Lan Wang, Zhiyuan Ren, Andrew Hou, Haobo Zhang, Yijiang Pang, Xiajun Jiang, Yuansheng Zhu, Zixin Yang, and Grace Cao, for their great friendships.

Last, but far from least, I must extend my sincere thanks to my family, who have been supporting my overseas Ph.D. study with unconditional love. I truly appreciate their understanding of my life decision to study abroad, especially in the hardships of the past pandemic times. In a word, my family is always the harbor of my life journey.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

1.1 Research Motivation

Visual understanding has been one of the most fundamental directions in computer vision. Its goal is to teach machines to understand the visual world analogous to humans from data captured by camera sensors. For a long time, visual understanding has been investigated in lab-controlled environments using small-scale data [54, 157]. In the past decades, such a situation has been revolutionized since the arising of deep learning methods [274, 106, 328, 65], large-scale datasets [53, 111, 22, 1], and advanced computation hardware. However, when it comes to real-world applications, most existing visual understanding methods can hardly work well because of dynamically changing visual environments and test-time requirements [363]. In reality, it is impractical to assume infinite observations or annotations in testing. In this dissertation, we denote such a testing scenario as an *open world*, in which we identify two key aspects in terms of the unknowns with respect to the model learning: (1) unknown data observation and (2) unknown task requirements. The former can be attributed to the infeasible access to the incoming data such as streaming videos, or unseen testing environments. The latter often deals with task requirements that contain new semantics, e.g., identifying unseen wild animals. In summary, at every testing moment in an open world, the visual understanding systems could be fed with only the historical observational data, in an unseen environment, and asked to identify unknown concepts. Given these open-world situations, we ask the question: *how to achieve the visual understanding of various unknowns in an open world?*

To answer the question, we first begin with open-world visual forecasting, because future events are naturally unseen in streaming videos. Due to limited visual observations, it requires a comprehensive abstraction of physical patterns and temporal dynamics from early observed visual data. In this dissertation, we take the accident risk anticipation for self-driving safety and 3D hand trajectory prediction for virtual reality applications as case studies. Three types of unseen are handled as shown in Fig. 1.1 (**left**). Specifically, the first one is to answer the question: *how to*

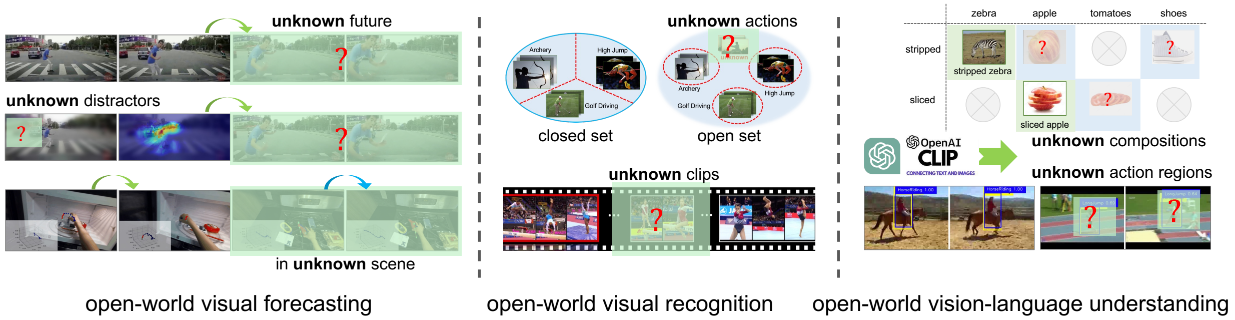


Figure 1.1 **Open-world Visual Understanding.** The many "unknowns" in an open world could be handled in various computer vision applications. This dissertation primarily covers the topics of visual forecasting, recognition, and vision-language understanding in an open world.

predict the unseen future accident risk from dashed camera videos? We answer this question from a Bayesian probabilistic view such that the predictive uncertainties of unseen future risk can be quantified in real-world testing. Furthermore, the second one is the unknown distractors in accident forecasting. This motivates us to study which regions are key to the model to "visually" attend in accident forecasting. Our work, for the first time, introduces selective visual attention to suppress the unknown driving distractors in forecasting. Lastly, when a forecasting system is deployed to a 3D physical world, it is interesting to answer the question: *can the model perform forecasting in an unknown 3D scene?* This motivates us to explore 3D hand trajectory forecasting in an open world where the testing scenes are unseen.

Next, regarding open-world visual recognition problems, we consider the most fundamental research question: *can we build a video model knowing what is unknown?* The unknown means that in testing, either the entire video contains unseen activities or only a few short clips of a long video contain unseen activities, as shown in Fig. 1.1 (**middle**). This is challenging because the model does not have any information about test-time unknowns in training, as evidenced by many open-set recognition works in the image understanding domain [278, 16, 86]. Furthermore, specific to the video modality, some key challenges such as how to efficiently quantify the unknown scores for large-scale video data and how to deal with the open-world temporal dynamics, have never been explored in literature. With these motivations, we develop the first works for the open-set video understanding tasks, including open-set video recognition and open-set temporal action

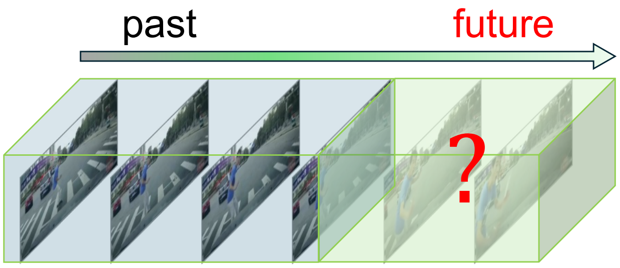


Figure 1.2 Visual Forecasting.

localization, that aim to recognize and localize the unseen human activities in videos.

Lastly, with the recent trend of language modeling in computer vision, we are further interested in the power of the current vision-language foundation models such as CLIP [261] and large-language models (LLM) [247] in handling more complex open-world visual understanding problems, as shown in Fig. 1.1 (**right**). Under the CLIP-like vision-language modeling paradigm, two complex open-world visual understanding research questions are explored in this dissertation. The first one is to understand unseen compositional concepts from images. Different from the prior works [239, 214], we aim to ask the question: *how to effectively leverage LLMs to understand images of unseen compositional concepts?* We introduce the idea of prompting the language-informed distributions when adapting the CLIP model. The other one is to understand unseen action regions from videos. This is motivated by the question: *can we detect any unseen video actions from an open-ended action vocabulary?* For the first time, we formulate it as an open-vocabulary action detection problem and explore key factors when adapting video-based CLIP models.

1.2 Organization

1.2.1 Part I: Open-World Visual Forecasting

Forecasting from early observed visual data is essentially to mimic the capability of human imagination. We human beings are able to imagine how things will evolve in time. For example, human drivers could avoid tragic collisions with other cars or pedestrians by a subconscious mind of the trend of their motions [329]. However, such a capability is extremely challenging for machines to learn from video data. The challenges lie in the following aspects. First, intuitively the less we have observed, the more uncertain our forecasting about the future will be. Furthermore, when the videos are captured from egocentric views, the relative motion between an ego-agent and the dynamic environments incurs more complexity in understanding the trend of physical motion from videos. Lastly, instead of video frame prediction, exploring visual forecasting problems in specific in-the-wild applications, such as traffic accident risks or 3D hand trajectories, requires injecting domain-specific knowledge or constraints into the model learning. These aspects are even more challenging in an open world where unseen distractors or environments are tested.

In this dissertation, we explore the challenges above through the lens of traffic accident anticipation and 3D hand trajectory forecasting problems, enabling safe autonomous driving and robust virtual reality applications. Specifically, in Sec. 2, we develop a novel model to early predict the occurrence of future traffic accidents from dashcam videos. In this work, we collected accident videos recorded by dash cameras mounted on driving cars, and annotated the start timestamps of the traffic accidents. The task is to early predict whether there will be an accident or not before it happens. Technically, the model first detects the traffic objects on each frame which are further structured as a graph. Then, the model is learned to capture the spatial and temporal dynamics from the sequence of graphs. Finally, the learned features are used to predict the probability of accident occurrence by Bayesian neural networks (BNNs) [241], which provide uncertainty estimation in prediction. The Bayesian uncertainty modeling is valuable for providing a learning regularization in training, and more importantly, the variability of unknown future accident risk can be quantified to achieve trustworthy autonomous driving in an open world.

Motivated by this work, in Sec. 3, we further explore the traffic accident anticipation by considering the visual explanation, that interprets how the unseen distractors are suppressed in forecasting. We argue that in autonomous driving scenarios, a better way to achieve trustworthy decisions from input videos is to provide visual attentional cues. Inspired by visual attention modeling which mimics the human visual attention mechanism [85, 245], we propose to formulate the driver's visual attention and the system's accident anticipation in a single Markov decision process (MDP). At each time step, our model “fixates" at the most risky regions on dashcam videos and predicts the occurrence of future accidents as well as the next fixation point. The model is

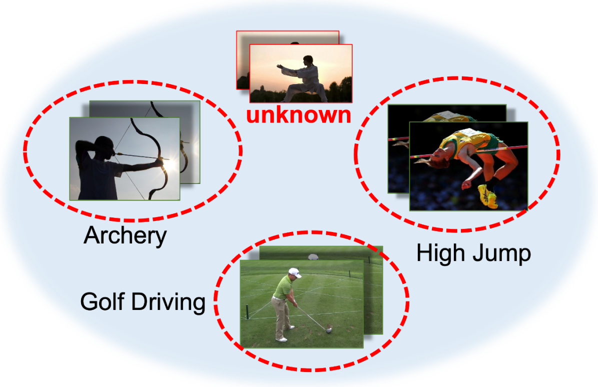


Figure 1.3 Open-Set Action Recognition.

learned by deep reinforcement learning (DRL) and achieves much better performance than prior arts while providing visual attention as the explanation.

In addition, we explore the visual forecasting problem through the tasks of 3D trajectory prediction in the open-world virtual reality scenario (in Sec. 4). It aims to early predict the future hand trajectory in either seen or unseen 3D physical world from egocentric (first-person view) videos, while only giving the historically observed videos and trajectories as input. Different from the previous accident anticipation, this task is challenging due to the requirement of 3D sensing of the indoor environment and helmet wearers from dynamic videos. In this work, for the first time, we build the benchmarks of the egocentric 3D hand trajectory forecasting by collecting and annotating the egocentric videos. Then, we develop a novel and effective model for the task by leveraging recent Transformers [328, 305] and the classical uncertainty-aware state-space modeling. The model empirically achieves the best trajectory forecasting performance on both seen and unseen 3D egocentric scenes.

1.2.2 Open-world Visual Recognition

Detecting the unknown from visual data has been one of the fundamental visual understanding topics. The unknown refers to the objects or actions not defined in the model learning process due to the open-world testing scenario [278, 317]. For example, in a self-driving scenario, if a self-driving system has only learned from common objects such as vehicles, pedestrians, trees, buildings, etc., we expect the system can also know that a wild animal standing on the road is unknown. Without such a capability, there could be a catastrophic traffic accident caused by the self-driving system.

For a long period, visual recognition has been dominated by empirical risk minimization (ERM) of the statistical machine learning [327], which aims to learn a recognition model by minimizing task error over the collected training data. When the test data exhibit a different semantic distribution from the training data, e.g., data from unknown categories exist, the recognition model would fail to identify the unknown. According to whether the unknown data exists in training (without labels), the testing unknown can be treated as the unknown-unknown and the known-unknown [86]. In this dissertation, for the first time, we tackle these two types of unknowns for open-world video understanding. Specifically, detecting the unknown-unknown is studied by open-set recognition while the known-unknown problem is studied by open-set action localization.

In Sec. 5, we formulate the video-based open-set action recognition (OSAR) problem, which aims to learn a video classification model with the capability of identifying the unknown action in testing. The fundamental challenge is to learn a scoring function to identify the unknown when training a video-based classifier. Inspired by evidential deep learning (EDL) [283, 4], we treat the video classification as an evidence collection process such that low total evidence indicates a high classification uncertainty of the testing video, i.e., more likely contains an unknown action. Besides, compared to image-based open-set recognition [278, 147, 34], OSAR models need to overcome the static bias issue that the model tends to learn the shortcut mapping from the confounding frame-level visual features to the class labels. We propose to debias the target video feature by introducing biased auxiliary branches, that encourage the biased feature to be discriminative in classification while statistically independent of the target feature. Eventually, our proposed method, termed the deep evidential action recognition (DEAR) model, achieves superior performance over different video benchmarks using multiple action recognition backbones.

In Sec. 6, following the DEAR work, we further extend the EDL method to tackle the known-unknown detection problem through the open-set temporal action localization (OpenTAL) task, which we formulate as the first work in existing literature. The task aims to temporally localize and recognize human actions while identifying the unknown action in recognition given a testing video.

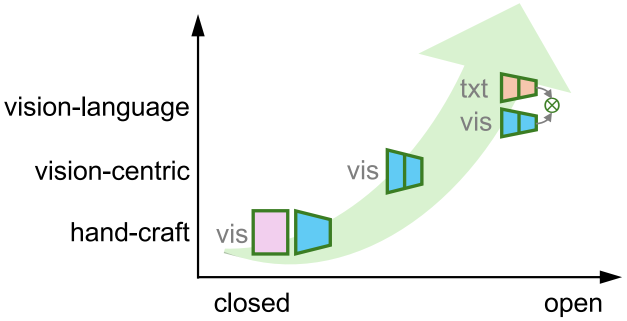


Figure 1.4 Visual understanding paradigm.

Since the model is learned on video data with annotations of only the known action categories, the unknown actions are mixed with the background segments. This presents a unique challenge that the model needs to simultaneously distinguish between the foreground and background and classify between known and unknown actions. To tackle this challenge, we develop an OpenTAL model based on DEAR, that uses evidential uncertainty to identify the unknown action proposals and proposes a semi-supervised binary classification module to predict the actionness score for each proposal. Besides, to comprehensively evaluate the OpenTAL task, we develop a new evaluation protocol, open-set detection rate (OSDR), to benchmark our method. Our method achieves much better performance than baselines that combine existing TAL models [192] with open-set recognition methods.

1.2.3 Open-world Vision-Language Understanding

Revisiting the history of computer vision research, most visual understanding literature has been exploring a key research question: *what feature representation do we need for visual data?* Past research experienced from hand-craft feature engineering [212, 48, 12] to the recent data-driven deep learning paradigm [158, 106, 65]. Despite the success achieved, they are still limited to representing visual features for complex requirements from diversified vision tasks. Thanks to the recent vision-language models [261], the traditional visual understanding evolves into a vision-language alignment problem, as shown in Fig. 1.4. For example, the CLIP model [261] that has been pre-trained on web-scale image-text pairs shows superior zero-shot recognition performance, i.e., recognizing unseen concepts without fine-tuning in an open world. Moreover, in the spirit of "next-token-prediction" by large-language models (LLM) [247], more general visual understanding tasks can be unified by utilizing language models as interfaces to represent task input or output. These advances inspire us to investigate more challenging open-world understanding problems by foundation models such as CLIP and LLM. In this dissertation, we first explore how to leverage LLMs to strengthen the CLIP model for recognizing unseen compositional concepts from image data. Next, for video understanding, we make contributions by adapting video-based CLIP and LLM for localizing unseen human actions in the space and time of videos.

In Sec. 7, the goal is to achieve compositional zero-shot learning (CZSL) capability for image data by pre-trained CLIP. With such a capability, a powerful image recognition model can be learned from limited data without large-scale compositional annotations. For example, the model could recognize unseen compositions such as `sliced tomatoes` when the model only learns the `sliced` from `sliced potatoes` and `tomatoes` from `red tomatoes`. In this work, we leverage LLM to generate multiple descriptions for compositional concepts, which enables class-specific Gaussian distribution modeling for margin-aware prompt optimization and enhances the textual class representation for vision-language alignment. We further decompose the text and image into simple primitives, i.e., states and objects, for hierarchical representation learning. Our method shows superiority in both the closed- and open-world testing environments.

In Sec. 8, for video understanding, we formulate the first open-vocabulary action detection (OVAD) work that could detect any human actions from videos. This work is inspired by existing open-vocabulary learning literature, that uses a pre-trained CLIP model to align visual features with corresponding language semantics from an open-ended vocabulary. To fully exploit the CLIP semantics for recognition and CLIP localizability for spacetime localization, we build an OpenMixer model that bridges the pre-trained video CLIP and detection transformer (DETR) [24] where an LLM is used to obtain generalizable language context for the open-ended action vocabulary.

In Sec. 9, we conclude the dissertation with discussions on the limitations of the mentioned publications and present some ideas for future work.

1.3 Relevant Publications

Following is the list of relevant first-authored publications for each chapter.

- Chapter 2 - Uncertainty-based Traffic Accident Anticipation with Spatio-Temporal Relational Learning [315] (ACM MM 2020)
- Chapter 3 - Deep Reinforced Accident Anticipation with Visual Explanation [316] (ICCV 2021)
- Chapter 4 - Uncertainty-aware State Space Transformer for Egocentric 3D Hand Trajectory Forecasting [310] (ICCV 2023)
- Chapter 5 - Evidential Deep Learning for Open Set Action Recognition [317] (ICCV 2021)
- Chapter 6 - Towards Open Set Temporal Action Localization [318] (CVPR 2022)
- Chapter 7 - Prompting Language-Informed Distribution for Compositional Zero-Shot Learning [311] (ECCV 2024)
- Chapter 8 - Exploiting VLM Localizability and Semantics for Open Vocabulary Action Detection [312] (arXiv 2024)

CHAPTER 2

UNCERTAINTY-BASED ACCIDENT ANTICIPATION

2.1 Introduction

Accident anticipation aims to predict an accident from dashcam video before it happens. It is one of the most important tasks for safety-guaranteed autonomous driving applications and has been receiving increasing attentions in recent years [29, 303, 68, 46]. Thanks to the accident anticipation, the safety level of intelligent systems on vehicles could be significantly enhanced. For example, even a successful anticipation made only a few seconds earlier before the occurrence of an accident can help a self-driving system to make urgent safety control, avoiding a possible car crash accident.

However, accident anticipation is still an extremely challenging task due to the noisy and limited visual cues in an observed dashcam video. Take Fig. 2.1 as an example, a traffic scene captured in an egocentric view is typically crowded with multiple cars, pedestrians, motorcyclists, and so on. In this scenario, accident-relevant visual cues could be overwhelmed by objects that are not relevant to the accident, making an intelligent system insensible to a car crash accident that will happen at the road intersection. Nevertheless, traffic accidents are foreseeable by training a powerful uncertainty-based model to distinguish the accident-relevant cues from noisy video data. For example, the inconsistent motions of multiple vehicles may indicate a high risk of possible future accidents.

In this paper, we propose a novel uncertainty-based accident anticipation model with spatio-temporal relational learning. The model aims to learn accident-relevant cues for accident anticipation by considering both spatial and temporal relations among candidate agents. The candidate agents are a group of moving objects like vehicles and their relational features are indicative of future unobserved accidents. The spatial relations of candidate agents are learned from their spatial

This chapter is from the following publication:
"Wentao Bao, Qi Yu, and Yu Kong. Uncertainty-based traffic accident anticipation with spatio-temporal relational learning. In *Proceedings of the 28th ACM International Conference on Multimedia (ACM MM)*, 2020."

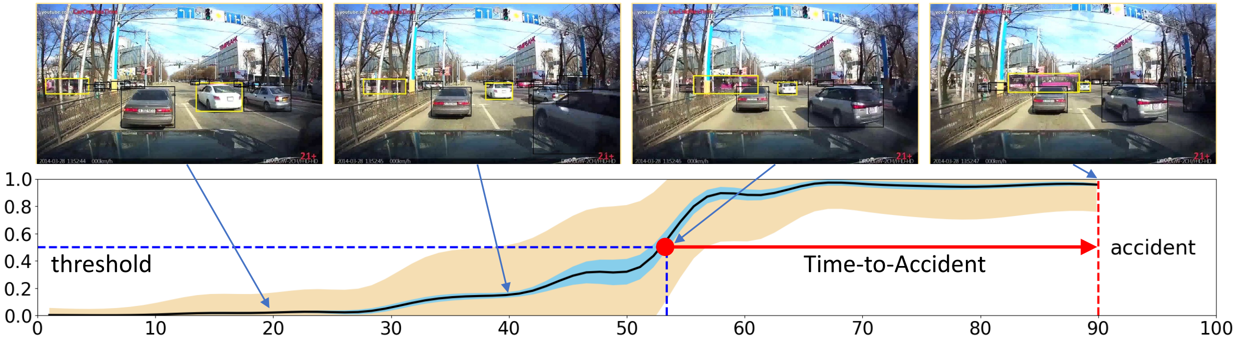


Figure 2.1 Illustration of Uncertainty-based Accident Anticipation. This paper presents a novel model to predict the probabilities (black curve) of a future accident (ranges from 90-th to 100-th frame). Our goal is to achieve early anticipation (large Time-to-Accident) giving a threshold probability (horizontal dashed line), while estimating two kinds of predictive uncertainties, i,e., aleatoric uncertainty (wheat color region) and epistemic uncertainty (blue region).

distance, visual appearance features, as well as historical visual memory. The temporal relations of agents provide learnable patterns to indicate how the agents evolve and end with an accident in the temporal context. It can be recurrently learned by updating historical memory with agent-specific features and spatial relational representation. To address the variability of the spatio-temporal relational representations, a probabilistic module is incorporated to simultaneously predict accident scores and estimate how much uncertainty when making the prediction.

As shown in Fig. 2.2, on the one hand, we propose to learn spatial relations with graph convolutional networks (GCN) [52, 145] by considering the hidden states from a recurrent neural network (RNN)) [199, 285] cell. On the other hand, we propose to build temporal relations with RNNs by considering both spatial relational and agent-specific features. The cyclic process of the coupled GCNs and RNNs could generate representative latent spatio-temporal relational features. Besides, we propose to incorporate Bayesian deep neural networks (BNNs) [58, 241] into our model to address the predictive uncertainty. With the Bayesian formulation, our derived epistemic uncertainty-based ranking loss is effective in improving the quality of the learned relational features and significantly leads to performance gain. At last, to further consider the global guidance of all hidden states in the training stage, we propose a self-attention aggregation layer as shown in Fig. 2.4, from which an auxiliary video-level loss is obtained and demonstrated beneficial to our model.

Compared with existing RNN-based methods [29, 303], our model captures not only agent-specific features but also relational features for accident anticipation. Compared with the recent approach [242] which is developed with 3D CNNs, our model is developed with GCNs and RNNs so that both spatial and temporal relations can be learned. Moreover, our method is capable of estimating the predictive uncertainty while all existing methods are deterministic.

The proposed model is evaluated on two public dashcam video datasets, i.e., DAD [29] and A3D [303], and our collected Car Crash Dataset (CCD). Experimental results show that our model can outperform existing methods on all datasets. For DAD datasets, our method can anticipate a traffic accident 3.53 seconds on average earlier before an accident happens. With the best precision setting, our model can achieve 72.22% average precision. Compared with DAD and A3D datasets, our CCD dataset includes diversified environmental annotations and accident reason descriptions, which could promote research on traffic accident reasoning.

The main contributions of this paper are summarized below:

- We propose a traffic accident anticipation model by considering both agent-specific features and their spatio-temporal relations, as well as the predictive uncertainty.
- With Bayesian formulation, the spatio-temporal relational representations can be learned with high quality by a novel uncertainty-based ranking loss.
- We propose a self-attention aggregation layer to generate video-level prediction in the training stage, which serves as global guidance and is demonstrated beneficial to our model.
- We release a new dataset containing real traffic accidents, in which diversified environmental annotations and accident reasons are provided.

2.2 Related Work

Traffic Accident Anticipation To anticipate traffic accidents that will happen in future frames, an intuitive solution is to iteratively predict the accident confidence score for each time step. Chan et al. [29] recently proposed DSA framework to leverage candidate objects appeared in each frame to represent the traffic status. They applied spatial-attention to these objects to get a weighted feature representation for each LSTM cell. Based on this work, Suzuki et al. [303] proposed an adaptive loss for early anticipation with quasi-recurrent neural networks [19]. Similar to the DSA that implements dynamic-spatial attention to focus on accident-relevant objects, Corcoran and James [46] proposed a two-stream approach to traffic risk assessment. They utilized features of candidate objects as input of a spatial stream and optical flow as input of a temporal stream, and the two-stream features are fused for the risk level classification. Instead of using dashcam videos, Shah et al. [286] proposed to use surveillance videos to anticipate traffic accidents by using the framework DSA. Different from previous works, recently Neumann and Zisserman [242] used 3D convolutional networks to predict the sufficient statistics of a mixture of 1D Gaussian distributions. In addition to using only dashcam video data, Takimoto et al. [304] proposed to incorporate physical location data to predict the occurrence of traffic accidents. Closely related to traffic accident anticipation, traffic accident detection has been recently studied by Yao et al. [368]. They proposed to detect traffic anomalies by predicting the future locations on video frames using ego-motion information. To anticipate both spatial risky regions and temporal accident occurrence, Zeng et al. [387] proposed a soft-attention RNN by considering the event agent such as the human that triggers the event.

However, existing work typically ignores the relations between accident-relevant agents which capture important cues to anticipate accidents in future frames. Besides, none of them considers the uncertainty estimation in developing their models, which is critical to safety-guaranteed systems.

Uncertainty in Sequential Modeling Uncertainty estimation is crucial to sequential relational modeling. One way is to directly formulate the latent representations of relational observations at each time step as random variables, which follow posterior distributions that can be approximated by deep neural networks. This is similar to variational auto-encoder (VAE) [143, 270]. Inspired by VAE, Chung et al. [43] proposed a variational recurrent neural network (VRNN), which formulates the hidden states of RNN as random variables and uses neural networks to approximate the posterior distributions of the variables. To further consider the relational representation of sequential data,

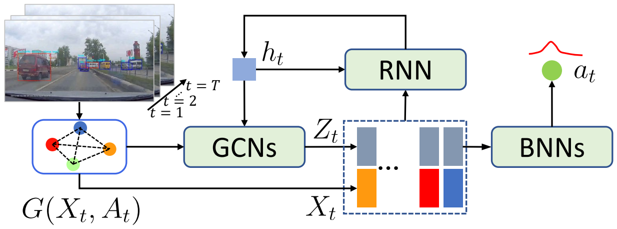


Figure 2.2 Framework of the proposed model. With graph embedded representations $G(\boldsymbol{X}_t, \boldsymbol{A}_t)$ at time step t, our model learns the latent relational representations $\boldsymbol{Z}_t$ by the cyclic process of graph convolutional networks (GCNs) and recurrent neural network (RNN) cell, and predicts the accident score $\boldsymbol{a}_t$ by Bayesian neural networks (BNNs).

Hajiramezanali et al. [103] proposed variational graph recurrent neural networks (VGRNN) for dynamic link prediction problem by combining the graph RNN and variational graph auto-encoder (VGAE) [144].

Another way to address uncertainty estimation is to formulate the weights of neural network as random variables such as Bayesian neural networks (BNNs) [58, 241]. Recently, Zhao et al. [400] proposed a Bayesian graph convolution LSTM model for skeleton-based action recognition. In this paper, we also use graph convolution and BNNs but the difference is that their method uses stochastic gradient Hamiltonian Monte Carlo (SGHMC) sampling for posterior approximation, while we use Bayes-by-backprop [18] as our approximation method. Compared with SGHMC, the Bayes-by-backprop method can be seamlessly integrated into a deep learning optimization process so that it is more flexible to handle the learning tasks with large-scale datasets, i.e., dashcam videos used in traffic accident anticipation.

2.3 Approach

Problem Setup. In this paper, the goal of accident anticipation is to predict an accident from dashcam videos before it happens. Formally, given a video with current time step t, the model is expected to predict the probability a_t that an accident event will happen in the future. Furthermore, suppose an accident will happen at time step y where $t < y$, the *Time-to-Accident* (TTA) is defined as $\tau = y - t$ when t is the first time that a_t is larger than given threshold (see Fig. 2.1). For any $t \geq y$ with a positive video that contains an accident, we define $\tau = 0$ which means the model fails to anticipate the accident. In this paper, our goal is to predict a_t and expect τ to be as large as possible for dashcam videos that contain accidents. Similar to [29], the ground truth of a_t is expressed with 2-dimensional one-hot encoding so that prediction target is $\boldsymbol{a}_t = (a_t^{(p)}, a_t^{(n)})^T$, where $a_t^{(p)}$ and $a_t^{(n)}$ represent the positive and negative predictions, respectively, meaning an accident will happen or not happen in the given video.

Framework Overview. The framework of our model is depicted in Fig. 2.2. With a dashcam video as input, a graph is constructed with detected objects and corresponding features at each time step. To learn the spatio-temporal relations of these objects, we use graph convolutional networks (GCNs) to learn the spatial relations and leverage the hidden state $\boldsymbol{h}_t$ of recurrent neural network (RNN) cell to enhance the input of the last GCN layer. Besides, the latent relational features are fused with corresponding object features as input of an RNN cell to update the hidden state at next time step. The cyclic process encourages our model to learn the latent relational features $\boldsymbol{Z}_t$ from both spatial and temporal aspects. Furthermore, we propose to use Bayesian neural network (BNN) to predict accident scores a^t so that predictive uncertainties are naturally formulated. During the training stage, we propose a self-attention aggregation (SAA, in Fig. 2.4) layer to predict video-level score, which can globally guide the learning of the proposed model.

In the following sections, each part of our model will be introduced in detail.

2.3.1 Spatio-Temporal Relational Learning

The spatio-temporal relations of traffic accident-relevant agents are informative to predict future accidents. In our model, we propose to use graph structured data to represent the observation at each time step. Then, the feature learning of spatial and temporal relations are coupled into a cyclic process.

Graph Representation. Graph representation for traffic scene has the advantages over full-frame feature embedding in that the impact of cluttered traffic background can be reduced and informative relations of traffic agents can be discovered for accident anticipation. Similar to [29, 286], we exploit object detectors [267, 23] to obtain a fixed number of candidate objects. These objects are treated as graph nodes so that a complete graph can be formed. However, the computational cost of graph convolution could be tremendous if the node features are with high dimensions.

In this paper, to construct low-dimensional but representative features for graph nodes $\boldsymbol{X}_t$, we introduce fully-connected (FC) layers to embed both the features of full-frame and candidate objects into the same low-dimensional space. Then, the frame-level and all object-level features are concatenated to enhance the feature representation capability:

$$\boldsymbol{X}_t^{(i)} = \left[\Phi\left(\boldsymbol{O}_t^{(i)}\right), \Phi\left(\boldsymbol{F}_t\right)\right], \tag{2.1}$$

where Φ denotes FC layer, $\boldsymbol{O}_t^{(i)}$ and $\boldsymbol{F}_t$ are high-dimensional features of the i-th object and corresponding frame at time t, respectively. The operator $[,]$ represents concatenation in feature dimension and is used throughout this paper for simplicity.

The graph edge at time t is expressed as an adjacent matrix $\boldsymbol{A}_t$ of a complete graph since we do not have information on which candidate object will be involved in an accident. Typically, an object with closer distance to others has higher possibility to be involved in an future accident. Therefore, the spatial distance between objects should be considered in edge weights such that we define $\boldsymbol{A}_t$ as

$$\boldsymbol{A}_t^{(ij)} = \frac{\exp\{-d(r_i, r_j)\}}{\sum_{ij}\exp\{-d(r_i, r_j)\}}, \tag{2.2}$$

where $d(r_i, r_j)$ measures the Euclidean distance between two candidate object regions r_i and r_j. By this formulation, closer distance leads to larger $\boldsymbol{A}_t^{(ij)}$. This means the two objects i and j will be applied with larger weight when we use graph convolution to learn their relational features for accident anticipation. Note that due to object occlusions, small distance defined in pixel space does not necessarily indicate close distance in physical world. It is possible to use 3D real-world distance if camera intrinsics are known. Nevertheless, the adjacency matrix defined in Eq. 2.2 has advantage to suppress the impact of irrelevant objects with significant large pixel distance to relevant objects.

Temporal Relational Learning. To build temporal relations at different time steps, RNN methods such as LSTM [113] and GRU [41] are widely adopted in existing works. However, traffic objects may not always be remained in each frame, the node features of the statically structured graph will be dynamically changing over time. Thanks to the recent graph convolutional recurrent network (GCRN) [285], it can handle the node dynamics defined over a static graph structure [103]. Therefore, we propose to adapt GCRN for temporal relational feature learning. Specifically, the hidden states $\boldsymbol{h}_t$ of RNN cell at each time step are recurrently updated by

$$\boldsymbol{h}_{t+1} = \mathrm{GCRN}\left(\left[\boldsymbol{Z}_t, \boldsymbol{X}_t\right], \boldsymbol{h}_t\right), \tag{2.3}$$

where $\boldsymbol{Z}_t$ is the relational feature generated by the last GCN layer. The feature fusion between $\boldsymbol{Z}_t$ and $\boldsymbol{X}_t$ ensures our model to make fully use of both agent-specific and relational features.

Spatial Relational Learning. To capture spatial relations of detected objects, we follow the graph convolution defined by [52, 145] for each GCN layer. In this paper, we use two stacked GCN layers and consider the hidden state $\boldsymbol{h}_t$ learned by RNNs to learn the spatial relational features:

$$\boldsymbol{Z}_t = \mathrm{GCN}\left(\left[\mathrm{GCN}\left(\boldsymbol{X}_t, \boldsymbol{A}_t\right), \boldsymbol{h}_t\right], \boldsymbol{A}_t\right). \tag{2.4}$$

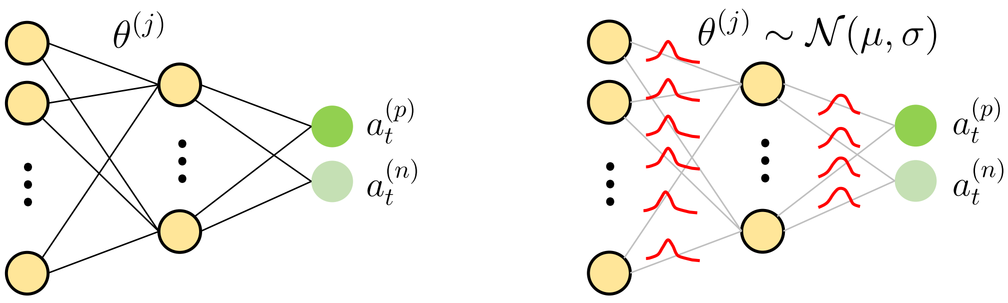
The fusion with $\boldsymbol{h}_t$ enables the latent relational representation aware of temporal contextual information. This fusion method is demonstrated to be effective to boost the performance of accident anticipation in our experiments.

2.3.2 BNNs for Accident Anticipation

To predict traffic accident score $\boldsymbol{a}_t$, a straightforward way is to utilize neural networks (NNs) as shown in Fig. 2.3a. However, the output of NNs is a point estimate which cannot address the intrinsic variability of the input relational features at each time step. Moreover, NNs could be overconfident in false model predictions when the model suffers from over-fitting problem.

To this end, we incorporate Bayesian neural networks (BNNs) [58, 241] into our framework for accident score prediction. The architecture is shown in Fig. 2.3b. The BNNs module consists of two BNN layers with latent representation $\boldsymbol{Z}_t$ given by Eq. 2.4 as input to predict accident score $\boldsymbol{a}_t$. To best of our knowledge, we are the first to incorporate BNNs into video-based traffic accident anticipation such that predictive uncertainty can be achieved. The predictive uncertainty could be utilized to not only guide the relational features learning (see Section 2.3.3), but also provide tools to interpret the model performance.

As we formulate the accident anticipation part as BNNs, the network parameters of BNNs such as weights and biases are all random variables, denoted as $\boldsymbol{\theta}$. Each entry of $\boldsymbol{\theta}$ is drawn from a Gaussian distribution determined by a mean and variance, i.e., $\boldsymbol{\theta}^{(j)} \sim \mathcal{N}(\mu, \sigma)$, in which



(a) Neural Networks

(b) Bayesian Neural Networks

Figure 2.3 Compared with NNs (Fig.. 2.3a), network parameters of BNNs (Fig. 2.3b) are sampled from Gaussian distributions so that both $\boldsymbol{a}_t$ and its uncertainty can be obtained.

$\boldsymbol{\alpha}^{(j)} = (\mu, \sigma)$ need to be learned with dataset $\mathcal{D} = (\boldsymbol{Z}_t, \boldsymbol{a}_t)$. Therefore, the likelihood of prediction can be expressed as $p(\boldsymbol{a}_t|\boldsymbol{Z}_t, \boldsymbol{\theta}) = \mathcal{N}(f(\boldsymbol{Z}_t; \boldsymbol{\theta}), \beta)$, where β is the predictive variance. However, according to the Bayesian rule, to obtain the true posterior of model parameters, i.e., $p(\boldsymbol{\theta}|\mathcal{D})$, in addition to the likelihood and prior of $\boldsymbol{\theta}$, the marginal distribution $\int p(\boldsymbol{a}_t|\boldsymbol{Z}_t, \boldsymbol{\theta})d\boldsymbol{\theta}$ is required, which is intractable since $\boldsymbol{a}_t = f(\boldsymbol{Z}_t, \boldsymbol{\theta})$ is modeled by a complex neural network. To estimate $p(\boldsymbol{\theta}|\mathcal{D})$, existing variational inference methods (VI) [96, 18, 79] could be used.

In this paper, we adopt the VI method Bayes-by-Backprop [18] to approximate $p(\boldsymbol{\theta}|\mathcal{D})$ since it can be seamlessly incorporated in standard gradient-based optimization to learn from large-scale video dataset. According to [18], the variational approximation aims to minimize the following objective:

$$\arg\min_{\boldsymbol{\alpha}} \sum_{i=1}^{J} \log q\left(\boldsymbol{\theta}_i|\boldsymbol{\alpha}\right) - \log p\left(\boldsymbol{\theta}_i\right) - \log\left(p\left(\mathcal{D}|\boldsymbol{\theta}_i\right)\right), \tag{2.5}$$

where J is the number of Monte Carlo samplings for $\boldsymbol{\theta}$. The first term $q\left(\boldsymbol{\theta}_i|\boldsymbol{\alpha}\right)$ is the variational posterior distribution parameterized by $\boldsymbol{\alpha}$. The distribution parameters $\boldsymbol{\alpha}$ can be efficiently learned by using reparameterization trick and standard gradient descent methods [18]. We denote this loss term as $\mathcal{L}_{VPOS}$. The second term $p(\boldsymbol{\theta}_i)$ is the prior distribution of $\boldsymbol{\theta}$. It is typically modeled with a spike-and-slab distribution, i.e., a mixture of two Gaussian density functions with zero means but different variances. We denote this loss term as $\mathcal{L}_{PRI}$.

The third term in Eq. 2.5 is the negative log-likelihood of model predictions. Since minimizing this term is equivalent to minimizing the mean squared error (MSE), in this paper, we propose to use exponential binary cross entropy to achieve this objective:

$$\mathcal{L}_{EXP} = \sum_{t=1}^{T} -e^{-\max\left(0, \frac{y-t}{f}\right)} \log a_t^{(p)} + \sum_{t=1}^{T} -\log\left(1 - a_t^{(n)}\right), \tag{2.6}$$

where f is the constant frame rate for the given video, and y is the beginning time of an accident provided by training set. The exponential weighted factor applies larger penalty to the time step that is closer to the beginning time of an accident.

2.3.3 Uncertainty-guided Ranking Loss

With the Bayesian formulation for accident anticipation, we can perform multiple forward passes at each time step such that an assembled prediction could be obtained by taking the average of these multiple outputs. Furthermore, as suggested by [136], the predictive uncertainty (variance) can be decomposed as **aleatoric** uncertainty and **epistemic** uncertainty [163, 289]:

$$\boldsymbol{U}_t = \underbrace{\frac{1}{M}\sum_{i=1}^{M}\left[\operatorname{diag}\left(\hat{\boldsymbol{a}}_i\right) - \hat{\boldsymbol{a}}_i\hat{\boldsymbol{a}}_i^T\right]}_{\text{Aleatoric Uncertainty}(\boldsymbol{U}_t^{alt})} + \underbrace{\frac{1}{M}\sum_{i=1}^{M}\left(\hat{\boldsymbol{a}}_i - \bar{\boldsymbol{a}}\right)\left(\hat{\boldsymbol{a}}_i - \bar{\boldsymbol{a}}\right)^T}_{\text{Epistemic Uncertainty}(\boldsymbol{U}_t^{ept})}, \tag{2.7}$$

where $\bar{\boldsymbol{a}} = \frac{1}{M}\sum_{i=1}^{M}\hat{\boldsymbol{a}}_i$ and $\hat{\boldsymbol{a}}_i = (\hat{a}_t^{(n)}, \hat{a}_t^{(p)})_i^T$. They are the predictions of the i-th forward pass at time step t with total M forward passes. The first term in Eq. 2.7 is the aleatoric uncertainty, which measures the input variability (noise) of BNNs. In our model, the aleatoric uncertainty serves as an indicator to the quality of the learned relational features from GCNs and RNNs.

The second term in Eq. 2.7 is epistemic uncertainty which is determined by the BNNs model itself. Inspired by Ma et al. [216], ideally the epistemic uncertainties of sequential predictions should be monotonically decreasing, since as more frames the model observes, the more confident of the learned model (smaller epistemic uncertainty) will be. Therefore, we propose a novel ranking loss:

$$\mathcal{L}_{RANK} = \max\left(0, \operatorname{trace}\left(\boldsymbol{U}_t^{ept} - \boldsymbol{U}_{t-1}^{ept}\right)\right), \tag{2.8}$$

where $\boldsymbol{U}_{t-1}^{ept}$ and $\boldsymbol{U}_t^{ept}$ are epistemic uncertainties of successive frames $t-1$ and t defined in Eq. 2.7. Note that $\boldsymbol{U}_t$ as well as the two terms in Eq. 2.7 are matrices with size 2×2, therefore in practice we propose to use matrix trace to quantify the uncertainties, which is similar to the method adopted

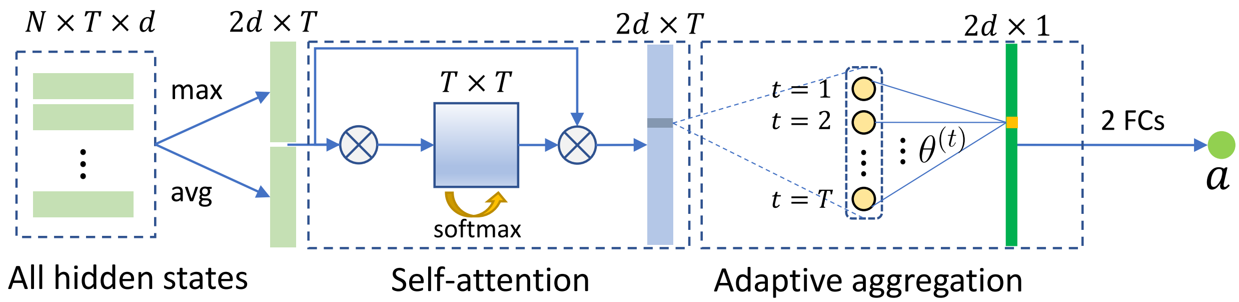


Figure 2.4 **SAA Layer.** First, all $N \times T$ hidden states are gathered and pooled by max-avg concatenation. Then, the simplified self-attention and adaptive aggregation are proposed to predict video-level accident score $\boldsymbol{a}$.

in [289]. Our proposed ranking loss aims to apply penalty to the predictions that do not follow the epistemic uncertainty ranking rule.

For aleatoric uncertainty $\boldsymbol{U}_t^{alt}$, it is not necessary to satisfy the monotonic ranking requirement since the noise ratio of accumulated data in video sequence is intrinsically not monotonic.

2.3.4 Temporal Self-Attention Aggregation

Recurrent network can naturally build temporal relations of observations. However, the drawback of RNNs is that inaccurate hidden states in early temporal stages could be accumulated in iterative procedure and mislead the model to give false predictions in latter temporal stages. Besides, the hidden states in different time steps should be adaptive to anticipate the occurrence of a future accident.

To this end, motivated by recent self-attention design [328], we propose a self-attention aggregation (SAA) layer in the training stage by adaptively aggregating hidden states of all time steps. Then, we use the aggregated representation to predict video-level accident score. The architecture of SAA layer is shown in Fig. 2.4.

Specifically, we first aggregate hidden states of N individual objects at each time step by applying the concatenation between mean- and max-pooling results. Then, the self-attention [328] is adapted to weigh the representation of all T time steps. In this module, the embedding layers are not used. Lastly, instead of using simple average pooling, we introduce an FC layer with T learnable parameters to adaptively aggregate the T temporal hidden states. The aggregated video-level representation is used to predict the video-level accident score $\boldsymbol{a}$ by two FC layers. This

Table 2.1 Comparison between CCD dataset and existing datasets. Information about DAD and A3D is obtained from their released sources. *Temp* means the temporal accident time annotations. *RandABT* means accidents beginning times are randomly placed. *EgoIn* means the ego-vehicles are involved in accidents. *Light* indicates the data is collected in day or night. *Weather* includes rainy, snowy, and sunny conditions. *Bbox* means the bounding boxes tracklets for accident participants. *Reasons* contains multiple possible reasons for each accident participant.

Datasets	# Videos	# Pos	Hours	Temp	RandABT	EgoIn	Light	Weather	Bbox	Reasons
DAD [29]	1,750	620	2.43 h	✓						
A3D [368]	1,500	1,500	3.56 h	✓	✓	✓				
Ours (CCD)	4,500	1,500	6.25 h	✓	✓	✓	✓	✓	✓	✓

network is trained with binary cross-entropy (BCE) loss:

$$\mathcal{L}_{BCE} = -\log a^{(p)} - \log\left(1 - a^{(n)}\right), \tag{2.9}$$

where $\boldsymbol{a} = (a^{(n)}, a^{(p)})^T$ normalized by softmax function. This auxiliary learning objective encourages the model to learn better hidden states even though SAA layer is not used in testing stage.

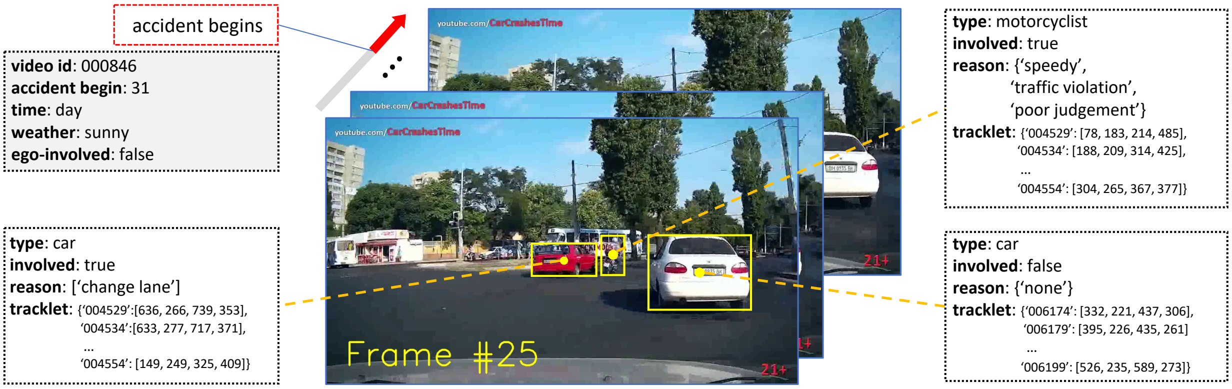
Finally, the complete learning objective of our model is to minimize the following weighted loss:

$$\mathcal{L} = \mathcal{L}_{EXP} + w_1 \cdot (\mathcal{L}_{VPOS} - \mathcal{L}_{PRI}) + w_2 \cdot \mathcal{L}_{RANK} + w_3 \cdot \mathcal{L}_{BCE} \tag{2.10}$$

where the $\mathcal{L}_{VPOS}$ and $\mathcal{L}_{PRI}$ are loss functions of variational posterior and prior. The constants w_1, w_2 and w_3 are set to 0.001, 10 and 10, respectively, to balance the magnitudes of these loss terms. The second penalty term $(\mathcal{L}_{VPOS} - \mathcal{L}_{PRI})$ is also termed as complexity loss and has similar effect to overcome over-fitting problem. The third penalty term $\mathcal{L}_{BCE}$ introduces video-level classification guidance while the fourth term $\mathcal{L}_{RANK}$ brings uncertainty ranking guidance to train our model.

2.4 Experiments

In this section, we evaluate our model on three real-world datasets, including our collected Car Crash Dataset (CCD) and two public datasets, i.e., Dashcam Accident Dataset (DAD) [29] and AnAn Accident Detection (A3D) dataset [368]. State-of-the-art methods are compared and ablation studies are performed to validate our model.



video id: 000846
accident begin: 31
time: day
weather: sunny
ego-involved: false

type: car
involved: true
reason: [‘change lane’]
tracklet: {‘004529’:[636, 266, 739, 353],
‘004534’:[633, 277, 717, 371],
…
‘004554’: [149, 249, 325, 409]}

type: motorcyclist
involved: true
reason: {‘speedy’,
‘traffic violation’,
‘poor judgement’}
tracklet: {‘004529’: [78, 183, 214, 485],
‘004534’: [188, 209, 314, 425],
…
‘004554’: [304, 265, 367, 377]}

type: car
involved: false
reason: {‘none’}
tracklet: {‘006174’: [332, 221, 437, 306],
‘006179’: [395, 226, 435, 261]
…
‘006199’: [526, 235, 589, 273]}

Figure 2.5 Annotation samples of our Car Crash Dataset (CCD). The gray box on top-left contains video-level annotations, while the other three white boxes provide instance-level annotations.

2.4.1 Datasets

CCD dataset[1]. In this paper, we collect a challenging Car Crash Dataset (CCD) for accident anticipation. We ask annotators to label YouTube accident videos with temporal annotations, diversified environmental attributes (day/night, snowy/rainy/good weather conditions), whether ego-vehicles involved, accident participants, and accident reason descriptions. For temporal annotations, the accident beginning time is labeled at the time when a car crash actually happens. To get trimmed videos with 5 seconds long, the accident beginning times are further randomly placed in last 2 seconds, generating 1,500 traffic accident video clips. We also collected 3,000 normal dashcam videos from BDD100K [375] as negative samples. The dataset is divided into 3,600 training videos and 900 testing videos. Examples are shown in Fig. 2.5 and comparison details with existing datasets are reported in Table 2.1. Compared with DAD [29] and A3D [368], our CCD is larger with diversified annotations.

DAD dataset. DAD [29] contains dashcam videos collected in six cities in Taiwan. It provides 620 accident videos and 1,130 normal videos. Each video is trimmed and sampled into 100 frames with totally 5 seconds long. For accident videos, accidents are placed in the last 10 frames. The dataset has been divided into 1,284 training videos (455 positives and 829 negatives) and 466 testing videos (165 positives and 301 negatives).

A3D dataset. A3D [368] is also a dashcam accident video dataset. It contains 1,500 positive

[1] CCD dataset is available at: https://github.com/Cogito2012/CarCrashDataset

traffic accident videos. In this paper, we only keep the 587 videos in which ego-vehicles are not involved in accidents. We sampled each A3D video with 20 fps to get 100 frames in total and placed the beginning time of each accident at the last 20 frames similar to DAD. The dataset is divided into 80% training set and 20% testing set.

2.4.2 Evaluation Metrics

Average Precision. This metric evaluates the **correctness** of identifying an accident from a video. Following the same definition as [29], at time step t, if $a_t^{(p)}$ is larger than a threshold, then the prediction at frame t is positive to contain an accident, otherwise it is negative. For accident videos, all frames are labeled with ones (positive), otherwise the labels are zeros (negative). By this way, the precision, recall, as well as the derived Average Precision (AP) can be adopted to evaluate models.

Time-to-Accident. This metric evaluates the **earliness** of accident anticipation based on positive predictions. For a range of threshold values, multiple TTA results as well as corresponding recall rates can be obtained. Then, we use mTTA and TTA@0.8 to evaluate the earliness, where mTTA is the average of all TTA values and TTA@0.8 is the TTA value when recall rate is 80%. Note that if a large portion of predictions are false positives, very high TTA results can still be achieved while corresponding AP would be low. That means the model is overfitting on accident video and may give positive predictions for arbitrary input. Therefore, except for fair comparison with existing methods, we mainly report TTA metrics when the highest AP is achieved, because it is meaningless to obtain high TTA if high AP cannot be guaranteed.

Predictive Uncertainty. Based on Eq. 2.7, we introduce to use the mean aleatoric uncertainty (mAU) and mean epistemic uncertainty (mEU) to evaluate the predictive uncertainties.

2.4.3 Implementation Details

We implement our model with PyTorch [255]. For DAD dataset, we use the candidate objects and corresponding features provided by DSA[2] for fair comparison. For the experiments on A3D and CCD datasets, we use the public detection codebase MMDetection[3] to train Cascade R-CNN [23]

[2] DSA: https://github.com/smallcorgi/Anticipating-Accidents

[3] MMDetection: https://github.com/open-mmlab/mmdetection

Table 2.2 Evaluation results on DAD, A3D, and CCD datasets. Results of baselines on DAD are obtained from [387] and [303]. The notation “–" means the metric is not applicable.

Datasets	Methods	mTTA(s)	AP(%)	mAU	mEU
DAD [29]	DSA [29]	1.34	48.1	–	–
	L-RAI [387]	3.01	51.4	–	–
	adaLEA [303]	3.43	52.3	–	–
	Ours	**3.53**	**53.7**	**0.0294**	**0.0011**
A3D [368]	DSA [29]	4.41	93.4	–	–
	Ours	**4.92**	**94.4**	**0.0095**	**0.0023**
CCD	DSA [29]	4.52	**99.6**	–	–
	Ours	**4.74**	99.5	**0.0137**	**0.0001**

with ResNeXt-101 [351] backbone and FPN [196] neck as our object detector on KITTI 2D detection dataset [84]. The trained detector is used to detect candidate objects and then extract VGG-16 features of full-frame and all objects. As suggested by Bayes-by-backprop [18], we set the number of forward passes M to 2 in training stage and 10 for testing stage. For the hyper-parameters of prior distribution, we set the mixture ratio π to 0.5 and the variances of the two Gaussian distributions σ_1 to 1 and σ_2 to $\exp(-6)$. The dimensions of both hidden state of RNN and output of GCNs are set to 256. In the training stage, we set batch size to 10 and initial learning rate to 0.0005 with *ReduceLROnPlateau* as learning rate scheduler. The model is trained by Adam optimizer for totally 70 training epochs.

2.4.4 Performance Evaluation

Compare with State-of-the-art Methods. Existing methods [29, 387, 303] are compared and results are reported in Table 2.2. For fair comparison, we use the model at the last training epoch for evaluation on DAD datasets. Nevertheless, the trained model with best AP is kept for evaluation on other two datasets since high AP is important to suppress impact of false positives on TTA evaluation. Note that these two metrics currently are only applicable to our model, since we are the first to introduce uncertainty formulation for accident anticipation.

From Table 2.2, our model on DAD dataset achieves the best mTTA which means the model anticipates on average 3.53 seconds earlier before an accident happens, while keeping competitive AP performance at 53.7% compared with L-RAI and adaLEA. Note that the video lengths of the

Table 2.3 TTA with different recall rates on DAD dataset.

Recall	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
DSA [29]	0.28	0.50	0.73	0.87	0.92	1.02	1.24	1.35	**2.28**
Ours	**0.59**	**0.75**	**0.84**	**0.96**	**1.07**	**1.16**	**1.33**	**1.56**	1.99

three datasets are all 5 seconds, our high performance on A3D and CCD demonstrate that our model is easier to be trained on different datasets. This can be explained by the mAU results due to their consistence with TTA evaluation results in Table 2.2. The low mAU values on A3D and DAD datasets reveal that our model has learned relational representations with high quality on these datasets.

We further report TTA results with different recall rates from 10% to 90% in Table 2.3. It shows that our model outperform DSA in most of recall rate requirements. For recall rates larger than 80%, our method performs poorly compared with DSA. However, high recall rate may also lead to too much false alarm so that AP cannot be guaranteed to be high. This finding also supports our motivation to use the trained model with best AP for evaluation.

Visualization We visualized accident anticipation results with samples in DAD dataset (see Fig. 2.6). The uncertainty regions indicate that in both early and late stages, the model is quite confident on prediction (low uncertainties), while in the middle stage when accident scores start are increasing, the model is uncertain to give predictions. Note that the predicted epistemic uncertainty (blue region) is not necessary to be monotonically decreasing since we only use Eq. 2.8 as training regularizer rather than strict guarantee on predictions. The results are with good interpretability, in that driving system is typically quite sure about the accident risk level when the self-driving car is far from or almost being involved in an accident, while it is uncertain about it when accumulated accident cues are insufficient to make decision.

2.4.5 Ablation Study

In this section, to validate the effectiveness of the several main components, the following components are replaced or removed, and compared with our model based on best AP setting. (1) **BNNs**: The BNNs are replaced with vanilla FC layers. Note that in this case, $\mathcal{L}_{VPOS} - \mathcal{L}_{PRI}$ and

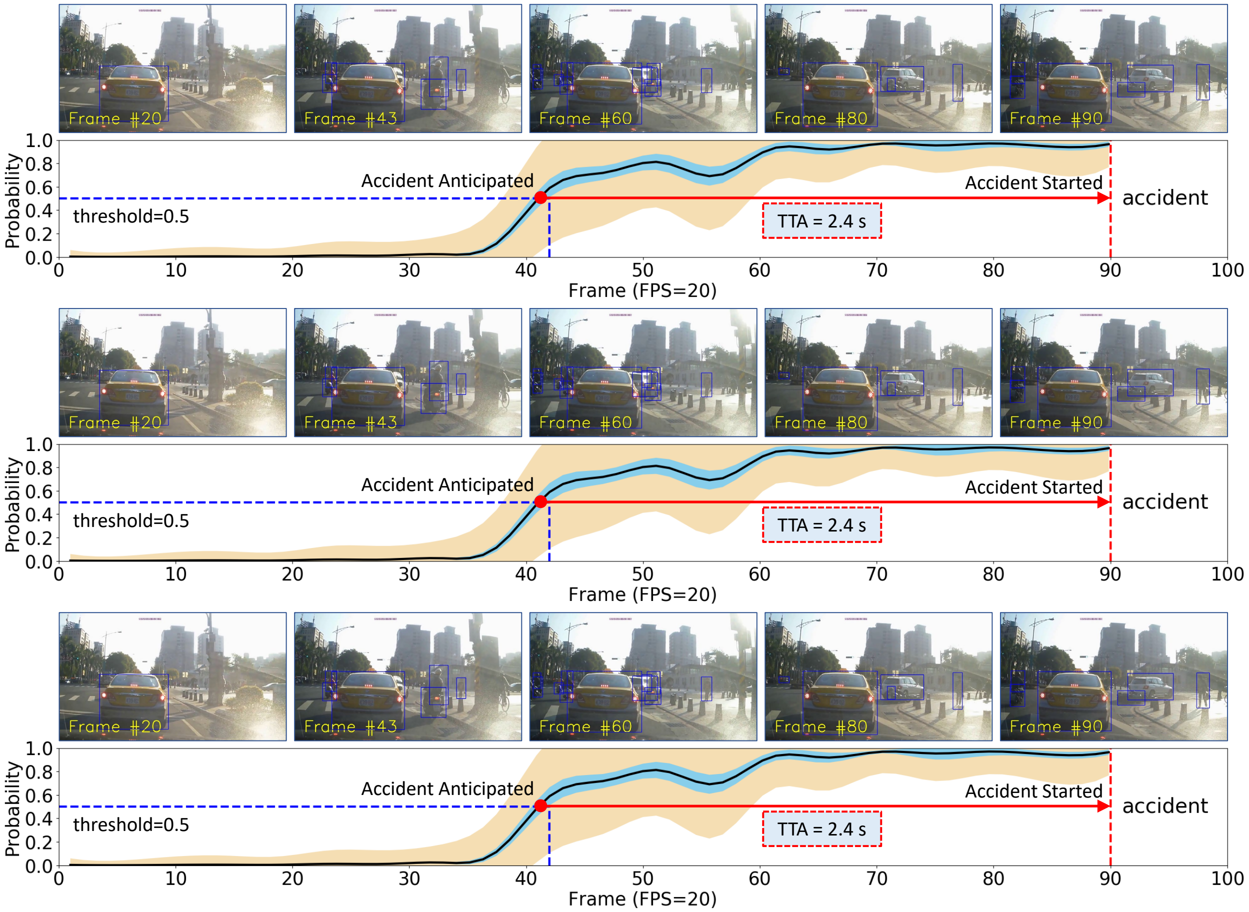


Figure 2.6 Examples of our predictions on DAD datasets. The red curves indicate smoothed accident scores as observed frames increase. The ground truths (beginning time of an accident) are labeled at the 90-th frame. We plot one time of squared epistemic (blue region) and aleatoric uncertainties (wheat color region). The horizontal line indicates the probability threshold 0.5.

our proposed ranking loss $\mathcal{L}_{RANK}$ in Eq. 2.10, as well as mAU are not applicable. (2) **SAA**: The SAA layer is removed so that $\mathcal{L}_{BCE}$ in Eq. 2.10 is not used. (3) **GCN**: We replace GCNs with vanilla FC layers in Eq. 2.3 and Eq. 2.4. (4) **Fusion**: For this variant, the fusion in Eq. 2.3 and Eq. 2.4 are removed such that only $\boldsymbol{Z}_t$ and $\text{GCN}(\boldsymbol{X}_t, \boldsymbol{A}_t)$ are used, respectively. (5) **RankLoss**: The epistemic uncertainty-based ranking loss is removed so that $\mathcal{L}_{RANK}$ in Eq. 2.10 is not applicable. Results are shown in Table 2.4.

We can clearly see that the uncertainty-based ranking loss contributes most to our model by comparing variant (2)(6) with (1), with about 7.6% performance gain. Though the BNNs module leads to small performance gain, we attribute the benefit of BNNs to its derived uncertainty

Table 2.4 Ablation studies results on DAD dataset.

Variants	BNNs	SAA	GCN	Fusion	RankLoss	AP(%)	mAU
(1)	✓	✓	✓	✓	✓	**72.22**	**0.0731**
(2)		✓	✓	✓		70.38	–
(3)	✓		✓	✓	✓	67.34	0.1150
(4)	✓	✓		✓	✓	67.10	0.1250
(5)	✓	✓	✓		✓	65.50	0.1172
(6)	✓	✓	✓	✓		64.60	0.0950

Table 2.5 Model size comparison. Our model variants (2), (4), and (5) are included for comparison. Unit M means a million.

Methods	DSA	Ours	v(2)	v(4)	v(5)
# Params. (M)	4.40	1.97	1.66	1.97	1.90

ranking loss as well as the interpretable results. Furthermore, the lowest mAU and highest AP for variant (1) demonstrate that the learned relational features are of the highest quality (smallest uncertainty) compared with other variants. The results of variant (3) validate the effectiveness of our self-attention aggregation (SAA) layer, while the results of variant (4) validate the superiority GCN over naive FC layers. The results of variant (5) show that the feature fusion between GCN outputs and hidden states, and the fusion between relational features and agent-specific features are important to accident anticipation, leading to approximately 7% performance gain.

Model Size Comparison. The number of network parameters are counted and reported in Table 2.5. It shows that the proposed model is much light-weighted than DSA, and only slightly increases the model size when compared with other variants of our model.

2.5 Conclusion

In this paper, we propose an uncertainty-based traffic accident anticipation with spatio-temporal relational learning. Our model can handle the challenges of relational feature learning and anticipation uncertainty from video data. Moreover, the introduced Bayesian formulation not only significantly boosts anticipation performance by using the uncertainty-based ranking loss, but also provides interpretation on predictive uncertainty. In addition, we release a CCD dataset for accident anticipation which contains rich environmental attributes and accident reason annotations.

CHAPTER 3

DEEP REINFORCED EXPLAINABLE ACCIDENT ANTICIPATION

3.1 Introduction

With increasing demand for autonomous driving, anticipating possible future accidents is becoming the central consideration to guarantee a safe driving strategy [29, 303, 315]. Given a dashcam video, an accident anticipation model aims to tell the driving system if and when a traffic accident will occur in the near future. Despite remarkable advances in visual perception [55, 84, 106], the decision-making of driving control has long been studied in isolation with vision perception research for the autonomous driving scenario [141, 277]. We target at bridging this gap by investigating a key research question: *where do drivers look when predicting possible future accidents?* This will lead to a visually explainable model that associates low-level visual attention and high-level accident anticipation.

The traffic accident anticipation is far from being solved due to the following challenges. First, the visual cues of a future accident are vital to training a discriminative model but in practice, they are difficult to be captured from the limited and noisy video data before the accident occurs. Previous works take advantage of object detection and learn the accident visual cues by either soft attention in [29] or graph relational learning in [315]. In this paper, we propose to explicitly learn the visual attention behavior to address *where to look* such that accident-risky regions can be localized.

Second, it is intrinsically a trade-off between an *early* decision and a *correct* decision since the earlier to anticipate an accident, the harder to make the decision right due to fewer accident-relevant cues. Existing works [29, 315] simply address the trade-off by training supervised deep learning models with an exponentially weighted classification loss. In this paper, we address this trade-off by formulating the task as a Markov Decision Process (MDP), where exploration and exploitation

This chapter is adapted from the following publication:
"Wentao Bao, Qi Yu, and Yu Kong. DRIVE: Deep Reinforced accident anticipation with visual explanation. In *International Conference on Computer Vision (ICCV)*, 2021."

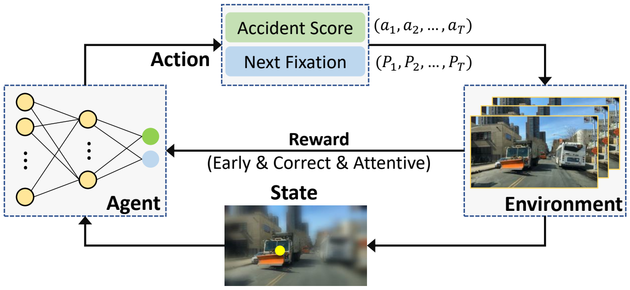

Accident Score
$(a_1, a_2, \ldots, a_T)$
Action
Next Fixation
$(P_1, P_2, \ldots, P_T)$
Reward
(Early & Correct & Attentive)
State
Agent
Environment

Figure 3.1 **The Markov decision process of the DRIVE model.** The neural network agent (left) learns to *exploit* visual attentive state (bottom) to predict the actions including the accident score and the next fixation (top), which in return *explore* the driving environment (right) to maximize the total reward (middle).

can be dynamically balanced in a driving environment. In the context of accident anticipation, the MDP model aims to *exploit* the immediate visual cues for accident anticipation and also *explore* more possibilities of accident scoring and attention allocation.

Our proposed DRIVE model is illustrated with the MDP perspective in Fig. 3.1. The DRIVE model simultaneously learns the policies of accident anticipation and fixation prediction based on a deep reinforcement learning (DRL) algorithm. At each time step, the agent takes actions to predict the occurrence probability of a future accident, as well as the fixation point indicating where drivers will look in the next time step. Our environmental model dynamically provides the observation state by considering both the bottom-up and top-down visual attention, which is recurrently modulated by the actions from the previous time step. We develop a novel dense anticipation reward to encourage early and accurate prediction, as well as a sparse fixation reward to enable visual explanation. Moreover, to effectively train the DRIVE model on real-world datasets, substantial improvements are made based on the DRL algorithm SAC [101]. Our method is demonstrated to be effective on the DADA-2000 dataset [68], and can be easily extended to the DAD dataset [29] without fixation annotations.

The proposed approach differs from existing works [29, 303, 46, 315] that are formulated within the supervised learning (SL) framework. The proposed DRL-based solution is fundamentally superior to SL in that the DRL could utilize immediate observations to achieve a long-term goal, i.e., making early decision for anticipating future accidents. Moreover, according to [142], our method

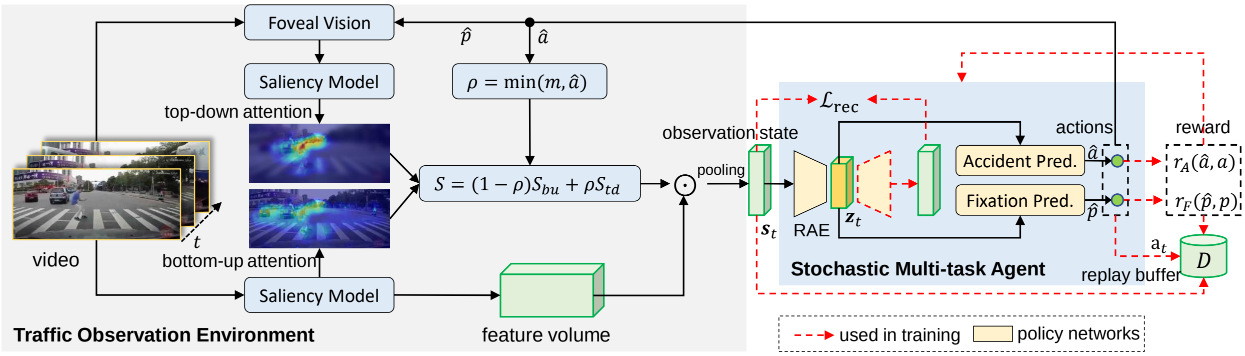


Figure 3.2 **The DRIVE Model.** At each time step t, the traffic observation environment model (left part) acquires visual attention from bottom-up and top-down pathways, generating an observation state $\mathbf{s}_t$ by the dynamic attention fusion (equation box) and feature pooling. The stochastic multi-task agent (right part) takes $\mathbf{s}_t$ as input to predict the actions $\mathbf{a}_t$ which includes the accident score $\hat{a}$ and the next fixation point $\hat{p}$. All the states, actions, and rewards are collected in the replay buffer $\mathcal{D}$ to train the two policy networks of the agent.

is introspectively explainable as compared to [29, 46], which simply provide rationalizations (post-hoc explanation), since we explicitly formulate drivers' visual attention during model learning. Our experimental results also validate that the learned visual attention serves as the causality of the outcome from the agent. The main contributions are threefold:

- The DRIVE model is proposed for traffic accident anticipation from dashcam videos based on deep reinforcement learning (DRL).

- The DRIVE model is visually explainable by explicitly simulating the human visual attention within a unified DRL framework.

- The proposed dense anticipation reward and sparse fixation reward are effective in training the model by our improved DRL training algorithm.

3.2 Related Work

Traffic Accident Anticipation. Different from recent works on accident detection [368, 118], accident recognition [373], and early action/activity prediction [35, 149], the accident anticipation problem is more challenging as the model needs to make an early decision before the accident occurs. The accident anticipation task was first formally proposed by Chan et al. [29], in which they proposed a DSA-RNN method to solve the accident anticipation problem. This method is based on object detection and dynamic soft-attention on each time step and uses LSTM to sequentially predict the accident score. In [303], an adaptive loss for early accident anticipation was introduced. Based on these works, Fatima et al. [71] proposed a feature aggregation method for LSTM-based sequential accident score prediction. Inspired by the success of the two-stream design, Corcoran et al. [46] adopted RGB video and optical flow for accident anticipation. Neumann et al. [242] formulated the temporal accident scores as a mixture of Gaussian distribution and proposed to use 3D-CNN to predict the sufficient statistics of the distribution. Recently, Bao et al. [315] proposed to use GCN and Bayesian deep learning for traffic accident anticipation. In addition to the dashcam video used in these works, Shah et al. [286] utilized surveillance videos to anticipate traffic accidents. Zeng et al. [387] recently proposed to anticipate failing accidents by localizing risky regions within an RNN framework.

By investigating these works, we found that they typically adopted recurrent neural networks or 3D convolutional networks as the model architecture. However, their supervised learning (SL) design requires large amounts of annotated training data. In terms of explainability, [29, 46] only give post-hoc bounding box explanations, which are essentially rationalizations rather than introspective explanations.

RL-based Visual Attention. Visual attention has been studied for several decades and it has been widely modeled as a Markov process [189, 165]. The earlier work [245] utilized the actor-critic RL algorithm on top-down attention modeling. Mnih et al. [228] developed an RL-based recurrent visual attention model for image classification. Jiang et al. [127] used the Least-Squares Policy Iteration for visual fixation prediction. Recent works such as [355] and [115] implemented deep RL algorithms for 360° video-based human head movement prediction. In addition to RL methods, inverse reinforcement learning (IRL) algorithms take the advantages of expert demonstrations to train policy networks for task objectives, and a recent work [367] showed that IRL can be leveraged to predict goal-directed human attention. Zhang et al. [394] proposed an imitation learning framework by using human fixations to learn a policy network for Atari games. In this paper, different from these works, we integrate visual attention and traffic accident anticipation into a unified RL framework in a real-world environment.

Explainable Self-driving. For self-driving applications, it is important to provide explainable decision making so that the self-driving system can be trusted by humans. Similar to our work, recently Xia et al. [349] proposed to use the foveal vision mechanism to model human visual attention for driving speed prediction. Kim et al. [141] used the visual attention model and causal filtering to visually explain the predicted steering control, i.e., steering angle and speed. Based on this work, [142] further proposed to combine both visual attention and textual description for self-driving behavior explanation. Though there are existing works investigating the visual attention of drivers in traffic scenario [55, 350, 3], few of them simultaneously formulate the up-stream visual attention and down-stream accident anticipation into a unified learnable model. Inspired by these works, in this paper we propose that the traffic accident anticipation can be visually explained by explicitly modeling the visual attention behavior of ego-vehicle drivers.

3.3 Approach

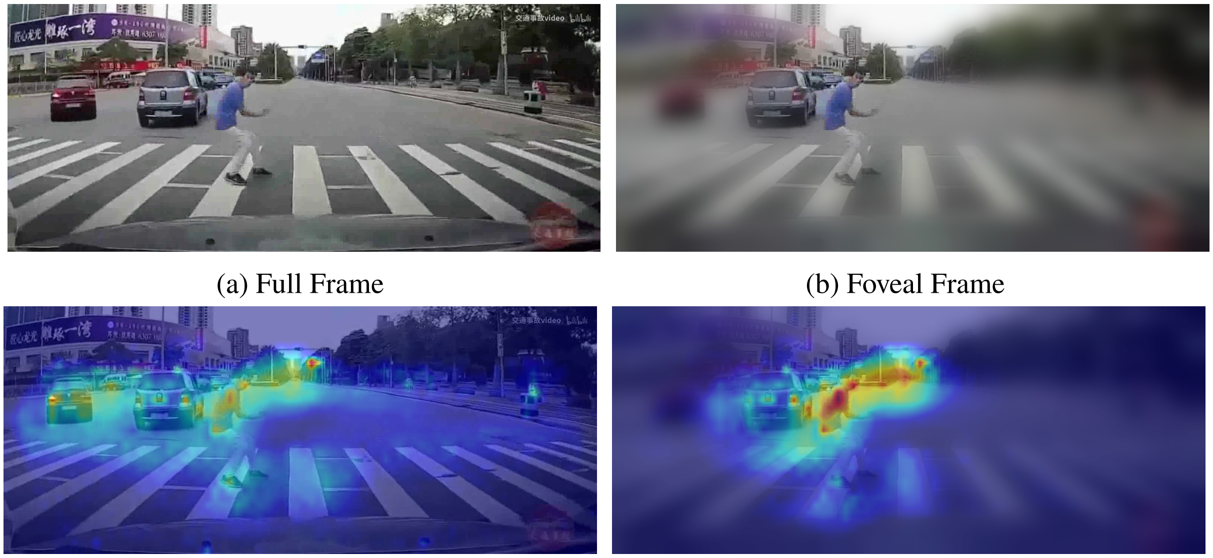
Framework Overview. Fig. 3.2 illustrates the framework of the DRIVE model. Given a dashcam video as input, the stochastic multi-task agent (right part) recurrently outputs the accident score $\hat{a}$ and the next fixation $\hat{p}$ at each time step based on the observation state from the environment (left part). In particular, the environment is built by considering the bottom-up and top-down attention of the dashcam video frames, while the agent consists of a shared state auto-encoder and two parallel prediction branches. The two actions $\hat{a}$ and $\hat{p}$ are guided by the reward r_A and r_F respectively to encourage earliness, correctness, and attentiveness. During inference, the DRIVE model simultaneously observes the driving environment by visual attention allocation to risky regions and predicts the occurrence probability of a future accident by the trained agent.

Problem Setup. In this paper, we follow the task setting in the existing literature [29, 315]. A traffic accident anticipation model aims to predict a frame-level accident score a^t that indicates the probability of the accident occurrence in the future. To evaluate the performance, *Time-to-Accident* (TTA) $tta = \max(0, t_a - t)$ is used to evaluate *earliness*, where t_a is the actual beginning time of an accident and t is the first point in time when the predicted score is higher than a threshold a_0, i.e., $a^t > a_0$. A larger tta indicates the earlier time the model can anticipate the traffic accident. Besides, binary classification and saliency evaluation metrics are adopted to evaluate the *correctness* and *attentiveness*.

Inspired by the natural decision-making process of human drivers, i.e., observe and anticipate, we formulate the traffic accident anticipation and fixation prediction tasks as a unified Markov Decision Process (MDP). Formally, let a tuple $(\mathcal{S}, \mathcal{A}, P, R, L, \gamma)$ represent a discounted MDP with finite horizon (video length) L, where $\mathcal{S}$ and $\mathcal{A}$ are spaces of action and state, R defines the reward for each state-action pair, and $\gamma \in (0, 1]$ is a discount factor. In this paper, the action $\mathbf{a}_t$ in the action space $\mathcal{A}$ consists of accident score a^t and the next fixation point $p^t = (x^{t+1}, y^{t+1})^T$ defined in the image domain such that $\mathbf{a}_t = (a^t, x^{t+1}, y^{t+1})^T$. The state $\mathbf{s}_t$ is shared with the two kinds of actions. The state representation and action policy will be introduced in Sec. 3.3.1 and 3.3.2, respectively. Note that P defines the state transition model, i.e., $P(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{a}_t)$. In our method, the state transition P is achieved by the fixation prediction module (Eq. (3.3)) and the environment observation module (Eq. (3.1) and 3.2). In Sec. 3.3.3 and 3.3.4, the reward design and training algorithm will be discussed, respectively.

3.3.1 Traffic Observation Environment

To anticipate a traffic accident, the observation state needs to be discriminative to distinguish accident-relevant cues from the cluttered traffic scene. In this paper, we are inspired by the perception mechanism of the human visual system. It is widely acknowledged that visual perception is dependent on two distinct types of attention procedure, i.e., bottom-up attention and top-down attention [44]. The bottom-up attention is determined by the salient visual stimuli from the sensory input, while the top-down attention is driven by the browsing task to achieve a long-term cognitive goal. These two mechanisms have been demonstrated to be successful in modeling the visual attention of drivers in traffic scene [56, 57]. For traffic accident anticipation, observing the entire scene is inefficient while the attention mechanism can be utilized to capture the discriminative accident-relevant cues for better traffic observation state modeling.



(a) Full Frame

(b) Foveal Frame

(c) Bottom-up Attention

(d) Top-down Attention

Figure 3.3 **Examples of Foveation and Attention.** With a saliency model, the full frame I (Fig. 3.3a) and its foveated frame $F(I, p)$ (Fig.3.3b) are used to generate bottom-up attention $G(I)$ (Fig. 3.3c) and top-down attention $G(F(I, p))$ (Fig. 3.3d), respectively.

Traffic Attention Modeling. Given the observation of current time step I^t, the bottom-up attention $S_{bu}^t \in \mathbb{R}^{H \times W}$ is simulated by a saliency prediction module G, i.e., $S_{bu}^t = G(I^t)$, where G is instantiated by recent deep convolutional neural networks (CNNs) such as [47, 225] so that feature extraction can be shared with the saliency module. As the saliency module is not updated by the actions, S_{bu}^t is only determined by the appearance of video frames.

To simulate top-down attention $S_{td}^t \in \mathbb{R}^{H \times W}$, we propose an auxiliary task to predict the fixation point $p^t \in \mathbb{R}^2$, which will dynamically guide the visual attention allocation to the risky region at each time step. Specifically, S_{td}^t is computed by applying a foveal vision module F before feeding into the saliency module, i.e., $S_{td}^t = G(F(I^t, p^t))$, where F is implemented by the widely used method in [85]. As S_{td}^t is dependent on the action of fixation prediction, the agent thus dynamically interacts with the attention-based observation environment.

In this paper, both the S_{bu}^t and S_{td}^t are normalized in $[0, 1]$ to follow their probability nature. Fig. 3.3 visualizes them along with corresponding video frames. It clearly shows that bottom-up attention highlights the most salient objects while top-down attention is more centralized in the risky region. This is because the foveal vision filters out irrelevant visual stimuli and only attends to the fixated area that indicates a high risk for a future accident.

To combine the two attention mechanisms, we propose a novel dynamic attention fusion (DAF) method which is a weighted-sum of S^t_{bu} and S^t_{td}:

$$S^t = (1-\rho^t)S^t_{bu} + \rho^t S^t_{td}, \tag{3.1}$$

where ρ^t is defined as $\rho^t = \min(m, a^t)$. Here, $a^t \in [0,1]$ is the predicted accident score and $m \in (0,1)$ serves as a hyperparameter. By introducing m to clip a^t, instead of directly using a^t, the DAF gains flexibility to utilize the learned top-down attention, because m controls the maximum percentage that S^t_{td} is utilized ($\rho^t \leq m$). Note that for any $a^t < m$, we have $\rho^t = a^t$ such that a^t and $1 - a^t$ are used as the weighting factors. The motivation is that the more probable there will be an accident ($a^t \rightarrow 1$) at the current time step, the more confident that the predicted top-down attention can be utilized at the next time step.

The benefits of Eq. (3.1) are enormous. Because both ρ^t and S^t_{td} are dependent on the actions from the agent, the proposed DAF method dynamically fuses visual attention by considering both the immediate observation from the environment and the previous decision made by the agent. Our experimental results show that DAF performs better accident anticipation than the static attention fusion (SAF), i.e., manually set a fixed weighting factor. Furthermore, because the attention mechanism is explicitly formulated for accident anticipation, the resulting decisions of the agent can be visually explained by telling which region is risky.

State Representation. Since CNNs show extraordinary capability to extract appearance features, we propose to utilize the feature volume $V^t \in \mathbb{R}^{C\times H\times W}$ from CNN-based saliency model G for state representation. To save the GPU memory usage while maintaining the representation capability of CNN features, the feature maps of the volume V^t are aggregated and further L_2-normalized by global max pooling ($\tilde{f}_{GMP}$) and global average pooling $\tilde{f}_{GAP}$. The normalized features are then concatenated as the observation state representation:

$$\mathbf{s}^i_t = \text{cat}\left(\tilde{f}_{GMP}(S^t \odot V^t_i), \tilde{f}_{GAP}(S^t \odot V^t_i)\right), \tag{3.2}$$

where $\odot$ is the element-wise product on the i-th channel of the feature volume V^t, and cat() is the concatenation along the channel dimension.

3.3.2 Stochastic Multi-task Agent

To simultaneously perform the accident anticipation and fixation prediction, the observation state $\mathbf{s}_t$ is shared with the two tasks. The state sharing brings two benefits. First, the state sharing establishes the causality relationship between the two tasks such that the visual attention modulated by the fixation prediction task could introspectively explain the accident anticipation outcomes, which distinguish our method from existing explanation-by-rationalization methods [29, 46]. The causal attention is also recently studied for explainable self-driving [141, 142]. Second, it significantly saves the communication workload between the environment and the agent, especially when the state $\mathbf{s}_t$ is of high dimensionality.

The quality of the state $\mathbf{s}_t$ is essential for improving the sample efficiency in DRL-based training. One of the typical ways is to include the auxiliary observation reconstruction task along with the prediction/control task of the agent [110, 88]. Inspired by this, we propose to use the Regularized Auto-Encoder (RAE) to encode the state $\mathbf{s}_t$ into a more compact low-dimensional latent representation $\mathbf{z}_t$, i.e., $\mathbf{z}_t = \mathcal{E}(\mathbf{s}_t)$ where $\mathcal{E}$ is the encoder part of RAE. And the decoder of RAE is to reconstruct the observation state.

To encourage more exploration of the environment, the agent policies are designed to be stochastic in recent state-of-the-art DRL algorithms [227, 280, 101]. In our model, an action is associated with both the accident score and the next fixation, drawn from a Gaussian, which can be leveraged for exploration. Therefore, the shared latent embedding $\mathbf{z}_t$ is used to predict the mean and the variance of each action dimension for the two tasks by two parallel policy networks, respectively (see the yellow boxes in Fig. 3.2). In the training stage, an action $\mathbf{a}_t$ is sampled from the predicted Gaussian distribution, i.e., $(\mathbf{a}_t) \sim \pi_\phi(\mathbf{a}_t|\mathbf{s}_t)$ where ϕ is the parameterized policy network. We implement the two policy networks by two fully-connected layers with ReLU activation. Besides, similar to the recent DRL-based attention models [228, 355], an LSTM [113] is utilized after the last FC layers to capture the temporal dependency of consecutive actions. In the testing stage, the predicted means of the accident score policy ϕ_A and the fixation policy ϕ_F are concatenated as action output:

$$\hat{\mathbf{a}}_t = \text{cat}\left(\phi_A\left(\mathcal{E}(\mathbf{s}_t)\right), \phi_F\left(\mathcal{E}(\mathbf{s}_t)\right)\right). \tag{3.3}$$

Note that we do not directly predict the top-down attention map but instead predict the fixation point as one of the actions in $\mathbf{a}_t$. The motivation is that the attention map prediction leads to a high dimension action space which is not efficient to be learned.

3.3.3 Reward Functions

With the observed state $\mathbf{s}_t$ and the executed action $\mathbf{a}_t$, the agent needs a scalar reward r from a driving environment to guide its learning. In this paper, we propose a dense anticipation reward r_A and a sparse fixation reward r_F to encourage early, accurate, and explainable decisions such that the total reward at each time step is $r = r_A + r_F$.

Dense Anticipation Reward. For the accident score a^t, we propose to reward it densely (for all time steps) by considering both the *correctness* and *earliness* at each time step. Given a score threshold a_0, we propose a temporally weighted XNOR[1] (also called Equivalence Gate) measurement as the accident anticipation reward r_A:

$$r_A^t = w_t \cdot \text{XNOR}\left[\mathbb{I}[a^t > a_0], y\right], \tag{3.4}$$

where w_t is the weighting factor and $\mathbb{I}(\cdot)$ is an indicator function. The binary label $y \in \{0, 1\}$ and $y = 1$ indicates there will be an accident in the future part of the video. The motivation to use XNOR is that it assigns one as the reward to the true predictions (either true positives or true negatives), while assigns zero reward to false predictions. Though in the autonomous driving scenario, false negative is more detrimental than false positive, it is non-trivial to manually design the weights to achieve the balance and it is out of the scope in this paper.

Furthermore, to encourage early anticipation (*earliness*), the temporally weighting factor w_t in Eq. (3.4) is designed as a normalized expression such that r_A and r_F can be numerically balanced

[1] XNOR: https://en.wikipedia.org/wiki/XNOR_gate

with the same magnitude scale:

$$w_t = \frac{1}{e^{t_a} - 1}\left(e^{\max(0, t_a - t)} - 1\right), \tag{3.5}$$

where t and t_a are the current time step t and the beginning time of a future accident, respectively. This factor exponentially decays from 1 to 0 before the accident occurs. Therefore, the earlier the decision is made, the larger reward will be given for the true positive prediction. After the accident occurs at t_a, there is no need to reward the agent.

Compared with the exponential binary-cross entropy loss in existing accident anticipation works [29, 315], our dense anticipation reward is more appropriate for DRL training.

Sparse Fixation Reward. Different from rewarding the accident scores, rewarding the predicted fixations is more challenging as the ground truth fixation data are valuable and typically only sparsely provided for a few accident frames [68]. To this end, we resort to a sparse rewarding scheme that is widely used in dynamic programming and reinforcement learning. In particular, our sparse fixation reward is given by

$$r_F^t = \mathbb{I}\left[t > t_a\right] \exp\left(-\frac{||\hat{p}^t - p^t||^2}{\eta}\right), \tag{3.6}$$

where the indicator function $\mathbb{I}\left[t > t_a\right]$ zeroes out the rewards of predictions before a future accident occurs. The $\hat{p}^t$ and p^t are 2-D coordinates of predicted and ground truth fixation point, respectively, defined in video frame space. The fixation points are normalized by the height and the width of the video frame for stable training. The motivation to use the radial kernel based on Euclidean distance is that the closer distance between $\hat{p}^t$ and p^t, the larger reward the agent will get. The hyperparameter η can be empirically set to ensure the same magnitude between r_F^t and r_A^t.

3.3.4 Model Training

To train the DRIVE model, we follow the soft actor-critic SAC model [101] but extend it to accommodate the accident anticipation task. SAC improves the exploration capacity of the traditional actor-critic RL through policy entropy maximization. Specifically, SAC aims to optimize the objective:

$$\max_{\phi} \sum_{t=1}^{T} \mathbb{E}_{(\mathbf{s}_t, \mathbf{a}_t) \sim \rho_{\pi_\phi}} \left[r(\mathbf{s}_t, \mathbf{a}_t) + \alpha \mathcal{H}(\pi_\phi(\cdot|\mathbf{s}_t))\right] \tag{3.7}$$

where α is the temperature that controls the contribution from the policy entropy $\mathcal{H}$. To achieve this objective, the actor which is the policy network, and the critic which approximates the state-action value function $Q(\mathbf{s}, \mathbf{a})$, are optimized in an interleaved way.

In our model, as the stochastic multi-task agent gives separate entropy estimation for accident anticipation and fixation prediction, we propose to express the total entropy as the sum of the entropy from each task. Using the logarithm rule, $-\mathcal{H}(\pi_\phi(\cdot|\mathbf{s}_t))$ can be expressed as

$$-\mathcal{H}(\pi_\phi(\hat{\mathbf{a}}|\mathbf{s})) = \log\left[\pi_{\phi_A}(\hat{a}|\mathbf{s}) \cdot \pi_{\phi_F}(\hat{p}|\mathbf{s})\right]. \tag{3.8}$$

This enables SAC to be extended to the multi-task agent. Our adapted SAC algorithm for training the DRIVE model is briefly summarized in the Algorithm 3.1.

Update Critic. For the critic network Q_θ, it is updated by minimizing the soft Bellman residual:

$$J(\theta) = \mathbb{E}\left[(Q_\theta(\mathbf{s}, \mathbf{a}) - y(r, \mathbf{s}', \mathbf{a}))^2\right], \tag{3.9}$$

where the target $y(r, \mathbf{s}', \mathbf{a})$ is greedily correlated with the reward r, the discounted soft Q-target $Q_{\bar{\theta}}$, and the entropy. Here, the $\bar{\theta}$ are parameters of the soft Q-target network which is the delayed soft copy of the critic network. More details are provided in [101] and our supplementary materials.

Update Actor. The policy networks (actor) are updated to maximize Eq. (3.7) by policy gradient method, which is equivalent to minimizing

$$J_o(\phi) = \mathbb{E}\left[\alpha \log \pi_\phi(\hat{\mathbf{a}}|\mathbf{s}) - \min_{j=1,2} Q_{\theta_j}(\mathbf{s}, \hat{\mathbf{a}})\right] + w_0||\phi||^2, \tag{3.10}$$

where $\hat{\mathbf{a}} \sim \pi_\theta(\cdot|\mathbf{s})$. The entropy term (logarithm part) is computed by Eq. (3.8). For the second term of expectation, the Clipped Double Q-learning [77] is used in practice. In this paper, we add an L_2 regularizer term for the policy network parameters ϕ to mitigate the over-fitting issue.

To accommodate SAC with multi-task policies in our model, we separately update each sub-policy network with corresponding losses $\mathcal{L}_A$ and $\mathcal{L}_F$ as regularizers:

$$\begin{aligned} J(\phi_A) &= J_o(\phi) + w_1\mathbb{E}\left[\mathcal{L}(\hat{a}^t, t_a, y)\right] \\ J(\phi_F) &= J_o(\phi) + w_2\mathbb{E}\left[\mathbb{I}[t > t_a] d(\hat{p}^t, p^t)\right], \end{aligned} \tag{3.11}$$

where $\phi = \{\phi_A, \phi_F\}$, and $(\hat{a}^t, \hat{p}^t, t_a, y, p^t)$ are sampled from a replay buffer $\mathcal{D}$. The distance $d(\cdot)$ defines the Euclidean distance. The accident anticipation loss $\mathcal{L}(\hat{a}^t, t_a, y)$ follows the definition in [315, 29]. The indicator function $\mathbb{I}[\cdot]$ ensures that only fixation points in accident frames can be accessed during training. Note that if $J_o(\phi)$ is removed, the SAC algorithm is reduced to a purely supervised learning (SL) without architectural modification.

Update Temperature. Recent works [102, 337] show that entropy-based RL training is brittle with respect to the temperature α. In this paper, we follow the automatic entropy tuning [102] that updates α by minimizing

$$J(\alpha) = \mathbb{E}\left[-\alpha \log \pi_\phi(\hat{\mathbf{a}}|\mathbf{s}) - \alpha \mathcal{H}_0\right], \tag{3.12}$$

where the negative target entropy $-\mathcal{H}_0$ is empirically set to the dimension of the action $\mathbf{a}$. In this paper, we found that α could be updated to zero such that the entropy (logarithm term) is hard to be optimized. To tackle this problem, we propose to clip α before it is updated:

$$\alpha \leftarrow \max(\alpha - \lambda_\alpha \hat{\nabla}_\alpha J(\alpha), \alpha_0) \tag{3.13}$$

where α_0 is a small value for α. This enables sufficient exploration of the agent during training.

Update RAE. The regularized auto-encoder (RAE) basically imposes L_2 regularizers on both the latent representation and model parameters for reconstruction learning:

$$J_{RAE}(\beta) = \mathcal{L}_{rec}(\mathbf{s}; \beta) + w_0 ||\beta||^2 + w_{\mathbf{s}} ||\mathbf{z}||^2, \tag{3.14}$$

where β are decoder parameters and $\mathbf{z}$ is the encoded state representation by RAE encoder. Similar to the existing work [369], the encoder parameters are updated by the critic loss $J(\theta)$ and the RAE loss $J_{RAE}(\beta)$ while the decoder parameters are only updated by $J_{RAE}(\beta)$. To enable the training on large-scale real-world videos, we reconstruct the observation state $\mathbf{s}$ rather than raw pixels as done in [369].

Summary of Our DRL Contribution. In this paper, the existing SAC algorithm is adapted to the real-world applications, which bridges the gap between simulation-based DRL applications and the challenging real-world tasks. Besides, for traffic accident anticipation, two novel reward functions by considering the earliness, correctness, and attentiveness are developed to guide the SAC-based model training. Moreover, to enable the multi-task learning by SAC, the proposed action entropy decomposition as well as other training techniques such as the temperature clipping and state reconstruction are empirically found useful.

3.4 Experiments

Datasets. Our method is evaluated on two traffic accident datasets, i.e., DADA-2000 [68] and DAD [29]. For the DADA-2000 dataset, we only use the beginning times of accidents and fixations of accident frames as ground truth. DAD is an accident dataset, in which the beginning times of accidents are fixed at the 90th frame for positive clips. The video clips of the two datasets are all 5 seconds long.

Evaluation Protocols. In this paper, we use the video-level Area Under ROC curve (AUC) to evaluate the anticipation correctness and the average time-to-accident (TTA) with score threshold 0.5 to evaluate the earliness. For classification metrics AUC, we only evaluate the predictions of accident frames since the output represents the occurrence probability of a future accident. To evaluate the visual attention, we adopt similarity (SIM), linear correlation coefficients (CC), and Kullback-Leibler distance (KLD). Smaller KL values indicate better performance.

Implementation Details. The proposed DRIVE model is implemented with the PyTorch framework. We adopt VGG-16-based MLNet [47] as the saliency module. The saliency module is pre-trained on the fixation data of the DADA-2000 training set and the parameters are kept frozen in DRIVE training. For the DAD dataset, as the fixations are not annotated, we remove the fixation prediction policy and top-down attention. For all datasets, the video frames are resized and zero-padded to 480×640 with an equal scaling ratio. The m of ρ in Eq. (3.1) and score threshold a_0 in Eq. (3.4) are set to 0.5 by default. We use the Adam optimizer for all gradient descent processes and train the DRIVE for 50 epochs. Other parameter settings are in the supplement.

3.4.1 Main Results

Baselines. We compare the proposed DRIVE with DSA-RNN [29] and UString [315] since their source codes are available. We also implement the accident anticipation loss function

Table 3.1 Comparison with state-of-the-art methods. The best results are marked with bold fonts. AUC and TTA evaluate the correctness and earliness of accident anticipation, respectively.

Methods	DADA-2000 [68]		DAD [29]	
	AUC (%)	TTA (s)	AUC (%)	TTA (s)
DSA-RNN [29]	47.19	3.095	71.57	1.169
AdaLEA [303]	55.05	**3.890**	58.06	2.228
UString [315]	60.19	3.849	65.96	0.915
DRIVE (ours)	**72.27**	3.657	**93.82**	**2.781**

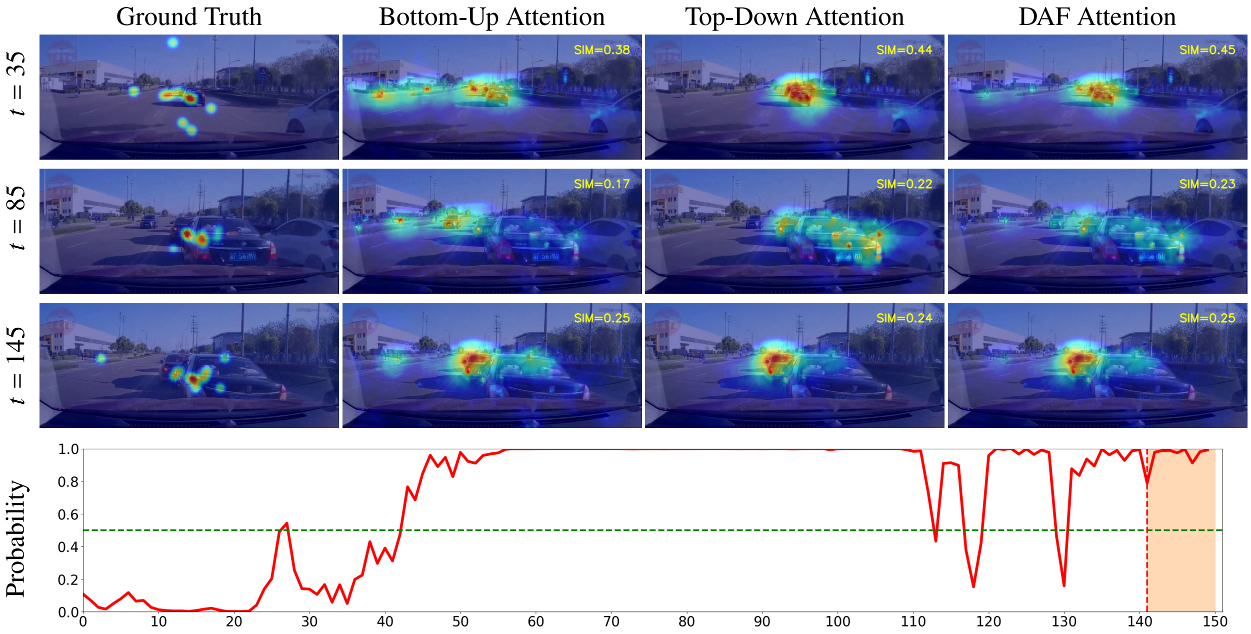
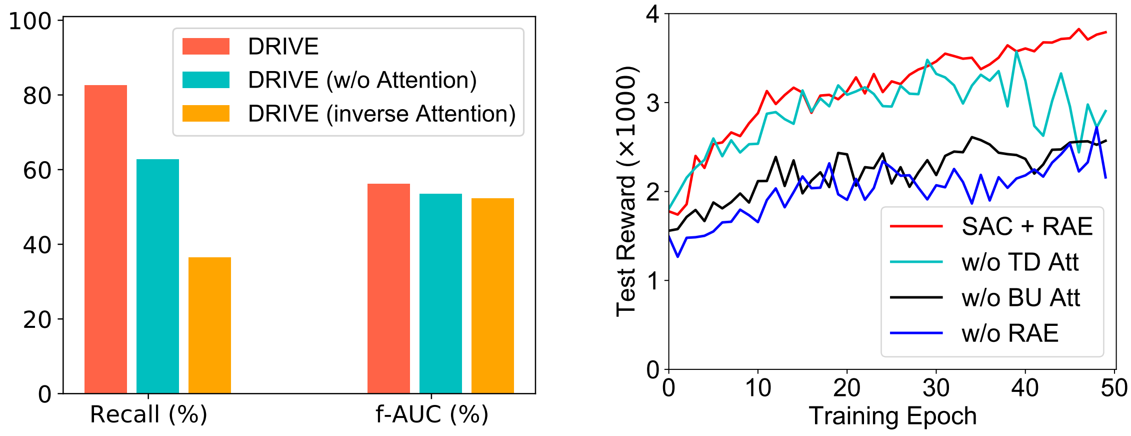


Figure 3.4 **Visualization on the DADA-2000 dataset.** The shaded region on the curve figure is the accident window (FPS=30). For this example, with the operation threshold 0.5 (dashed horizontal line) and a five-frame decision window, the model could anticipate future accidents at around the 42-th frame which is more than 3 seconds earlier before the accident occurs.

AdaLEA [303] on top of the DSA-RNN method (AdaLEA). Note that all methods are using VGG-16 [292] as the backbone. The AUC and TTA results on DADA-2000 and DAD datasets are reported in Table 3.1.

Results for Accident Anticipation. It shows that our DRIVE method significantly outperforms other baselines on both DADA-2000 and DAD dataset with the AUC metric. This demonstrates that our method is advantageous to accurately anticipate if a future accident will occur or not. Note that AdaLEA achieves the best TTA performance on DADA-2000 dataset, i.e., 0.23 seconds higher



(a) Intervention on Attention

(b) Reward Curves

Figure 3.5 Experimental results on DADA-2000 dataset.

than our DRIVE method. The advantage of AdaLEA on TTA metric can be attributed to that during training, AdaLEA utilizes validation set to compute TTA to drive the model to make early anticipation. In contrast, we do not use validation set guidance but still achieve comparable TTA results on DADA-2000 and much better TTA on DAD datasets.

3.4.2 Visual Explanation Results

Correlation Results. To investigate how the visual attention mechanism could explain the accident anticipation decision, we first jointly compare the performance of the fused saliency (Eq. (3.1)) and corresponding AUC score for both the proposed dynamic attention fusion (DAF) strategy and an alternative one, i.e., static attention fusion, for which the fusion parameter is manually set without updating. Results are reported in Table 3.2. We can see that DAF consistently outperforms SAF for both the saliency prediction and accident prediction (AUC), which demonstrates the superiority of DAF and the strong correlation between the visual attention and accident anticipation.

Causality Results. The visual attention learned by the proposed DRIVE model should exhibit the causality of accident anticipation performance. Therefore, inspired by the causal saliency analysis [141] and conterfactual visual explanation [93], we adopt two different intervention tests on the saliency S^t in Eq. (3.1), i.e., removing the attention ($S^t \leftarrow 1$) and inverse the attention ($1-S^t$) in testing stage. Results are reported in Fig. 3.5a. It shows that with either recall rate or frame-level AUC (f-AUC), the performance of the DRIVE model would decrease for both test cases, which

Table 3.2 Evaluation of visual attention and accident anticipation on DADA-2000. The best results are shown in bold. The parameters (Params) represent the values of ρ for SAF and m for DAF.

Params	Methods	AUC	SIM	CC	KLD (↓)
0.5	SAF	0.645	0.188	0.322	2.679
	DAF	**0.659**	**0.192**	**0.331**	**2.654**
0.8	SAF	0.691	0.144	0.190	3.087
	DAF	**0.726**	**0.158**	**0.226**	**2.986**
1.0	SAF	0.632	0.080	0.079	12.948
	DAF	**0.679**	**0.112**	**0.143**	**7.836**

demonstrate causality relation between the learned DAF visual attention and accident anticipation.

Attention Visualization. In Fig. 3.4, we visualized the saliency maps on three representative time steps using test data from DADA-2000 dataset. For reference, the curve of the predicted accident probability is also presented. We can see that the bottom-up attention captures visually attentive regions while the top-down attention indicates the risky region both before and after the accident occurs. The DAF attention maps exhibit the fused attention.

3.4.3 Ablation Studies

In Table 3.3, we report the results of ablation studies with DADA-2000 dataset. In the first row, we remove the fixation prediction policy and corresponding learning objectives, we see the AUC is about 10% lower than our full model (the last row). The second row shows that the RAE module also contributes a lot to the performance gain. To further see if it is the DRL framework itself that leads to good performance, we keep the DRIVE architecture unchanged and only remove the $J_O(\phi)$ in Eq. (3.11) for training the multi-task policy networks such that the algorithm reduces to a supervised learning (SL). Results in the third row of Table 3.3 show that DRL-based learning method (SAC) is superior to SL algorithm for accident anticipation.

To show the performance of SAC-based DRIVE variants during training, we plot their reward curves in Fig. 3.5b. It shows that training DRIVE model by SAC + RAE could achieve stable increasing reward. Besides, both top-down attention (w/o BU Att) and bottom-up attention (w/o TD Att) could contribute to the learning process. In particular, we see that RAE contributes most to the performance gain.

Table 3.3 Ablation studies on DADA-2000 dataset. In the Type column, “RL" and “SL" represent reinforcement learning and supervised learning, respectively.

Type	SAC	RAE	Fixations	AUC (%)
RL	✓	✓	✗	61.91
RL	✓	✗	✓	66.21
SL	✗	✓	✓	63.96
RL	✓	✓	✓	**72.27**

3.5 Conclusion

In this paper, we propose the DRIVE model to anticipate traffic accidents from dashcam videos. Based on deep reinforcement learning (DRL), we explicitly simulate both the bottom-up and top-down visual attention in the traffic observation environment and develop a stochastic multi-task agent to dynamically interact with the environment. The DRIVE model is learned by the improved DRL algorithm SAC. Experimental results on real-world traffic accident datasets show that our method achieves the best anticipation performance as well as good visual explainability.

3.6 Supplementary Material

This document provides further details about the training algorithms of SAC [101], and implementation settings.

3.6.1 Soft Actor Critic

As introduced in Section 3.3.4 in the main paper, to adapt the soft actor critic (SAC) [101] algorithm to our DRIVE model training, original SAC algorithm needs to be substantially adapted. The Algorithm 3.1 summarizes the training steps of the improved SAC algorithm.

At first, the transitions including the current state $\mathbf{s}_t$, action $\mathbf{a}_t$, immediate reward r_t, next state $\mathbf{s}_{t+1}$, and the hidden states of LSTM layer $\mathbf{h}_t$ are gathered into the replay buffer $\mathcal{D}$. For each gradient step, a mini-batch of transitions are uniformly sampled from $\mathcal{D}$ to update different model components, including the policy networks (**actor**), Q-networks (**critic**), and **RAE**. As the actor update, automatic entropy tuning, and RAE update are elaborated clearly in the main paper, here we only present more details about how the critic networks are learned during SAC training.

To update the critic, in practice, the Clipped Double Q-learning [77] is used that two identical

Q-networks θ_i ($i \in \{1,2\}$) are maintained. The loss function takes the sum of the losses from the two outputs, i.e., $J(\theta) = \sum_i J(\theta_i)$, where each of them $J(\theta_i)$ is defined as the expectation of mean-squared error:

$$J(\theta_i) = \mathbb{E}\left[\left(Q_{\theta_i}(\mathbf{s},\mathbf{a}) - y(r,\mathbf{s}',\mathbf{a})\right)^2\right], \tag{3.15}$$

Here, the optimization target $y(r,\mathbf{s}',\mathbf{a})$ is defined as

$$y(r,\mathbf{s}',\mathbf{a}) = r + \gamma(1-d)\left(\min_{j=1,2} Q_{\bar{\theta}_j}(\mathbf{s}',\hat{\mathbf{a}}') - \alpha \log \pi_\theta(\hat{\mathbf{a}}'|\mathbf{s}')\right) \tag{3.16}$$

where r is the reward batch, γ is the discounting factor, and d labels whether the sampled transitions are at the last step T. Note that the sate $\mathbf{s}'$ is the batch of next state from replay buffer, while the action $\hat{\mathbf{a}}'$ is sampled from the output of pre-updated policy network π_θ, i.e., $\hat{\mathbf{a}}' \sim \pi_\theta(\cdot|\mathbf{s}')$ which enables SAC to be an off-policy method. The entropy term $\log \pi_\theta(\hat{\mathbf{a}}'|\mathbf{s}')$ is obtained by the Eq. 3.8 in our main paper.

In this paper, the critic network parameters θ are updated more frequently than other parameters by the gradients of $J(\theta)$ to achieve more stable training. The Table 3.4 summarizes the hyperparameter setting in experiments. Note that the major hyperparameters are following existing literature [101]. For different datasets, we used the same set of hyperparameters and did not tune them specifically.

3.6.2 Implementation Details

Network Architecture. As shown in Fig. 3.2 in the main paper, the saliency model is implemented with the existing CNN-based saliency model [47], which takes as input with the size $480 \times 640 \times 3$ and output the feature volume V^t with the size $60 \times 80 \times 64$ by default. The stochastic multi-task agent consists of a shared RAE and two policy networks, i.e., accident prediction and fixation prediction branches. In our implementation, the encoder of RAE consists of three fully-connected (FC) and the decoder is symmetric to the encoder. Each policy branch consists of two FC layers and one LSTM layer, followed by an FC layer for predicting means and an FC layer for predicting the variance. ReLU activations are used for all layers except for the last FC output layer. According to the default SAC setting, the output of policy networks $\mathbf{a}_t$ are activated by *tanh*

Algorithm 3.1 Improved SAC for the DRIVE Model Training

```
Require: $\theta_1, \theta_2, \phi, \beta$                          ▷ Initial parameters
$\bar{\theta}_1 \leftarrow \theta_1, \bar{\theta}_2 \leftarrow \theta_2$                  ▷ Initialize target networks
$\mathcal{D} \leftarrow \emptyset, \mathbf{h}_0 \leftarrow \mathbf{0}$                       ▷ Replay buffer and hidden states
for each iteration do
    for each environment step do
        Sample actions $(\mathbf{a}_t, \mathbf{h}_t) \sim \pi_\phi(\mathbf{a}_t|\mathbf{s}_t, \mathbf{h}_{t-1})$
        Compute state $\mathbf{s}_t$ with actions                  ▷ See Eq. 3.2
        Compute reward $r_t = r_A^t + r_F^t$                     ▷ See Eq. 3.4-3.6
        $\mathcal{D} \leftarrow \mathcal{D} \cup \{(\mathbf{s}_t, \mathbf{a}_t, r_t, \mathbf{h}_t, \mathbf{s}_{t+1})\}$
    end for
    for each gradient step do
        for each critic update do
            $\theta \leftarrow \theta - \lambda \hat{\nabla}_\theta J_Q(\theta)$              ▷ Update by Eq. 3.9
        end for
        $\phi \leftarrow \phi - \lambda \hat{\nabla}_\phi J_\pi(\phi)$                  ▷ Update by Eq. 3.11
        $\alpha \leftarrow \max(\alpha - \lambda_\alpha \hat{\nabla}_\alpha J(\alpha), \alpha_0)$          ▷ See Eq. 3.12
        $\bar{\theta} \leftarrow \tau\theta + (1-\tau)\bar{\theta}$                    ▷ Update Q-target
        $\beta \leftarrow \beta - \lambda \hat{\nabla}_\beta J_{\mathrm{RAE}}(\beta)$               ▷ Update by Eq. 3.14
    end for
end for
Ensure: $\theta_1, \theta_2, \phi, \beta$
```

functions so that the values are constrained in $(-1, 1)$. In order to map the values to accident scores a^t and fixation coordinates p^t, we linearly scale the values by

$$a^t = 0.5(\mathbf{a}_t^{(0)} + 1.0) \tag{3.17}$$

$$p^t = \psi(\mathbf{a}_t^{(1)}, \mathbf{a}_t^{(2)}) \tag{3.18}$$

where the equation of a^t applied to *tanh* activation is equivalent to *sigmoid* activation on FC layer output. The function ψ maps the scaling factors (within $(-1, 1)$) defined in image space $H \times W$ to the input space $h \times w$. This scaling process is illustrated in Fig. 3.6.

Implementation. Our training algorithm is implemented based on the SAC source code[2]. Since the image foveation method [85] incurs computational cost due to the Gaussian pyramid filtering, we implement this algorithm as well as all the DRL environmental components by PyTorch to support for GPU acceleration. For DADA-2000 videos, the positive video clips (contains accident)

[2] SAC: https://github.com/pranz24/pytorch-soft-actor-critic

Table 3.4 SAC Hyperparameter Settings

Parameters	values
general learning rate (λ)	$3 \cdot 10^{-4}$
temperature learning rate (λ_α)	$5 \cdot 10^{-5}$
discounting factor (γ)	0.99
replay buffer size ($\mathcal{D}$)	10^6
target smoothing coefficient (τ)	0.005
temperature threshold (α_0)	10^{-4}
weight decay (w_0)	10^{-5}
anticipation loss coefficient (w_1)	1
fixation loss coefficient (w_2)	10
latent regularizer coefficient (w_s)	10^{-4}
sparse fixation reward parameter (η)	0.1
gradient updates per time step	4
actor gradient updates per time step	2
dim. of FC/LSTM layers output	64
dim. of latent embedding ($\mathbf{z}$)	64
dim. of state ($\mathbf{s}$)	128
dim. of action ($\mathbf{a}$)	3
sampling batch size	64
video batch size	5

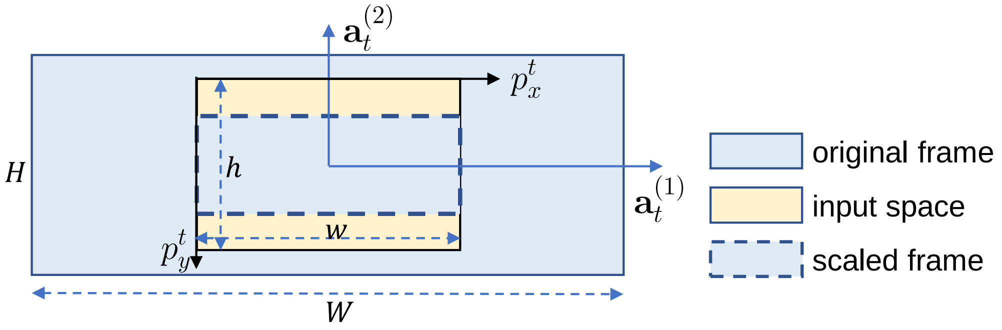


Figure 3.6 **The scaling process** (ψ). The continuous values $\mathbf{a}_t^{(1)}$ and $\mathbf{a}_t^{(2)}$ which are within $(-1, 1)$ defined in video frame space $H \times W$ are mapped into the discrete input space $h \times w$ to represent the 2-D coordinates of a fixation point.

are obtained by trimming the video into be 5 seconds where the beginning times are placed in the last one second with random jittering, while the negative video clips are randomly sampled without overlap with positive clips. The spatial and temporal resolutions for DADA-2000 videos are reduced with ratio 0.5 and interval 5, respectively, so that 30 time steps are utilized and for each step the observation frames are with the size 330×792. For DAD dataset, we only reduce the temporal resolution with interval 4 so that 25 time steps of each 5-seconds video clip are used.

CHAPTER 4

EGOCENTRIC 3D TRAJECTORY FORECASTING

4.1 Introduction

Egocentric video understanding aims to understand the camera wearers' behavior from the first-person view. It is receiving increasing attention in recent years [95, 181, 90, 202, 177, 238, 334, 203, 50, 49, 182] due to its analogousness to the way human visually perceives the world. An important egocentric vision task is to forecast the egocentric hand trajectory of the camera wearer [205], which has great value in Augmented/Virtual Reality (AR/VR) applications. For example, the predicted 3D trajectories can help plan and stabilize a patient's 3D hand motion who has upper-limb neuromuscular disease [181]. Besides, the early predicted 3D hand trajectory is key to reducing rendering latency in VR games for achieving an immersive gaming experience [80].

In the existing literature, egocentric 3D hand trajectory forecasting is far from being explored. The method in [205] could only predict 2D trajectory on an image and cannot forecast precise 3D hand movements. Recent works [17, 331, 379] predict the trajectory or 3D human motions from egocentric views, but they do not predict the 3D trajectory of the camera wearer. Besides, though forecasting the 3D hand pose provides fine-grained information about 3D hands [63], it is out of our scope as we focus on the camera wearers' planning behavior revealed by 3D hand trajectory.

The challenges of egocentric video-based 3D trajectory forecasting are significant. First, accurate large-scale 3D trajectory annotations are labor-intensive and expensive. They rely on wearable markers or multi-camera systems for hand motion capture in a controlled environment. Second, learning the depth of 3D trajectory from egocentric videos is challenging. On one hand, using 2D video frames to estimate 3D trajectory depth is an ill-conditioned problem similar to other monocular 3D tasks [309, 253, 313]. Even if the historical 3D hand trajectory is utilized

This chapter is adapted from the following publication:
"Wentao Bao, Lele Chen, Libing Zeng, Zhong Li, Yi Xu, Junsong Yuan, and Yu Kong. Uncertainty-aware state space transformer for egocentric 3D hand trajectory forecasting. In *International Conference on Computer Vision (ICCV)*, 2023."

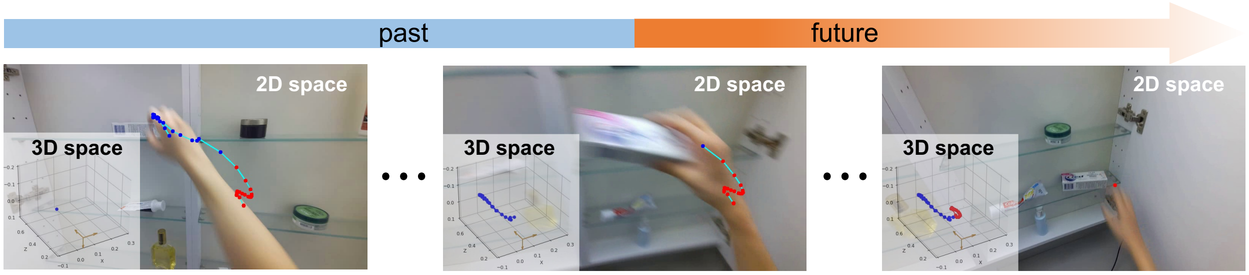


Figure 4.1 **Egocentric 3D Hand Trajectory Forecasting**. Our goal is to predict the future 3D hand trajectory (in **red**) given the past observations of egocentric video and trajectory (in **blue**). Compared to the 2D image space, predicting trajectory in a global 3D space is practically more valuable to understanding human intention and behavior in AR/VR applications.

as the input, how to exploit the visual and trajectory information for forecasting is still nontrivial. On the other hand, due to the inevitable camera motion in an egocentric view, the background of the scene is visually dynamic which poses a significant barrier to inferring the foreground depth [187, 397]. Third, as a Seq2Seq forecasting problem, it is critical to formulate the latent transition dynamics [91, 10] that allows the variances of data due to anytime forecasting and limited observations.

In this paper, we address these challenges by developing an uncertainty-aware state space transformer (USST). It follows the state-space model [263] by taking the observed egocentric RGB videos with the historical 3D trajectory as input to predict future 3D trajectory. Our model deals with the depth noise of trajectory annotation by introducing the depth robust aleatoric uncertainty in training. To fuse the information from the dense RGB pixels and sparse historical trajectory, we leverage visual and temporal transformer encoders as backbones and utilize the recent visual prompt tuning (VPT) to enhance the visual features. Following the state space model, we develop a novel attention-based state transition module and an emission module with a predictive link to predict the 3D global trajectory coordinates. Moreover, to take the hand motion inertia into consideration, we propose a velocity constraint to regularize the model training, which helps generalize to unseen scenarios.

To enable egocentric 3D hand trajectory prediction, we follow [181] to develop a scalable annotation workflow to automate the annotation on RGB-D data from head-mounted Kinect devices.

In particular, camera motion is estimated to transform the 3D trajectory annotations from local to global camera coordinate system. Experimental results on H2O and EgoPAT3D datasets show that our method is effective and superior to existing Transformer-based approaches [205] and other general Seq2Seq models. In summary, our contributions are as follows:

- We propose an uncertainty-aware state space transformer (USST) that consists of a novel state transition and emission, aleatoric uncertainty, and visual prompt tuning, which are empirically found effective.
- We collected and will release our annotations on H2O [162] and EgoPAT3D [181] datasets that will benefit the egocentric 3D hand trajectory forecasting research.
- We benchmarked recent methods on the proposed task and experimental results show that our method achieves the best performance and could be generalizable to unseen egocentric scenes.

4.2 Related Work

Trajectory Prediction Predicting the physical trajectory of moving objects is a long-standing research topic. It has been widely studied in applications for pedestrians [2, 393], vehicle [59, 219]. Many of them are developed for the third-person view and predict trajectory in 2D pixel space. Given that the first-person view is more realistic in AR/VR applications, recent few works [205, 203] are trying to predict the hand-object interaction from egocentric videos. Though the method in [17] predicts the pedestrian trajectory in 3D space, their method primarily addresses the social interaction of multiple pedestrians. The recent work [260] also targets the egocentric 3D trajectory for pedestrian scenarios, but their method leverages depth modality and nearby person's trajectory as context information which are practically uneasy to collect. Besides, due to the annotation noise and uncertain nature of trajectory prediction, probabilistic modeling is widely adopted in existing literature [341, 320, 219]. In this paper, following the probabilistic setting, we step toward the egocentric 3D hand trajectory prediction using practically accessible RGB videos for AR/VR scenarios.

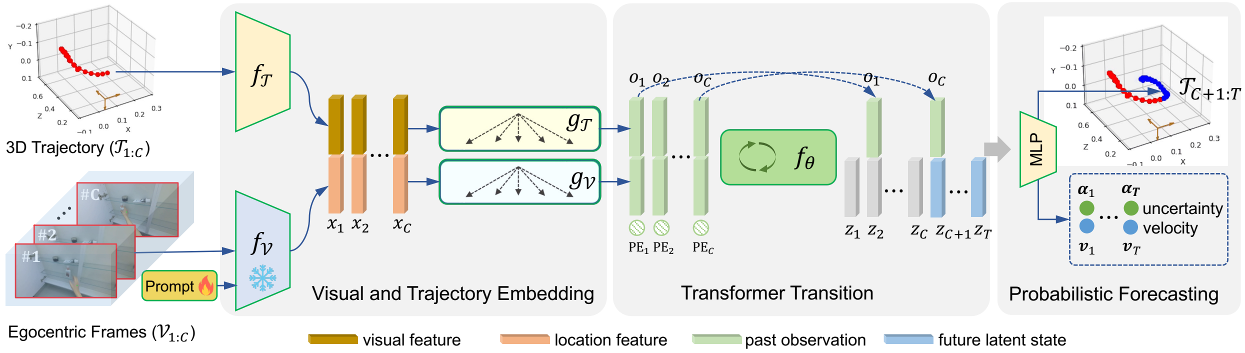


Figure 4.2 **Proposed USST Model**. Given the RGB frames and 3D hand locations of C observed time steps, we extract and concatenate their features as $\mathbf{x}_{1:C}$ by the prompted backbone $f_\mathcal{V}$ and MLP model $f_\mathcal{T}$, which are further fed into transformer encoders to produce temporal observations $\mathbf{o}_{1:C}$. Together with positional encodings $\mathtt{PE}_{1:T}$ of the full horizon, our state transition layer could recursively extrapolate the latent states $\mathbf{z}_{C+1:T}$ for 3D trajectory forecasting along with uncertainty and velocity ($\boldsymbol{\alpha}$ and $\mathbf{v}$) in $T - C$ future steps.

Egocentric Video Representation Egocentric videos are recorded in a first-person view. Different from the videos in a third-person view, learning an egocentric video representation is more challenging due to the dynamic background caused by camera motion and implicit intention of activities from camera wearers [177, 70, 244]. For 3D trajectory prediction, existing commonly used egocentric video datasets such as EPIC-Kitchens [49, 50] do not provide the depth information and camera parameters, which are essential for annotating the 3D hand trajectory. Though the recent Ego4D [95] benchmark provides a hand forecasting subset, the annotations are defined as 2D locations in image space. Therefore, we resort to a cost-effective workflow to collect 3D hand trajectory annotations from existing 3D hand pose datasets such as the EgoPAT3D [181] and H2O [162] datasets. Our annotation workflow can be deployed to any egocentric dataset collected by head-mounted RGB-D sensors.

State Space Model State-space Model (SSM) originates from the control engineering field. It is conceptually general and inspires many classical SSMs such as the Kalman Filtering for prediction tasks. Recent deep SSMs [13, 43, 75] are increasingly popular by combining Recurrent Neural Networks (RNNs) with the Variational AutoEcoders (VAE). However, these approaches are limited in practice due to the complex long-term dependency on highly-structured sequential data. In [263], a deep SSM is proposed by combining Kalman Filtering with deep neural networks. However, the linear Gaussian assumption is uneasy to hold for high-dimensional data in the real world. To address these challenges, ProTran [305] introduces a probabilistic Transformer [328] under a variational SSM for time-series prediction. However, maximizing the variational low bound of ProTran suffers from notorious KL vanishing issue [76]. AgentFormer [380] can also be regarded as a Transformer-based SSM, but its autoregressive decoding limits its efficiency to low-dimensional motion data. In this paper, we propose a Transformer-based SSM that allows for long-term dependency and latent dynamics efficiently in practice.

4.3 Approach

Problem Setup As shown in Fig. 4.1, the Egocentric 3D hand trajectory forecasting model takes as input C observed RGB frames $\mathcal{V}_{1:C} = \{\mathbf{I}_1, \dots, \mathbf{I}_C\}$ and 3D hand trajectory $\mathcal{T}_{1:C} = \{\mathbf{p}_1, \dots, \mathbf{p}_C\}$ to predict the future 3D hand trajectory $\mathcal{T}_{C+1:T} = \{\mathbf{p}_{C+1}, \dots, \mathbf{p}_T\}$ in a finite horizon T. Here, $\mathbf{I}_t \in \mathbb{R}^{H\times W\times 3}$ and $\mathbf{p}_t = [x_t, y_t, z_t]^\top$ are egocentric RGB frame and 3D hand trajectory point at time step t, respectively. In practice, the 3D point $\mathbf{p}_t$ is defined in a global 3D world coordinate system. The ultimate goal is to learn a model $\mathbf{\Phi}$ by maximizing the expectation of the likelihood over the training dataset $\mathcal{D}$:

$$\max_{\mathbf{\Phi}} \mathbb{E}_{\mathcal{V},\mathcal{T}\sim\mathcal{D}} \left[p_{\mathbf{\Phi}}(\mathcal{T}_{C+1:T} | \mathcal{T}_{1:C}, \mathcal{V}_{1:C}) \right] . \tag{4.1}$$

In this paper, we formulate the problem as a state-space model. In the following sections, we will introduce the proposed model in detail.

4.3.1 Uncertainty-aware State Space Transformer

Existing SSMs could formulate the probabilistic nature of trajectory prediction. However, they do not explicitly handle the data noise issue which is commonly encountered when using RGB-D sensors to get 3D trajectory annotations. To mitigate the uncertainty from data labeling, we follow the line of research [136, 108] and propose an uncertainty-aware state space transformer (USST) to handle the dynamics of 3D hand trajectory in egocentric scenes.

Formally, following the latent variable modeling, the probability in Eq. (4.1) over a full sequence

$\mathcal{T}_{1:T}$ can be factorized by introducing T latent variables $\mathcal{Z}_{1:T}$:

$$p(\mathcal{T}_{1:T}|\mathcal{T}_{1:C}, \mathcal{V}_{1:C}) = \int p(\mathcal{T}_{1:T}|\mathcal{Z}_{1:T}) p(\mathcal{Z}_{1:T}|\mathcal{T}_{1:C}, \mathcal{V}_{1:C}) d\mathcal{Z}, \tag{4.2}$$

where the *state transition* $p(\mathcal{Z}_{1:T}|\mathcal{T}_{1:C}, \mathcal{V}_{1:C})$ and the *emission* $p(\mathcal{T}_{1:T}|\mathcal{Z}_{1:T})$ are learned from data $\mathcal{D}$. Following the SSM formulation, we propose to factorize the two terms by the independency assumptions:

$$\begin{aligned} p_\theta(\mathcal{Z}_{1:T}|\mathcal{T}_{1:C}, \mathcal{V}_{1:C}) &= \prod_{t=1}^{T} p_\theta(\mathbf{z}_t|\mathbf{z}_{1:t-1}, \mathbf{p}_{1:C}, \mathbf{I}_{1:C}), \\ p_\phi(\mathcal{T}_{1:T}|\mathcal{Z}_{1:T}) &= \prod_{t=1}^{T} p_\phi(\mathbf{p}_t|\mathbf{z}_t, \mathbf{p}_{t-1}), \end{aligned} \tag{4.3}$$

where the latent variable $\mathbf{z}_t \in \mathcal{Z}_{1:T}$ is generated by taking as input $\mathbf{z}_{t-1}$ and the previous trajectory point $\mathbf{p}_t$ and RGB frame $\mathbf{I}_t$. To address the label noise issue, we formulate the emission model $p_\phi(\mathbf{p}_t|\mathbf{z}_t, \mathbf{p}_{t-1})$ as a probabilistic module to learn the aleatoric uncertainty. In the following paragraphs, we will elaborate on feature embedding, state transition, and probabilistic prediction.

Visual and Trajectory Embedding As advocated by [305, 380], Transformers are effective to capture the long-term dependency for sequential data. Therefore, we propose to leverage visual and temporal Transformers [328] as encoders to learn the features from the dense RGB frames and sparse trajectory points. Specifically, we first embed the observed sequence of egocentric RGB frames and 3D trajectory by models $f_\mathcal{V}$ and $f_\mathcal{T}$, followed by modality-specific transformers $g_\mathcal{V}$ and $g_\mathcal{T}$. This process can be expressed as

$$\begin{aligned} &[\mathbf{x}_t^{(\mathcal{V})}, \mathbf{x}_t^{(\mathcal{T})}] = [f_\mathcal{V}(\mathbf{I}_t), f_\mathcal{T}(\mathbf{p}_t)], \\ &\mathbf{o}_1^{(\mathcal{V})}, \ldots, \mathbf{o}_C^{(\mathcal{V})} = g_\mathcal{V}(\mathbf{x}_1^{(\mathcal{V})}, \ldots, \mathbf{x}_C^{(\mathcal{V})}), \\ &\mathbf{o}_1^{(\mathcal{T})}, \ldots, \mathbf{o}_C^{(\mathcal{T})} = g_\mathcal{T}(\mathbf{x}_1^{(\mathcal{T})}, \ldots, \mathbf{x}_C^{(\mathcal{T})}), \end{aligned} \tag{4.4}$$

where $f_\mathcal{V}$ is a vision backbone, e.g., ResNet [106] and ViT [65]. $f_\mathcal{T}$ is implemented as MLP following [205, 181]. $g_\mathcal{V}$ and $g_\mathcal{T}$ are transformer encoders that consist of B stacked multi-head attention blocks. For each block b, a single-head attention block can be expressed as

$$\text{Attn}(\mathbf{Q}_b, \mathbf{K}_b, \mathbf{V}_b) = \text{softmax}\left(\frac{\mathbf{Q}_b \mathbf{K}_b^\top}{\sqrt{d}} + \mathbf{M}\right) \mathbf{V}_b, \tag{4.5}$$

where $\mathbf{Q}_b, \mathbf{K}_b, \mathbf{V}_b \in \mathbb{R}^{T\times d}$ are the projected query, key, and value matrices from the output of the previous block $b-1$, i.e, $[\mathbf{Q}_b; \mathbf{K}_b; \mathbf{V}_b] = [\mathbf{W}_b^Q \mathbf{Q}_{b-1}; \mathbf{W}_b^K \mathbf{K}_{b-1}; \mathbf{W}_b^V \mathbf{V}_{b-1}]$. All the $\mathbf{W}_b$ are learnable parameters. The binary mask $\mathbf{M} \in \mathbb{R}^{T\times T}$ zeros out the last $T-C$ columns and rows for the trajectory prediction problem. To capture global temporal interaction, the input of the first block $\mathbf{Q}_0$, $\mathbf{K}_0$, and $\mathbf{V}_0$ are the same as $\mathbf{x}_t + \mathrm{PE}(t)$ where $\mathrm{PE}(t)$ is the positional encoding for $t \in [1, T]$.

Transformer Transition With the encoded observations $\mathbf{o}_t = [\mathbf{o}_t^{(\mathcal{V})}; \mathbf{o}_t^{(\mathcal{T})}]$ where we use $[;]$ to represent the feature concatenation, it is critical to formulate the state transition and future trajectory prediction based on Eq. (4.3). Inspired by [305], we propose an attention-based autoregressive module to formulate posterior $p_\theta(\mathbf{z}_t|\mathbf{z}_{1:t-1}, \mathbf{o}_{1:C})$. Specifically, we first embed $\mathbf{o}_t$ with positional encoding by $\mathbf{h}_t = \mathrm{LayerNorm}(\mathrm{MLP}(\mathbf{o}_t) + \mathrm{PE}(t))$. Then, the latent feature $\mathbf{z}_t$ is recursively encapsulated by the hidden variables $\bar{\mathbf{w}}_t$ and $\hat{\mathbf{w}}_t$ by attention modules (illustrated in Fig. 4.3):

$$\begin{aligned}
\bar{\mathbf{w}}_t &= \mathrm{LayerNorm}([\mathbf{z}_{t-1}; \mathrm{Attn}(\mathbf{z}_{t-1}, \mathbf{z}_{1:t-1}, \mathbf{z}_{1:t-1})]), \\
\hat{\mathbf{w}}_t &= \mathrm{LayerNorm}([\bar{\mathbf{w}}_t; \mathrm{Attn}(\bar{\mathbf{w}}_t, \mathbf{h}_{1:C}, \mathbf{h}_{1:C})]), \\
\mathbf{z}_t &= \mathrm{LayerNorm}(\mathrm{MLP}([\hat{\mathbf{w}}_t; \mathrm{MLP}(\hat{\mathbf{w}}_t)]) + \mathrm{PE}(t)),
\end{aligned} \tag{4.6}$$

where the two multi-head attention modules capture the previously generated state $\mathbf{z}_{1:t-1}$ and the hidden states of all observations $\mathbf{h}_{1:C}$. Contrary to ProTran, we use concatenation $[;]$ rather than addition before layer normalization. The insight behind this is that the queried feature from a historical context can be better preserved without being dominated by $\mathbf{z}_{t-1}$ in addition operation. Moreover, we remove the stochasticity of $\mathbf{z}_t$ and instead use a probabilistic decoder as introduced next to handle the dynamics of trajectory prediction. The benefit is that we avoid the KL divergence vanishing issue from optimizing the ELBO objective which is known to exist in variational recurrent models [76].

Probabilistic Forecasting Instead of placing the stocasticity in the transition model $p_\theta(\mathbf{z}_t|\mathbf{z}_{1:t-1}, \mathbf{o}_{1:C})$, we propose to formulate the emission process $p_\phi(\mathbf{p}_t|\mathbf{z}_t, \mathbf{p}_{t-1})$ as a probabilistic model by predicting

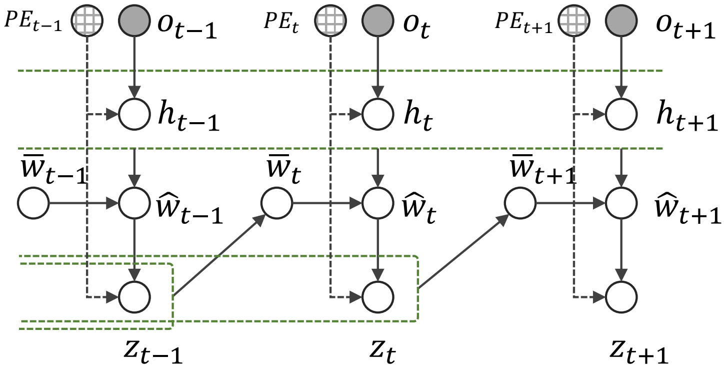


Figure 4.3 **Unrolled illustration of Eq.** (4.6)**.** Bold arrows are learnable, and green dashed lines show the attention ranges.

both the mean $\hat{\mathbf{p}}_t$ and variance $\hat{\sigma}_t^2$ of each 3D hand trajectory point:

$$[\hat{\mathbf{p}}_t, \hat{\alpha}_t] = [\text{MLP}([\mathbf{z}_t; \mathbf{o}_{t-1}^{(\mathcal{T})}]), \text{softplus}(\text{MLP}([\mathbf{z}_t; \mathbf{o}_{t-1}^{(\mathcal{T})}]))], \tag{4.7}$$

where the uncertainty $\hat{\alpha}_t := \log \hat{\sigma}_t^2$ to enable numerical stability. The trajectory point $\mathbf{p}_t$ follows a predictive Gaussian distribution, i.e., $\mathbf{p}_t \sim \mathcal{N}(\hat{\mathbf{p}}_t, \hat{\sigma}_t)$. As the $\mathbf{o}_{t-1}^{(\mathcal{T})}$ encodes the observation from $\mathbf{p}_{t-1}$ and its global historical context, our emission model is thus more powerful to predict $\mathbf{p}_t$. This predictive mode has also been successful in traditional methods such as the SRNN [75] and VRNN [43].

Discussion Compared to ProTran [305], our formulation could individually formulate the trajectory and visual context $\mathbf{p}_{1:C}$ and $\mathbf{I}_{1:C}$ in state transition by modality-specific embeddings and the predictive link from $\mathbf{p}_{t-1}$ to $\mathbf{p}_t$, while ProTran only handles single modality context $\mathbf{p}_{1:C}$ in state transition and emission. In addition, to learn model parameters θ and ϕ, ProTran has to use variational posterior distribution $q_\theta(\mathbf{z}_t|\mathbf{z}_{1:t-1}, \mathbf{p}_{1:C})$ to help approximate Eq. (4.2) by ELBO maximization. In contrast, our method does not need approximation and formulates the data uncertainty of $\mathbf{p}_t$ from a Bayesian perspective (Eq. (4.6)), which is empirically more effective to handle data noise.

4.3.2 Model Training

Depth Robust Aleatoric Uncertainty With the predicted trajectory points $\hat{\mathbf{p}}_t$ along with the uncertainty $\hat{\alpha}$, according to [136, 10], the model training is essentially to learn the heteroscedastic aleatoric uncertainty (HAU) from data. As shown in [137, 108], by minimizing the KL divergence between the predictive Gaussian distributions and Dirac distribution of the ground truth trajectory, the objective in Eq. (4.1) is equivalent to minimizing the HAU loss at each time t:

$$\mathcal{L}_{\text{haul}}(\hat{\alpha}, \hat{\mathbf{p}}, \mathbf{p}) = e^{-\hat{\alpha}} \sum_{i=1}^{|\mathbf{p}|} \|p_i - \hat{p}_i\|^2 + \hat{\alpha}, \tag{4.8}$$

where p_i and $\hat{p}_i$ are 3D coordinate values (x, y, z) of ground truth and model prediction, respectively.

In our task, the trajectory depth z is more challenging to predict than x and y due to 1) the weak implicit correspondence between the past visual context $\mathcal{V}_{1:C}$ and future hand depth, and 2) more importantly, the inevitable annotation noise from depth sensors. To handle these challenges, we propose to decouple the aleatoric uncertainty into $\hat{\alpha}_t$ which is isotropic for (x, y) and $\hat{\beta}_t$ specifically for z, respectively. Then, the predictions of (x, y) and z are weighted by factors w_t so that the regression loss becomes

$$\mathcal{L}^{(t)}_{\text{DRAU}}(\hat{\mathbf{y}}, \hat{\boldsymbol{\alpha}}) = \mathcal{L}_{\text{haul}}(\hat{\alpha}_t, \hat{\mathbf{p}}^{(2\text{d})}_t, \mathbf{p}^{(2\text{d})}_t) + w_t \mathcal{L}_{\text{haul}}(\hat{\beta}_t, \hat{z}_t, z_t), \tag{4.9}$$

where the weight w_t is determined by the negative temporal difference of ground truth $z_{1:T}$ with softmax normalization:

$$w_t = \frac{\exp(-\Delta z_t)}{\sum_{t=1}^{T} \exp(-\Delta z_t)}, \quad \Delta z_t = |z_t - z_{t-1}|. \tag{4.10}$$

Since Δz_t indicates the stability of depth transition, the motivation of w_t is to encourage large weight on the stable depth transitions (small Δz_t) in a trajectory, which enables the training to focus less on the unstable depth so that the model is robust to noisy depth annotations.

Velocity Constraints To explicitly inject the physical rule of hand motion into the model, we additionally take the motion inertia into consideration. Specifically, we leverage the transitioned states $\{\mathbf{z}_1, \ldots, \mathbf{z}_T\}$ learned from Eq. (4.6) to predict the velocities $\{\mathbf{v}_1, \ldots, \mathbf{v}_T\}$ by an MLP. Then, we propose the following velocity constraint in training:

$$\begin{aligned} \mathcal{L}_{\text{velo}}(\hat{\mathbf{v}}, \mathbf{p}) = & \sum_{t=1}^{T} \left(\|\mathbf{p}_t - \mathbf{p}_{t-1} - \hat{\mathbf{v}}_t\|^2 \right) \\ & + \gamma \sum_{t=C+1}^{T} \left(\|\mathbf{p}_C + \sum_{i=C+1}^{t} \hat{\mathbf{v}}_i - \hat{\mathbf{p}}_t\|^2 \right), \end{aligned} \tag{4.11}$$

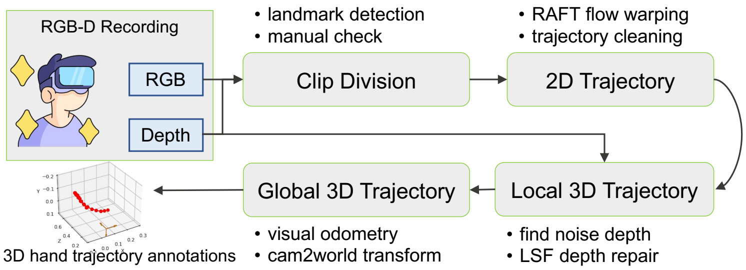


Figure 4.4 **Annotation Workflow.** For more details of the annotation procedure, please refer to our supplementary materials.

where the first term uses the first-order difference of locations $\mathbf{p}_t$ to supervise the predicted velocity $\hat{\mathbf{v}}$ and we set $\mathbf{p}_0$ to zero. The second term is to constrain the future predicted trajectory point $\hat{\mathbf{p}}_t$ with the warped point, which is computed by adding the accumulative predicted velocities onto the last observed point $\mathbf{p}_C$ since the time interval is one. Empirically, we found the velocity constraint enables better generalization capability to unseen data (see Table 4.3).

Visual Prompt Tuning Visual prompt tuning (VPT) [125] has been recently successful to adapt large visual foundation models to downstream vision tasks. In this paper, we leverage VPT to adapt pre-trained visual backbone $f_{\mathcal{V}}$ for the trajectory prediction task. The basic idea is to append learnable prompt embeddings $\mathbf{P}$ to the input image $\mathbf{I}$ and only learn a few layers of MLP head h_ψ while keeping the backbone parameters Ψ frozen as Ψ^* in training:

$$\mathbf{x}_t^{(\mathcal{V})} = h_\psi(f_{\mathcal{V}}^{\Psi^*}(\mathbf{I}_t, \mathbf{P})) \tag{4.12}$$

where only the head parameters ψ and visual prompt $\mathbf{P}$ are learned in training. Since $\{\psi, \mathbf{P}\}$ are much smaller than Ψ, the VPT is highly efficient in training. We implemented $f_{\mathcal{V}}$ with both ResNet [106] and ViT [65], without noting significant performance difference. However, applying VPT achieves better 3D hand trajectory prediction performance than traditional fine-tuning (see Fig. 4.8). This is interesting as there is no existing literature that explores the VPT for vision-based trajectory prediction problems.

4.4 Experiments

4.4.1 Datasets

Since there is no available egocentric 3D hand trajectory dataset, we collect annotations based on two existing datasets, i.e., H2O [162] and EgoPAT3D [181], which contain egocentric RGB-D raw recordings for annotation purpose.

H2O [162] dataset is initially collected for 3D hand pose and interaction recognition using RGB-D data from both egocentric and multiple third-person views. We first use the precisely annotated 3D hand poses to compute the 3D centroids as the ground truth of the 3D hand trajectory, named **H2O-PT**, which is guaranteed to be of high-quality in [162] by multi-view verification.

EgoPAT3D [181] dataset is much larger than H2O. It is initially collected for predicting the 3D action targets from egocentric 3D videos. However, it does not provide either the 3D hand trajectory or the 3D hand poses. Thus, similar to [181], we develop an annotation workflow as shown in Fig. 4.4. More details about the annotation workflow are in the supplement. Eventually, we obtained sufficiently large collections of 3D hand trajectories for training and evaluation, named **EgoPAT3D-DT**. To verify the reliability of the annotation workflow, we also apply it to H2O, resulting in a **H2O-DT** dataset.

Dataset Split The H2O(-PT/DT) dataset consists of 184 untrimmed long videos. We temporally sample the videos into multiple 64-frame clips with a step-size of 15 frames, resulting in 8203, 1735, and 3715 samples in training, validation, and testing splits, respectively. The EgoPAT3D-DT consists of 14 scenes and we split it into 11 seen scenes containing 8807 samples and 3 unseen testing scenes containing 2334 samples. The unseen scenes are not used in training, and the seen scenes are split into 6356, 846, and 1605 samples for training, validation, and seen testing.

Evaluation Setting We use the 3D Average Displacement Error (ADE) and Final Displacement Error (FDE) in meters as the evaluation metrics. The 2D trajectory results are normalized with reference to the video frame size. For all metrics, a small value indicates better performance. Each model is trained with 3D and 2D trajectory targets individually and evaluated by 3D metrics

Table 4.1 **ADE and FDE results on H2O-PT dataset**. All models are built with ResNet-18 backbone. Best and secondary results are viewed in bold **black** and **blue** colors, respectively.

Model	ADE (↓)			FDE (↓)		
	3D$_{(3D)}$	2D$_{(3D)}$	2D$_{(2D)}$	3D$_{(3D)}$	2D$_{(3D)}$	2D$_{(2D)}$
DKF [152]	0.159	0.186	0.211	0.137	0.163	0.185
RVAE [168]	0.046	0.055	0.056	0.067	0.081	**0.037**
DSAE [371]	0.051	0.060	0.059	0.057	0.067	0.076
STORN [13]	0.043	0.053	0.053	0.094	0.141	0.076
VRNN [43]	0.041	0.050	0.050	0.051	0.081	0.068
SRNN [75]	0.040	0.048	0.049	0.036	0.061	0.044
EgoPAT3D[1] [181]	0.039	0.046	0.048	**0.034**	0.064	0.060
AGF[1] [380]	0.039	0.046	0.081	0.069	0.065	0.146
OCT[1] [205]	0.252	0.311	0.387	0.278	0.471	0.381
ProTran[1] [305]	0.066	0.088	0.109	0.099	0.168	0.123
USST	**0.031**	**0.037**	**0.040**	0.052	**0.043**	0.043

(3D$_{(3D)}$) and 2D metrics (2D$_{(2D)}$), respectively. The 3D model is additionally evaluated by 2D metrics (2D$_{(3D)}$) by projecting the 3D trajectory outputs to the 2D image plane.

4.4.2 Implementation Detail

The proposed method is implemented by PyTorch. In pre-processing, RGB videos are downscaled to 64×64. The 3D global trajectory data are normalized and further centralized to the range [-1,1]. By default, we set the observation ratio to 60%, the feature dimensions of $\mathbf{o}^{(\mathcal{V})}$ and $\mathbf{o}^{(\mathcal{T})}$ to 256, and the dimension of $\mathbf{z}$ to 16 for all methods. In training, we use Huber loss to compute the location error. We adopt the Adam optimizer with base learning rate 1e-4 and cosine warmup scheduler for 500 training epochs on EgoPAT3D and 350 epochs on H2O datasets, respectively. More implementation details are in the supplement.

4.4.3 Main Results

Table 4.1 and 4.2 show the comparison between our method and existing sequential prediction approaches on H2O-PT and EgoPAT3D-DT datasets, respectively. The methods in the first multi-row section are general RNN-based while those in the second multi-row section show recent

[1] We adapted the task-specific outputs of EgoPAT3D, AGF, OCT, and ProTran to fulfill the 3D hand trajectory forecasting task in this paper.

Table 4.2 **ADE results on EgoPAT3D-DT dataset**. All models are built with ResNet-18 backbone. Best and secondary results are viewed in bold **black** and **blue** colors, respectively.

Model	Seen Scenes (↓)			Unseen Scenes (↓)		
	3D$_{(3D)}$	2D$_{(3D)}$	2D$_{(2D)}$	3D$_{(3D)}$	2D$_{(3D)}$	2D$_{(2D)}$
DKF [152]	0.294	0.237	0.157	0.260	0.202	0.133
RVAE [168]	0.216	0.110	0.121	0.194	0.104	0.109
DSAE [371]	0.214	0.129	0.143	0.188	0.116	0.131
STORN [13]	0.194	0.092	0.083	0.161	0.084	0.070
VRNN [43]	0.194	0.092	0.083	0.164	0.086	0.070
SRNN [75]	0.192	0.088	0.079	0.166	0.081	0.067
EgoPAT3D[1] [181]	0.186	**0.081**	**0.079**	0.170	0.080	0.068
AGF[1] [380]	6.149	0.136	0.099	6.045	0.119	0.087
OCT[1] [205]	0.853	0.163	0.098	0.782	0.139	0.091
ProTran[1] [305]	0.314	0.179	0.135	0.240	0.154	0.107
USST	**0.183**	0.089	0.082	**0.120**	**0.075**	**0.060**

Transformer-based models. We put the FDE results on EgoPAT3D-DT in the supplement. The tables show our method achieves the best ADE performance and comparable FDE results with AGF and SRNN on H2O-PT, and significantly outperforms AGF, OCT, and ProTran on EgoPAT3D-DT. The competitive performance of SRNN is because of its both forward and backward passes over time such that all future positional encodings are utilized for forecasting. We notice that AGF, OCT, and ProTran do not work well on EgoPAT3D-DT, potentially due to the KL divergence vanish issue. The higher performance on the unseen split than the seen split can be attributed to the less distribution shift between unseen test trajectories and the training trajectories.

4.4.4 Model Analysis

Ablation Study To validate the effectiveness of the proposed modules and loss functions, we report the results of the ablation study in Table 4.3 on the EgoPAT3D-DT dataset.

We first compare the ProTran with the proposed state space transformer (SST), which is a vanilla version of USST without uncertainty modeling, velocity constraint, and VPT. For a fair comparison, we implement a deterministic (det) version of ProTran in addition to the original method that uses stochastic variational inference (svi). Table 4.3 shows a clear advantage of our method over ProTran, which demonstrates the superiority of our SSM for state transition.

Table 4.3 **Ablation Study**. All models are trained with 3D targets and tested with both 3D and 2D ADE.

Variants	Seen (↓)		Unseen (↓)	
	3D	2D	3D	2D
ProTran (svi) [305]	0.314	0.179	0.240	0.154
ProTran (det) [305]	0.201	0.104	0.195	0.106
SST (ours)	**0.190**	**0.088**	**0.174**	**0.084**
USST w/o. $\mathcal{L}_{\text{haul}}$	0.292	0.237	0.267	0.204
USST w/o. $\mathcal{T}_{1:C}$	0.244	0.176	0.267	0.208
USST w/o. $\mathbf{p}_{t-1}$	0.196	0.090	0.169	0.098
USST w/o. $\mathcal{L}_{\text{velo}}$	0.189	0.091	0.168	0.099
USST w/o. w_t	0.183	0.090	0.130	0.077
USST (full model)	**0.183**	**0.089**	**0.120**	**0.075**

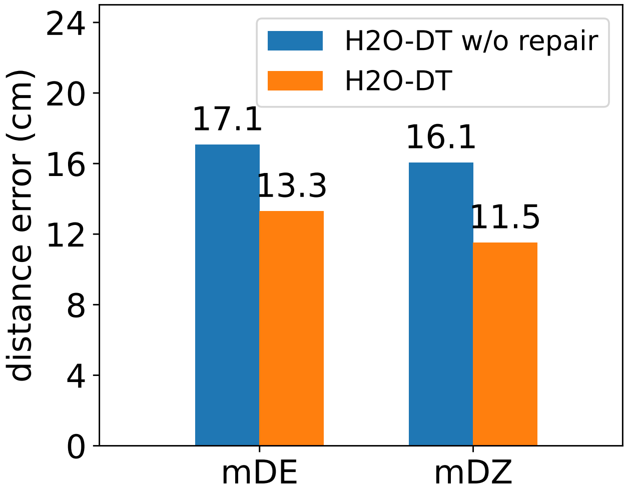


Figure 4.5 **Effect of depth repair**. mDE and mDZ are the mean errors of 3D displacement and depth, respectively.

Next, in Table 4.3, we individually remove the new components and compare them with the full model of USST, including 1) the uncertainty loss function $\mathcal{L}_{\text{haul}}$, 2) the trajectory context $\mathcal{T}_{1:C}$, 3) the predictive link in $p_\phi(\mathbf{p}_t|\mathbf{z}_t, \mathbf{p}_{t-1})$, 4) the velocity constraint $\mathcal{L}_{\text{velo}}$, and 5) the depth robust weight w_t. It shows that uncertainty modeling is critical to guarantee reasonable forecasting results. Without historical trajectory $\mathcal{T}_{1:C}$, as expected, the performance degradation is significant. The predictive link from $\mathbf{p}_{t-1}$ to $\mathbf{p}_t$ is also important for the forecasting problem, which is consistent with the recent finding in [91]. It is noticeable that the velocity constraint shows a larger performance gain (4.8cm of 3D trajectory) on the unseen test set than on the seen data (0.6cm of 3D trajectory), revealing

Table 4.4 **Annotation Reliability**. ADE results (3D$_{(3D)}$ / 2D$_{(2D)}$) are from testing on the same H2O-PT test set.

Metrics		SRNN	USST
H2O-**DT**	ADE	0.087 / 0.076	0.033 / 0.041
	FDE	0.124 / 0.045	0.052 / 0.041
H2O-**PT**	ADE	0.040 / 0.049	0.031 / 0.040
	FDE	0.036 / 0.044	0.052 / 0.043

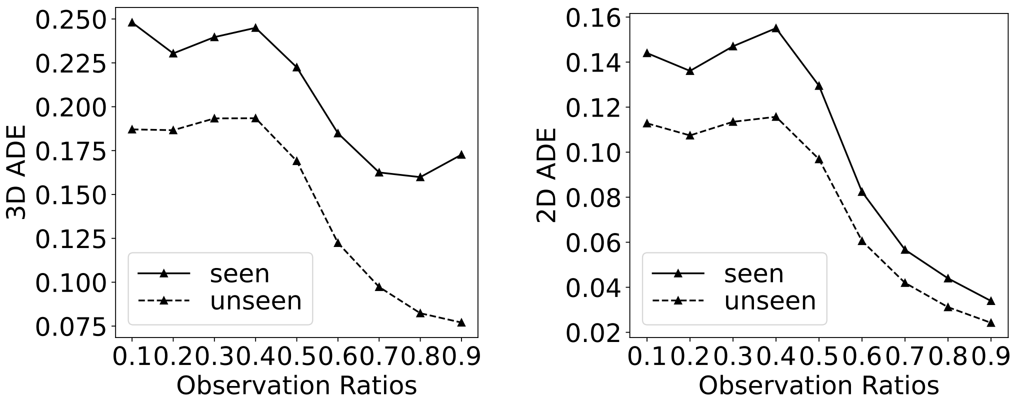


Figure 4.6 **Arbitrary Observation Ratios.** We report the results of 3D ADE (left) and 2D ADE (right) on EgoPAT3D-DT dataset.

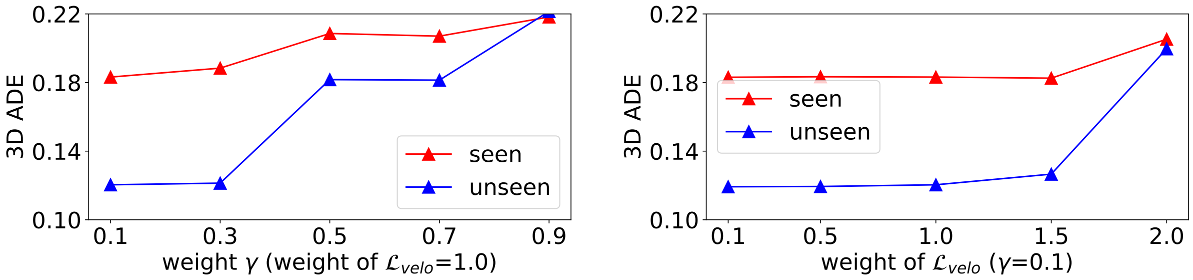


Figure 4.7 **Impact of loss weights.** Left: set the weight of $\mathcal{L}_{\text{velo}}$ to 1.0 and tune γ. Right: set γ to 0.1 and tune the weight of $\mathcal{L}_{\text{velo}}$.

the importance of the physical rule for generalizable trajectory prediction. Lastly, the depth robust weight w_t (Eq. (4.10)) also boosts the performance of unseen data, showing the importance of modeling the depth noise from annotations.

Annotation Reliability By using the accurate H2O-PT as a reference, in Fig. 4.5, we show the effect of repairing depth of the 3D trajectory annotations from raw RGB-D data. We see a clear

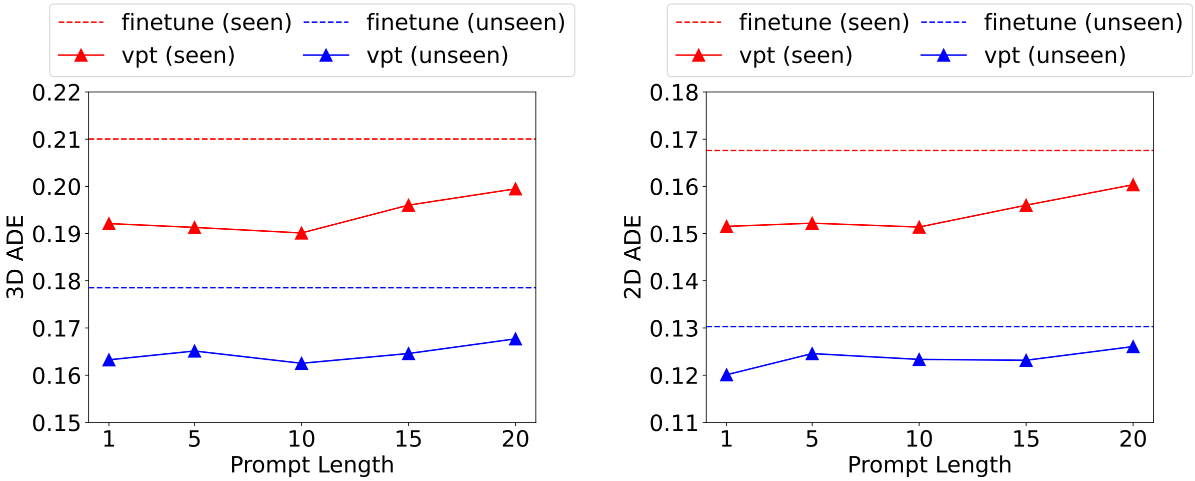


Figure 4.8 **Impact of Prompt Length.** We report the results of 3D (left) and 2D (right) ADE on EgoPAT3D-DT. The “finetune (unseen)" means finetune model on seen but test on unseen scenes.

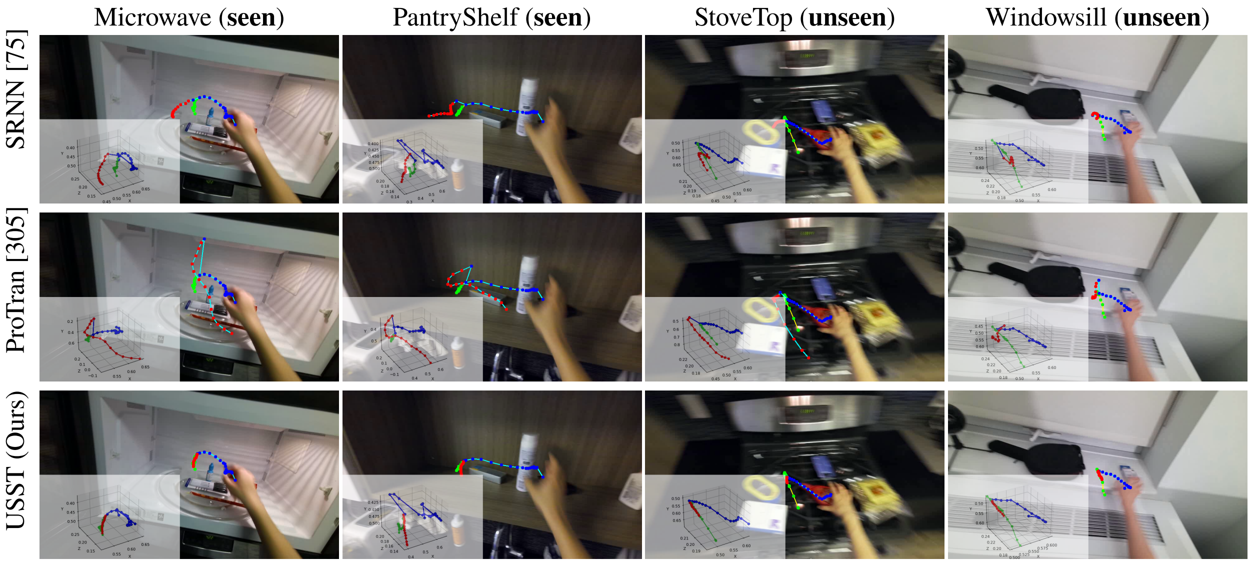


Figure 4.9 **Visualization on EgoPAT3D.** For each example (in a column), we show the global 3D trajectory and its 2D projection on the first frame. The blue, green, and red trajectory points represent the past observed, future ground truth, and future predictions, respectively.

improvement in mDE and mDZ measurements. In Table 4.4, we further show the performance impact of annotation quality on SRNN and our USST models. It shows that our USST achieves more consistent ADE and FDE results than SRNN over the H2O-PT and H2O-DT. For reference, in the supplement, we additionally report the full results and analysis on H2O-DT and H2O-DT w/o repair.

Table 4.5 **Impact of 3D coordinate systems.** “Local" and “global" mean using 3D camera and world coordinates, respectively.

3D Target	Backbone	Seen (↓)		Unseen (↓)	
		3D	2D	3D	2D
Local	R18	0.202	**0.083**	0.174	**0.062**
Global	R18	**0.183**	0.089	**0.120**	0.075
Local	ViT	0.183	**0.081**	0.133	**0.067**
Global	ViT	**0.182**	0.087	**0.119**	0.075

Forecast at Any Time To simulate the real-world practice that forecasting trajectory at an arbitrary time, we take the advantage of the Transformer attention mask to fulfill random observation ratios ranging from 10% to 90%. The results are summarized in Fig. 4.6. It shows that with more percentage of information observed, both the 2D and 3D forecasting error are reduced as expected. It is interesting to see the slight increase of 3D ADE for the seen test data when using more observations. It could be caused by more inaccurate trajectory depth values at the end of trajectories.

Loss Weights Fig. 4.7 shows the EgoPAT3D-DT results of tuning the weights in Eq. (4.11), where the best performance is achieved when γ is set to 0.1 and the weight of $\mathcal{L}_{\text{velo}}$ is set to 1.0, respectively. We apply them to H2O-PT by default.

Prompt Length of VPT As indicated in VPT literature [125], the length of the visual prompt in ViT models needs to be carefully tuned for downstream tasks. In experiments, based on the SST model, we select the prompt length from $\{1, 5, 10, 15, 20\}$ and compare their performance with the baseline that fine-tunes the entire ViT backbone. Results are reported in Fig. 4.8. It shows that VPT could steadily achieve lower 2D and 3D ADE than fine-tuning, and the best performance is achieved when the prompt length is 10.

Local vs Global 3D Trajectory We note that the ambiguity of learning the appearance-location mapping exists when using local 3D targets. To justify the choice of global 3D trajectory targets, in Table 4.5, we compare the 3D and 2D ADE results using both ResNet-18 and ViT as visual backbones. It clearly shows that for 3D trajectory prediction, a global 3D coordinate system is a better choice, while for 2D trajectory evaluation, the local 3D target is better. These observations are expected as in the local 3D coordinate system, the projected 2D pixel locations of moving hands tend to be in the visual center due to the egocentric view so that the model training is dominated by the 2D hand locations.

Qualitative Results As shown in Fig. 4.9, the proposed USST model is compared with the Transformer-based approach ProTran and the most competitive method SRNN. It clearly shows that our trajectory forecasting is much better than the three compared methods.

Limitations & Discussions The dataset annotation is limited in scenarios when the RGB-D sensors or camera poses are not available. The model is limited in the recursive way of state transition, which is not hardware-friendly for parallel inference. Besides, in the future, other tasks like the 3D hand pose and interaction recognition can be jointly studied for a fine-grained egocentric understanding.

4.5 Conclusion

In this paper, we propose to forecast human hand trajectory in 3D physical space from egocentric videos. For this goal, we first develop a pipeline to automate the 3D trajectory annotation. Then, we propose a novel uncertainty-aware state space transformer (USST) model to fulfill the task. Empirically, with the aleatoric uncertainty modeling, velocity constraint, and visual prompt tuning, our model achieves the best performance on both H2O and EgoPAT3D datasets and good generalization to the unseen scenes.

4.6 Supplementary Material

In this section, we provide more details of the data collection and annotation, model implementation, and evaluation results.

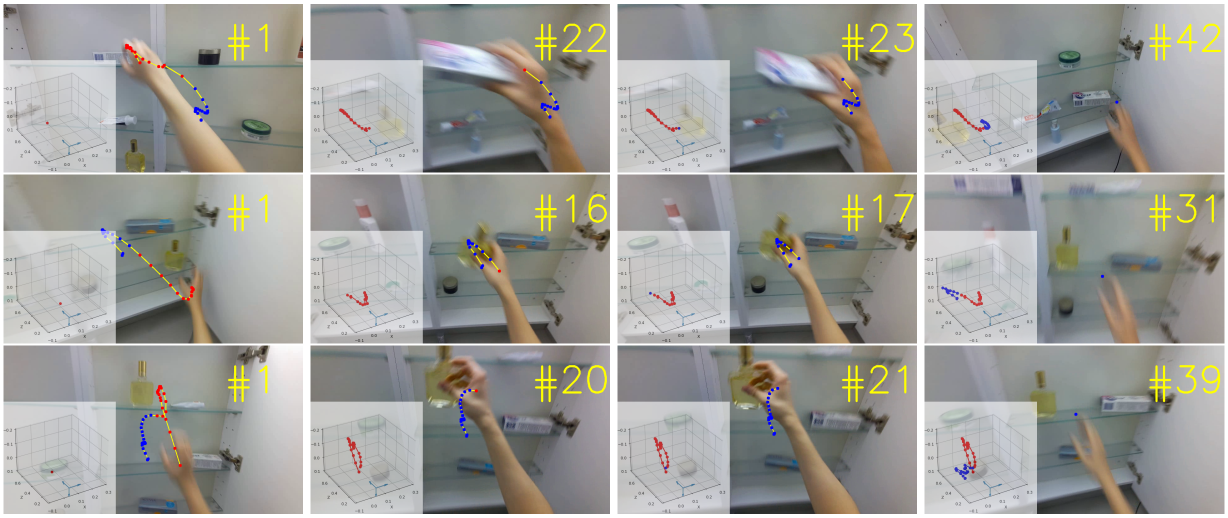


Figure 4.10 **Dataset Examples**. For each video (in a row), the global 3D trajectory and the projected 2D trajectory are visualized, where the past and future trajectory segments are in red and blue, respectively. Zoom in for more details.

4.6.1 Details of the Datasets

4.6.1.1 Annotation Workflow

Following the similar pipeline in the EgoPAT3D [181], we propose to obtain the 3D hand trajectory annotations based on egocentric RGB-D recordings. In the following paragraphs, we elaborate on each processing step based on RGB-D data from the EgoPAT3D [181] and H2O [162].

Clip Division The EgoPAT3D dataset consists of RGB-D data of hand-object manipulation in 14 indoor scenes. We leverage the provided manual clip divisions and the hand landmarks to obtain more accurate trajectory divisions. Specifically, let (s_m, e_m) denote the start and end of a manually annotated trajectory, and $\{t_s, \ldots, t_e\}$ denote the indices of detected 3D hand landmarks, our trajectory start and end are determined by $\max(s_m, t_s)$ and $\min(e_m, t_e)$, respectively. This technique could mitigate the ambiguity of trajectory start and end. Then, we use them to obtain the RGB video clips from the raw recordings. The H2O dataset contains 184 long videos and each video is annotated with 3D poses of the left and right hand as well as the binary validity flag. The trajectory start and end are determined by the validity flag.

2D Trajectory For each clip, we found the hand trajectory is not stable if only using the centers of frame-wise hand landmarks as trajectory points. Therefore, for the EgoPAT3D dataset, we propose to leverage the optical flow model RAFT [308] to warp the hand landmark center as the 2D hand trajectory. Specifically, we apply the RAFT to the forward pass starting from the first 2D location $\mathbf{p}_1$ of the hand and backward pass starting from the last location $\mathbf{p}_T$ of the hand, resulting in the forward trajectory $\{\mathbf{p}_t^{(f)}\}_{t=1}^T$ and backward trajectory $\{\mathbf{p}_t^{(b)}\}_{t=1}^T$. Then, for each frame t, the ultimate 2D location is determined by a temporally weighted sum $\tilde{\mathbf{p}}_t = w_t\mathbf{p}_t^{(f)} + (1 - w_t)\mathbf{p}_t^{(b)}$ where the weight w_t is temporally decreasing from 1.0 to a constant c by $w_t = c + (1 - c)/(1 + \exp(t - T/2))$. In practice, we set c to 0.3. The rationale of weighing is to mitigate the error accumulation from the RAFT model. It assigns more weight to the earlier locations by forward flow and more weight to the latter locations by backward flow, with the margin c between the two passes.

Local 3D Trajectory With the 2D hand trajectory, it is straightforward to obtain the 3D hand trajectory by fetching the depth of each trajectory point from the RGB-D clips. However, we noticed that due to the fast motion of the hand and camera, the recorded depth channels in those frames could be missing, i.e., depth values are zeros (see the red dots in Fig. 4.11). To obtain high-quality 3D hand trajectory annotations, we initially attempted to use the state-of-the-art depth estimation model NewCRFs [378] to estimate the missing depths from RGB frames. However, it cannot work well due to the camera motion that results in dynamic scenes in RGB frames. Instead, we found that a simple least-square fitting (LSF) by combining the third-order polynomial and sine functions, i.e., $z_\alpha(t) = \alpha_1 t^3 + \alpha_2 t^2 + \alpha_3 t + \alpha_4 + \alpha_5 \sin(\alpha_6 t)$, could repair the missing depth. For both EgoPAT3D and H2O, we apply the LSF to repair 3D hand trajectory depth. To enable successful depth fitting, we use at least 10 valid trajectory points to fit a multinomial model on each 3D hand trajectory that contains invalid depths.

Global 3D Trajectory Note that the 3D trajectory points from the previous step are defined in the local camera coordinate system. When the camera is moving in an egocentric view, using RGB videos to predict the local 3D trajectory will be ambiguous. In other words, distinct visual contents

Table 4.6 Summary of camera intrinsics

EgoPAT3D	resolution	$H = 2160, \quad W = 3840$
	focal length	$f_x = 1808.203, \quad f_y = 1807.946$
	principle point	$o_x = 1942.287, \quad o_y = 1123.822$
H2O	resolution	$H = 720, \quad W = 1280$
	focal length	$f_x = 636.659, \quad f_y = 636.252$
	principle point	$o_x = 635.284, \quad o_y = 366.874$

are forced to learn to predict numerically similar coordinates. To eliminate the ambiguity, similar to EgoPAT3D [181], we propose to transform the 3D trajectory targets into a global world coordinate system with reference to the first frame. This is a visual odometry procedure that computes the 3D homogeneous transformation $\mathbf{M}_t \in \mathbb{R}^{4\times4}$ between camera poses at two successive frames $t-1$ and t. Eventually, a local 3D trajectory point $\mathbf{p}_t^l$ is transformed as a global 3D trajectory point $\mathbf{p}_t^g$ by the accumulative matrix product $\mathbf{p}_t^g = \prod_{k=1}^t \mathbf{M}_k \mathbf{p}_t^l$. In experiments, we use the global 3D trajectory $\{\mathbf{p}_t^g\}_{t=1}^T$ as the ground truth for model training, evaluation, and visualization by default. Fig. 4.10 shows three video examples with global 3D trajectory annotations.

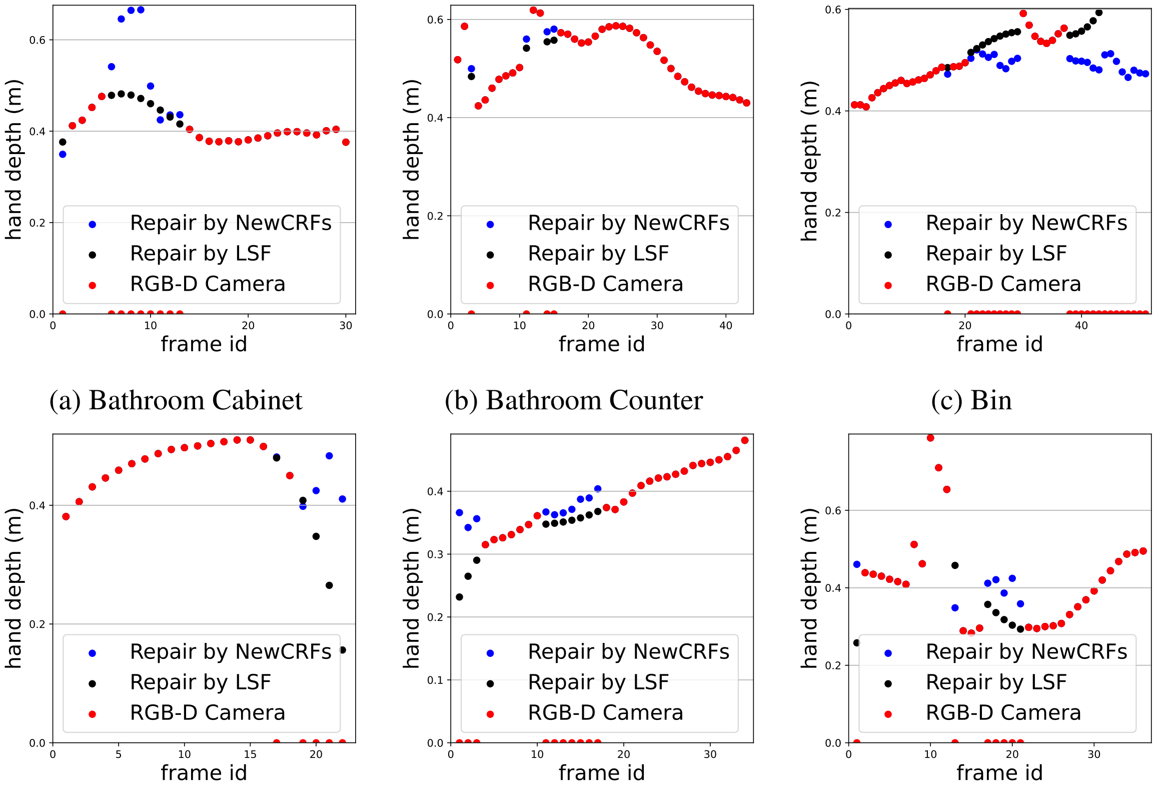
4.6.1.2 Camera Intrinsics and Poses

For both the EgoPAT3D and H2O, the camera intrinsics are fixed across all samples. Table 4.6 summarizes the camera intrinsics of the dataset we used in this paper. Note that the intrinsics are scaled with the factor 0.25 when we down-scale the RGB videos to the input resolution. For camera poses of EgoPAT3D, we use Open3D [410] library to perform visual odometry[1] by using adjacent RGB-D pairs so that the camera motion is obtained. The camera poses of H2O dataset are given for each video frame.

4.6.2 Additional Implementation Details

Data Structure To enable efficient parallel training with batches of data input that contain videos of varying lengths, we adopt the mask mechanism in our implementation. Specifically, we set the maximum length of each video to 40 and 64 for EgoPAT3D and H2O, respectively. The lengths of the past observation and future frames are determined by the actual video length. For instance,

[1] In practice, we followed the EgoPAT3D to use the Open3D API (RGBDOdometryJacobianFromHybridTerm) to compute the 3D camera motion.



(a) Bathroom Cabinet

(b) Bathroom Counter

(c) Bin

(d) Kitchen Cupboard

(e) Microwave

(f) Nightstand

Figure 4.11 Examples of comparison between the Least Square Fitting (LSF) and the depth estimation model NewCRFs [378] for repairing the noisy depth values from EgoPAT3D RGB-D data. It's clear that on this video dataset with dynamic background, a simple LSF with a multinomial model could achieve a much better depth repairing effect than the state-of-the-art deep learning model NewCRFs.

when the observation ratio is set to 0.6, a sample with 35 frames in total has 21 observed frames, 14 unobserved frames, and 5 zero-padded frames. Since the visual background of RGB videos is relatively clean, we resize videos into the size of 64 × 64 in training and inference.

Model Structure For the ResNet-18 backbone, we replace the global pooling layer after the last residual block with `torch.flatten`, in order to preserve as much visual contextual information as possible. When the visual prompt tuning (VPT) is utilized, the width of the padded learnable pixels is set to 5 as suggested by [125], resulting in 1380[2] additional parameters to learn. For the ViT backbone, we adopt the `vit/b16-224` architecture provided by TIMM, which is pre-trained on the

[2] For 64 × 64 input, the number of learnable parameters in prompt embeddings is computed by $(64+5\times 2)^2 - 64^2 = 1380$.

Table 4.7 **Results of models training on H2O-DT dataset**. We report all results of models trained by annotations from H2O-DT (left) and its version without depth repair (right), and tested on the accurate H2O-PT test set. All models are built with ResNet-18 backbone. Best and secondary results are viewed in bold **black** and **blue** colors, respectively.

Models	H2O-DT						H2O-DT (w/o depth repair)					
	ADE (↓)			FDE (↓)			ADE (↓)			FDE (↓)		
	3D$_{(3D)}$	2D$_{(3D)}$	2D$_{(2D)}$	3D$_{(3D)}$	2D$_{(3D)}$	2D$_{(2D)}$	3D$_{(3D)}$	2D$_{(3D)}$	2D$_{(2D)}$	3D$_{(3D)}$	2D$_{(3D)}$	2D$_{(2D)}$
DKF [152]	0.236	0.235	0.269	0.138	**0.030**	**0.020**	0.199	0.186	0.208	0.181	0.153	0.187
RVAE [168]	0.125	0.209	0.094	0.057	0.082	0.047	0.051	0.060	0.059	0.058	0.071	0.059
DSAE [371]	0.081	0.113	0.078	0.043	0.059	0.040	0.072	0.068	0.063	0.067	0.047	0.077
STORN [13]	0.091	0.100	0.070	0.245	0.078	0.040	0.067	0.061	0.054	0.135	0.097	0.121
VRNN [43]	0.080	0.092	0.068	0.042	0.035	0.039	0.065	0.063	0.054	0.133	0.087	0.087
SRNN [75]	0.087	0.097	0.076	0.124	0.072	0.045	0.055	0.059	0.061	0.083	0.089	0.135
AGF [380]	0.108	0.065	0.080	0.171	0.061	0.214	0.099	0.075	0.065	0.186	0.044	0.056
OCT [205]	0.360	0.473	0.350	0.348	0.362	0.520	0.381	0.519	0.403	0.403	0.521	0.505
ProTran [305]	0.080	0.082	0.099	**0.023**	0.031	0.107	0.070	0.093	0.064	0.162	0.146	0.041
USST	**0.033**	**0.041**	**0.041**	0.052	0.050	0.041	**0.032**	**0.041**	**0.040**	**0.053**	**0.041**	**0.041**

ImageNet-21K dataset. For either ResNet-18 or ViT-based frame encoder $f_{\mathcal{V}}$, the output feature is embedded by a two-layer MLP with 512 and 256 hidden units. For the trajectory encoder $f_{\mathcal{T}}$, we use a two-layer MLP with 128 and 256 hidden units. For both visual and trajectory transformer encoders, we utilize the standard transformer encoder architecture, which consists of 6 multi-head self-attention blocks where the number of heads is 8 and the MLP ratio is 4. For the decoder, we implement the three prediction branches, i.e., future trajectory prediction, uncertainty prediction, and velocity prediction, using three MLP heads, each of which consists of 128 and 3 hidden units. For trajectory and velocity prediction outputs, we use `tanh` activation, while for the uncertainty output, we use `softplus` activation. Besides, for the velocity prediction, layer normalization is applied to each hidden layer.

Learning and Inference In training, we set the δ parameter of Huber loss to $1e-5$, and set the γ coefficient of the velocity-based warping loss to 0.1. For the cosine learning rate scheduler, we adopt warm-up training in the first 10 epochs. For 500 training epochs in total, our model training can be completed within 5 hours on a single RTX A6000 GPU. In testing, we evaluate the predicted 3D trajectory in the global coordinate system by referring to the camera at the first time step, while

Table 4.8 **FDE results of 2D hand trajectory forecasting**. Compared models are built with ResNet-18 (R18) backbone. Best and secondary results are in bold **black** and blue colors, respectively.

Model	Seen (↓)	Unseen (↓)
DKF [152]	0.150	0.239
RVAE [168]	0.152	0.201
DSAE [371]	0.144	0.233
STORN [13]	0.145	0.266
VRNN [43]	0.155	0.237
SRNN [75]	0.157	0.198
OCT [205]	0.090	0.147
ProTran [305]	0.134	**0.049**
USST (R18)	0.075	0.107
USST (ViT)	**0.066**	0.114

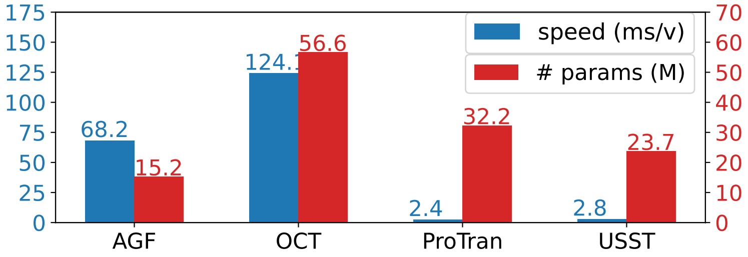


Figure 4.12 Inference speed in milliseconds/video (ms/v) and the number of model parameters in million (M), tested on a single RTX 6000Ada GPU with input video size 64 × 64 × 64.

visualizing the 2D trajectory by first projecting the global 3D trajectory into the local 3D trajectory, and then projecting the local 3D coordinates onto a video frame as 2D pixel coordinates.

4.6.3 Additional Evaluation Results

Full results on H2O-DT We additionally provide full experimental results by training models on **H2O-DT** and **H2O-DT** w/o depth repair in Table 4.7. It shows that our USST method could still achieve the best performance using training data with inaccurate trajectory annotations.

FDE results on EgoPAT3D-DT We additionally provide the Final Displacement Error (FDE) results for 2D hand trajectory forecasting as shown in Table 4.8. Our method could achieve the best performance on the seen test data while being competitive on the unseen test data. Besides, ProTran shows the best result on the unseen data, which could be attributed to its extra trajectory supervision from the full observation of the latent Gaussian distributions.

Inference speed In Fig. 4.12, we compared with the Transformer-based methods. It shows the USST achieves competitive speed to ProTran while comparable model size to AGF. With certain improvements, our method could potentially benefit the rendering latency in AR/VR.

CHAPTER 5

OPEN-SET ACTION RECOGNITION

5.1 Introduction

Video action recognition aims to classify a video that contains a human action into one of the pre-defined action categories (closed set). However, in a real-world scenario, it is essentially an *open set* problem [291], which requires the classifier to simultaneously recognize actions from known classes and identify actions from unknown ones [278, 86]. In practice, open set recognition (OSR) is more challenging than closed set recognition, while it is important for applications such as face recognition [207], e-commerce product classification [353], autonomous driving [272], and so on.

OSR was originally formalized in [278] and many existing approaches have been proposed using image datasets such as MNIST [166] and CIFAR-10 [156]. However, unlike OSR, limited progress has been achieved for open set action recognition (OSAR) which is increasingly valuable in practice. In fact, novel challenges arise in OSAR from the following key aspects. First, the temporal nature of videos may lead to a high diversity of human action patterns. Hence, an OSAR model needs to capture the temporal regularities of closed set actions but also be *aware of what it does not know* when presented with unknown actions from an open set scenario. Second, the visual appearance of natural videos typically contains *static biased* cues [183, 42] (e.g., “surfing water" in totally different scenes as shown in Fig. 5.2). Without addressing the temporal dynamics of human actions, the static bias could seriously hamper the capability of an OSAR model to recognize unknown actions from an unbiased open set. Due to these challenges, existing effort on OSAR is quite limited with few exceptions [291, 153, 366]. They simply regard each video as a standalone sample and primarily rely on image-based OSR approaches. As a result, they fall short in addressing the inherent video-specific challenges in the open set context as outlined above.

This chapter is adapted from the following publication:
"Wentao Bao, Qi Yu, and Yu Kong. Evidential deep learning for open set action recognition. In *International Conference on Computer Vision (ICCV)*, Oral, 2021."

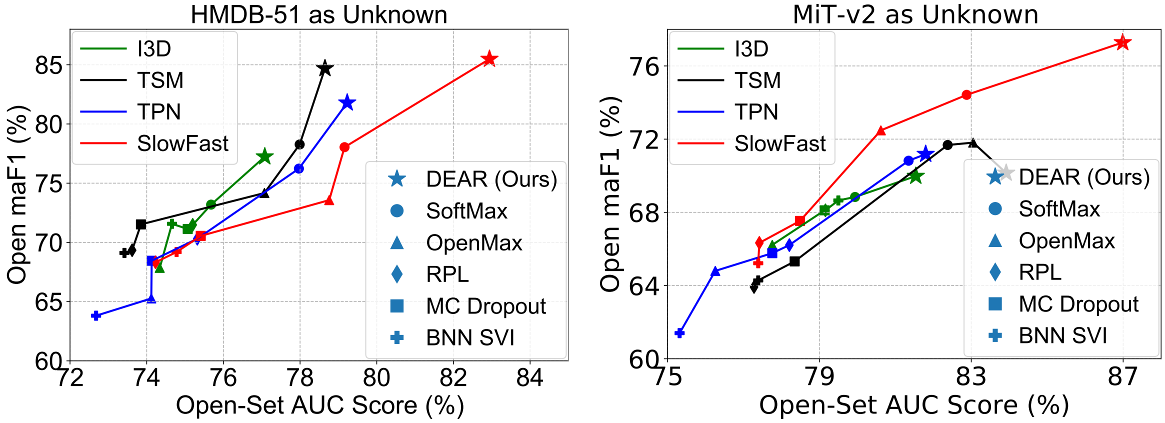
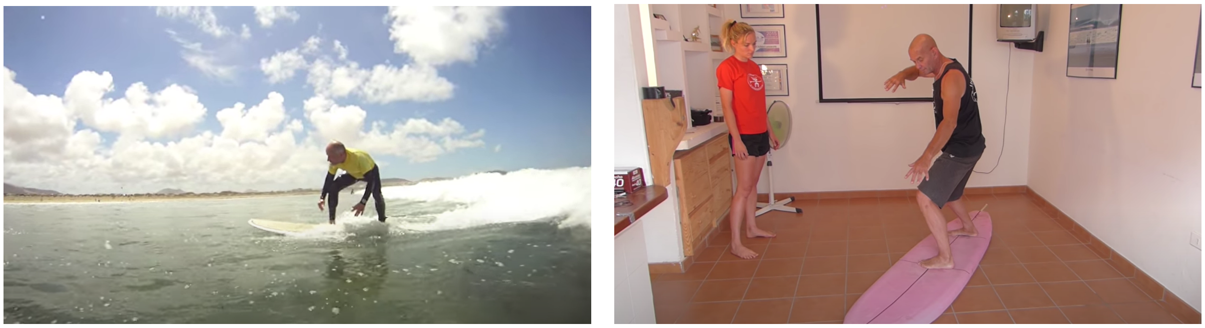


Figure 5.1 **Open Set Action Recognition Performance.** HMDB-51 [159] and MiT-v2 [230] are separately used as small- and large-scale unknown data for models trained on the closed set UCF-101 [296]. Our DEAR method (★) significantly outperforms existing approaches on multiple action recognition models.

In this paper, we propose a Deep Evidential Action Recognition (DEAR) method for the open-set action recognition task. To enable the model to "*know unknown*" in an OSAR task, our method formulates it as an uncertainty estimation problem by leveraging evidential deep learning (EDL) [283, 401, 287, 4, 282]. EDL utilizes deep neural networks to predict a Dirichlet distribution of class probabilities, which can be regarded as an evidence-collection process. The learned evidence is informative to quantify the predictive uncertainty of diverse human actions so that unknown actions would incur high uncertainty, i.e., the model knows the unknown. Furthermore, to overcome the potential over-fitting risk of EDL in a closed set, we propose a novel model calibration method to regularize the evidential learning process. Besides, to mitigate the static bias problem for video actions, we propose a plug-and-play module to debias the learned representation through contrastive learning. Benefiting from the evidential theory, our DEAR method is practically flexible to implement and provides a principled way to quantify the uncertainty for identifying unknown actions. Experimental results show that the DEAR method boosts the performance of existing powerful action recognition models with both small and large-scale unknown videos (see Fig. 5.1), while still maintaining a high performance in traditional closed set recognition settings.

Distinct from existing OSR methods [291, 153], the proposed DEAR is the first evidential learning model for large-scale video action recognition. DEAR is superior to existing Bayesian



(a) Kinetics [27]

(b) Mimetics [339]

Figure 5.2 **An Example of Static Bias.** To recognize the human action (i.e., “Surfing Water"), the recognition model which is biased to the background of water and sky in the closed set (Kinetics) would be unable to recognize the same action with the indoor scene in open set (Mimetics as unknown).

uncertainty-based methods [153] in that model uncertainty can be directly inferred through evidence prediction that avoids inexact posterior approximation or time-consuming Monte Carlo sampling [4]. Moreover, our proposed model calibration method ensures that DEAR is confident in accurate predictions while being uncertain about inaccurate ones. Compared to [291] which incrementally learns a classifier for unknown classes, our method is more flexible in training without access to unknown actions. Moreover, our proposed debiasing module could reduce the detrimental static bias of video actions so that the model is robust to out-of-context actions in the open set setting.

In summary, the contribution of this paper is threefold:

- Our Deep Evidential Action Recognition (DEAR) method performs novel evidential learning to support open-set action recognition with principled and efficient uncertainty evaluation.
- The proposed Evidential Uncertainty Calibration (EUC) and Contrastive Evidential Debiasing (CED) modules effectively mitigate over-confident predictions and static bias problems, respectively.
- The DEAR method is extensively validated and consistently boosts the performance of state-of-the-art action recognition models on challenging benchmarks.

5.2 Related Work

Open Set Recognition. OSR problem originates from face recognition scenario [171] and it is firstly formalized by Scheirer et al. [278]. In [278], to reject the unknown classes, a binary support vector machine (SVM) was introduced by adding an extra hyper-plane for each new class. Based on this work, the Weibull-calibrated SVM (W-SVM) [279] and P_I-SVM [123] are further proposed to calibrate the class confidence scores by leveraging the statistical extreme value theory (EVT). With the recent success of deep learning, deep neural networks (DNN) are widely used in OSR problems. To overcome the drawbacks of softmax in DNN, Bendale et al. [16] proposed OpenMax to upper-bound the open space risk for DNN models. Based on this work, G-OpenMax [83] adopted a generative method to synthesize unknown samples in the training of DNNs. Similarly, recent deep generative adversarial networks (GANs) were used to generate samples of unknown class for OSR task [240, 64]. To reject the unknown, variational auto-encoder (VAE) was recently used to learn the reconstruction error in OSR task [249, 372, 302]. Different from these methods, our method is the first work to introduce evidential deep learning (EDL) for the OSR task and show the advantage over existing approaches.

For the open set action recognition (OSAR) problem, it is much more challenging than the OSR problem while only a few existing literature explored it. Shu et al. [291] proposed ODN by incrementally adding new classes to the action recognition head. To capture the uncertainty of unknown classes, Bayesian deep learning is recently introduced to identify the unknown actions in [153, 298, 154]. Busto et al. [21] proposed an open-set domain adaptation method. However, existing methods ignore the importance of uncertainty calibration and static bias of human actions in video data. In a broader context, uncertainty-based OSR is also closely related to out-of-distribution (OOD) [326]. Other less related topics such as anomaly detection [252], generalized zero-shot learning [221], and open world learning [15] are out of the scope in this paper and comprehensively reviewed in [86].

Deep Learning Uncertainty. To distinguish between the unknown and the known samples, an appropriate OOD scoring function is important. A recent line of research works [153, 218,

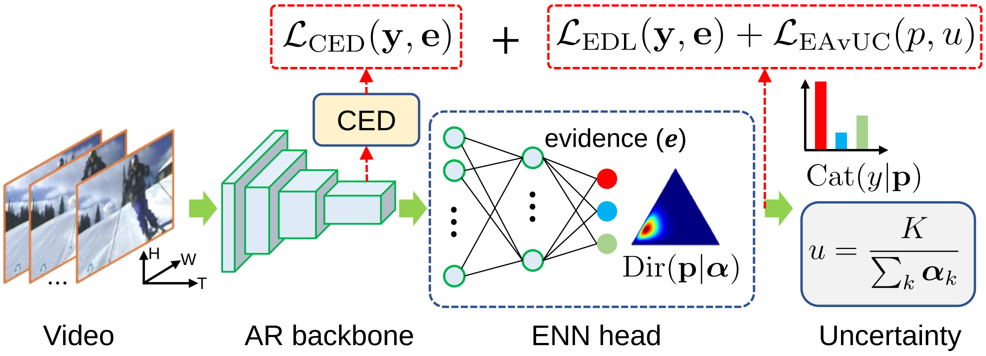

$\mathcal{L}_{\text{CED}}(\mathbf{y}, \mathbf{e})$
+
$\mathcal{L}_{\text{EDL}}(\mathbf{y}, \mathbf{e}) + \mathcal{L}_{\text{EAvUC}}(p, u)$
CED
evidence ($\boldsymbol{e}$)
$\text{Cat}(y|\mathbf{p})$
H
W
T
...
$\text{Dir}(\mathbf{p}|\boldsymbol{\alpha})$
$u = \frac{K}{\sum_k \boldsymbol{\alpha}_k}$
Video
AR backbone
ENN head
Uncertainty

Figure 5.3 Proposed DEAR method. We use 3-class ($K = 3$) action recognition (AR) for illustration. On top of the AR backbone, the Evidential Neural Network (ENN) head predicts the evidence $\mathbf{e}$ to build the Dirichlet distribution of class probability $\mathbf{p}$. The evidential uncertainty (u) from the Dirichlet is used for rejecting the unknown in open-set testing.

31, 287, 282] show that the predictive uncertainty learned by deep neural networks (DNN) can be a promising scoring function to identify OOD samples. It is assumed that OOD samples should be highly uncertain during inference. Bayesian neural networks (BNN) has been introduced to model the epistemic and aleatoric uncertainty for multiple computer vision tasks [136, 151, 315]. However, BNN is limited by the intractability of exact posterior inference, the difficulty of choosing suitable weight priors, and the expensive sampling for uncertainty quantification [4]. Recently, evidential deep learning (EDL) has been developed by incorporating the evidential theory into deep neural networks with promising results in both classification [283] and regression [4] tasks. In this paper, to the best of our knowledge, we are the first to incorporate evidential learning for large-scale and uncertainty-aware action recognition.

Video Action Recognition. Video action recognition has been widely studied in closed set setting [344, 148, 390]. In this paper, we select several representative and powerful methods, including the 3D convolution method I3D [27], the 2D convolution method TSM [193], the two-stream method SlowFast [74], and the method focusing on neck structure of a recognition model TPN [360]. Note that our method can be easily applied to any existing video action recognition models to enable them for open-set action recognition.

5.3 Approach

Overview. The proposed DEAR method is illustrated in Fig. 5.3. Given a video as input, the Evidential Neural Network (ENN) head on top of an Action Recognition (AR) backbone[1] predicts the class-wise evidence, which formulates a Dirichlet distribution so that the multi-class probabilities and predictive uncertainty of the input can be determined. For the open set inference, high uncertainty videos can be regarded as unknown actions while low uncertainty videos are classified by the learned categorical probabilities. The model is trained by Evidential Deep Learning (EDL) [283] loss regularized by our proposed Evidential Uncertainty Calibration (EUC) method. In training, we also propose a plug-and-play Contrastive Evidence Debiasing (CED) module to debias the representation of human actions in videos.

5.3.1 Deep Evidential Action Recognition

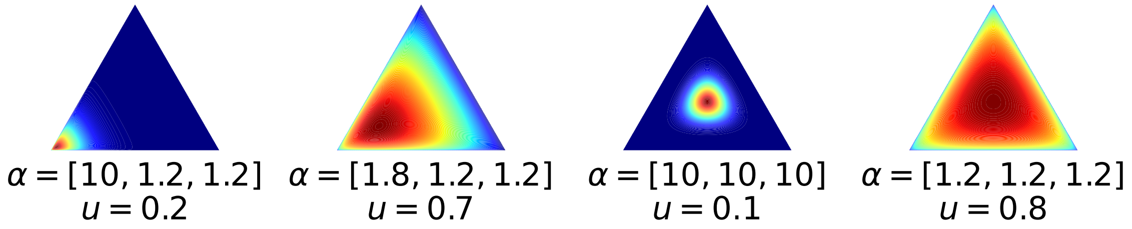
Background of Evidential Deep Learning. Existing deep learning-based models typically use a softmax layer on top of deep neural networks (DNNs) for classification. However, these softmax-based DNNs are not able to estimate the predictive uncertainty for a classification problem because the softmax score is essentially a point estimation of a predictive distribution [78] and the softmax outputs tend to be over-confident in false prediction [99].

Recent evidential deep learning (EDL) [283, 4] was developed to overcome the limitations of softmax-based DNNs by introducing the evidence framework of Dempster-Shafer Theory (DST) [284] and the subjective logic (SL) [130]. EDL provides a principled way to jointly formulate the multi-class classification and uncertainty modeling. In particular, given a sample $\mathbf{x}^{(i)}$ for K-class classification, assuming that class probability follows a prior Dirichlet distribution, the cross-entropy loss to be minimized for learning evidence $\mathbf{e}^{(i)} \in \mathbb{R}_{+}^{K}$ eventually reduces to the following form:

$$\mathcal{L}_{EDL}^{(i)}(\mathbf{y}^{(i)}, \mathbf{e}^{(i)}; \theta) = \sum_{k=1}^{K} \mathbf{y}_k^{(i)} \left(\log S^{(i)} - \log(\mathbf{e}_k^{(i)} + 1) \right) \tag{5.1}$$

where $\mathbf{y}^{(i)}$ is an one-hot K-dimensional label for sample $\mathbf{x}^{(i)}$ and $\mathbf{e}^{(i)}$ can be expressed as $\mathbf{e}^{(i)} =$

[1] In our experiments, we use four different action recognition models which are I3D [27], TSM [193], SlowFast [74], and TPN [360].



(a) AC (b) AU (c) IC (d) IU

Figure 5.4 **Examples of Probability Simplex**. We use a 3-class classification as an example and assume the first class as the correct label. A well-calibrated model should give **A**ccurate and **C**ertain (AC) predictions (Fig. 5.4a) or **I**naccurate and **U**ncertain (IU) predictions (Fig. 5.4d), while the AU (Fig. 5.4b) and IC (Fig. 5.4c) cases need to be reduced.

$g\left(f(\mathbf{x}^{(i)}; \theta)\right)$. Here, f is the output of a DNN parameterized by θ and g is the evidence function to keep evidence $\mathbf{e}_k$ non-negative. S is the total strength of a Dirichlet distribution $\mathrm{Dir}(\mathbf{p}|\boldsymbol{\alpha})$, which is parameterized by $\boldsymbol{\alpha} \in \mathbb{R}^K$, and S is defined as $S = \sum_{k=1}^{K} \boldsymbol{\alpha}_k$. Based on DST and SL theory, the $\boldsymbol{\alpha}_k$ is linked to the learned evidence $\mathbf{e}_k$ by the equality $\boldsymbol{\alpha}_k = \mathbf{e}_k + 1$. In the inference, the predicted probability of the k-th class is $\hat{\mathbf{p}}_k = \boldsymbol{\alpha}_k / S$ and the predictive uncertainty u can be deterministically given as $u = K/S$. More detailed derivations can be found in our supplementary material.

EDL for Action Recognition. In this paper, we propose to formulate action recognition from the EDL perspective. In the training phase, by applying the EDL objective in (5.1) for the action dataset, we are essentially trying to collect evidence of each action category for an action video. In the testing phase, since the action probability $\mathbf{p} \in \mathbb{R}^K$ is assumed to follow a Dirichlet, i.e., $\mathbf{p} \sim \mathrm{Dir}(\mathbf{p}|\boldsymbol{\alpha})$, the categorical probability and uncertainty of a human action can be jointly expressed by a $(K-1)$-simplex (see the triangular heat map in Fig. 5.3). The EDL uncertainty enables the action recognition model to “know the unknown".

However, due to the deterministic nature of EDL, the potential over-fitting issue would hamper the generalization capability for achieving good OSAR performance. Besides, the static bias problem in video data is still not addressed by EDL. To this end, we propose a model calibration method and a representation debiasing module below.

5.3.2 Evidential Uncertainty Calibration

Though the evidential uncertainty from EDL can be directly learned without sampling, the uncertainty may not be well calibrated to handle the unknown samples in the OSAR setting. As pointed out in existing model calibration literature [231, 155], a well-calibrated model should be confident in its predictions when being accurate, and be uncertain about inaccurate ones. Besides, existing DNN models have empirically demonstrated that miscalibration is linked to the over-fitting of the negative log-likelihood (NLL) [99, 232]. Since the EDL objective in (5.1) is equivalent to minimizing the NLL [283], the trained model is likely to be over-fitted with poor generalization for OSAR tasks. To address this issue, we propose to calibrate the EDL model by considering the relationship between accuracy and uncertainty.

To this end, we follow the same goal as [231, 155] to maximize the *Accuracy versus Uncertainty* (AvU) utility function for calibrating the uncertainty:

$$\text{AvU} = \frac{n_{AC} + n_{IU}}{n_{AC} + n_{AU} + n_{IC} + n_{IU}} \tag{5.2}$$

where the n_{AC}, n_{AU}, n_{IC}, and n_{IU} represent the numbers of samples in four predicted cases, i.e., (1) **A**ccurate and **C**ertain (**AC**), (2) **A**ccurate and **U**ncertain (**AU**), (3) **I**naccurate and **C**ertain (**IC**), and (4) **I**naccurate and **U**ncertain (**IU**). A well-calibrated model could achieve high AvU utility so that the predictive uncertainty can be consistent with accuracy. Fig. 5.4 shows a toy example of the four possible EDL outputs. To calibrate the predictive uncertainty, the EDL model is encouraged to learn a skewed and sharp Dirichlet simplex for accurate prediction (Fig. 5.4a), and to provide an unskewed and flat Dirichlet simplex for incorrect prediction (Fig. 5.4d). To this end, we propose to regularize EDL training by minimizing the expectations of AU and IC cases (Fig. 5.4b and Fig. 5.4c) such that the other two cases can be encouraged. Therefore, if a video is assigned with high EDL uncertainty, it is more likely to be incorrect so that an unknown action is identified.

In particular, we propose an *Evidential Uncertainty Calibration* (EUC) method to minimize the following sum of AU and IC cases by considering the logarithm constraint between the confidence

p_i and uncertainty u_i:

$$\begin{aligned}\mathcal{L}_{EUC} = -\lambda_t &\sum_{i\in\{\hat{y}_i=y_i\}} p_i \log(1-u_i) \\ -(1-\lambda_t) &\sum_{i\in\{\hat{y}_i\neq y_i\}} (1-p_i)\log(u_i)\end{aligned} \tag{5.3}$$

where p_i is the maximum class probability of an input sample $\mathbf{x}^{(i)}$ and u_i is the associated evidential uncertainty. The first term aims to give low uncertainty ($u_i \rightarrow 0$) when the model makes an accurate prediction ($\hat{y}_i = y_i$, $p_i \rightarrow 1$), while the second term tries to give high uncertainty ($u_i \rightarrow 1$) when the model makes inaccurate prediction ($\hat{y}_i \neq y_i$, $p_i \rightarrow 0$). Note that the annealing factor $\lambda_t \in [\lambda_0, 1]$ is defined as $\lambda_t = \lambda_0 \exp\{-(\ln \lambda_0/T)t\}$. Here, λ_0 is a small positive constant, i.e., $\lambda_0 \ll 1$, such that λ_t is monotonically increasing w.r.t. training epoch t, and T is the total number of training epochs. As the training epoch t increases to T, the factor λ_t will be exponentially increasing from λ_0 to 1.0.

The motivation behind the annealing weighting is that the dominant periods of accurate and inaccurate predictions in model training are different. In the early training stages, the inaccurate predictions are the dominant cases so that the IC loss (second term) should be more penalized, while in the late training stages, the accurate predictions are the dominant so the AU loss (first term) should be more penalized. Therefore, the annealing weighing factor λ_t dynamically balances the two terms in training.

Discussion. Our EUC method is advantageous over existing approaches [231] and AvUC [155] in following aspects. First, compared with [231], our EUC method takes the same merit of AvUC in that it is a fully differentiable regularization term. Second, compared with both [231] and AvUC, the EUC loss does not rely on the distribution shifted validation set during training which is not reasonable for the OSAR model to access the OOD samples. Therefore, our method provides better flexibility to calibrate deep learning models on large-scale datasets, such as the real-world videos of human actions addressed in this paper. Our experimental results (Table 5.3) show that the model calibration performance of the EUC method is more significant for open-set recognition than for closed-set recognition.

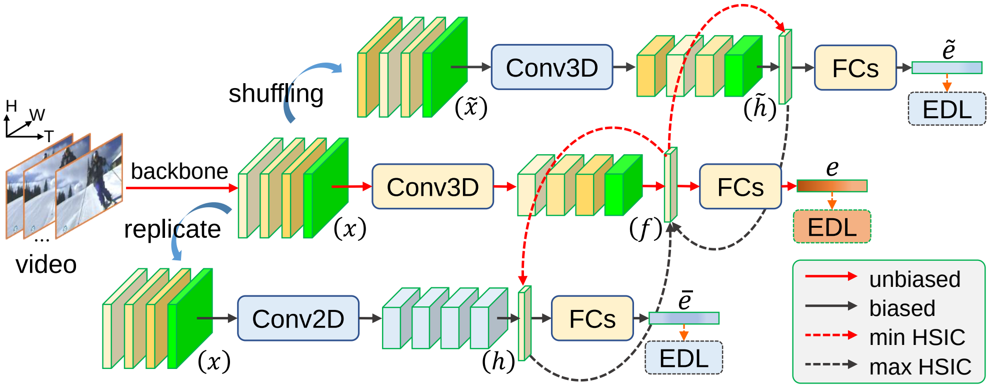


Figure 5.5 **Contrastive Evidence Debiasing (CED) Module**. The module consists of three branches with similar structures. In contrast to the middle branch, the top and bottom ones aim to learn biased evidence by temporally shuffled feature input and 2D convolution (*Conv2D*), respectively. The generated feature **f** is contrastively pushed to be independent of biased feature **h**.

5.3.3 Contrastive Evidence Debiasing

For OSAR task, static bias (see example in Fig. 5.2) in a video dataset is one of the most challenging problems that limit the generalization capability of a model in an open set setting. According to [183], static bias can be categorized into scene bias, object bias, and human bias. Existing research work [42, 183, 139, 6] has empirically shown that debiasing the model by input data or learned representation can significantly improve the action recognition performance. As pointed out in [183], it is intrinsically nothing wrong about the bias if it can be “over-fitted" by an action recognition model for achieving a “good" performance in a traditional closed-set setting. However, in an open set setting, the static bias could result in a vulnerable model that falsely recognizes an action video containing similar static features but out-of-contextual temporal dynamics.

In this paper, we propose a Contrastive Evidence Debiasing (CED) module to mitigate the static bias problem. As shown in Fig. 5.5, the CED consists of three branches. The middle branch is a commonly-used 3D convolutional structure (Conv3D) to predict unbiased evidence (**e**) while the top and bottom branches predict biased evidence ($\tilde{\mathbf{e}}$ and $\bar{\mathbf{e}}$). In particular, the top branch keeps the same network structure as the middle one but takes temporally shuffled features ($\tilde{\mathbf{x}}$) as input. The bottom branch keeps the same input feature (**x**) as the middle one but replaces the Conv3D with a 2D convolutional structure (Conv2D). Finally, with the HSIC-based min-max optimization, the feature **f** for predicting unbiased evidence is encouraged to be contrastive to the features **h** and $\tilde{\mathbf{h}}$ for predicting biased evidence.

In particular, motivated by the recent method ReBias [6], the min-max optimization is defined by using the Hilbert-Schmidt Independence Criterion (HSIC). The HSIC function measures the degree of independence between two continuous random variables. With radial basis function (RBF) kernel k_1 and k_2, $\text{HSIC}^{k_1,k_2}(\mathbf{f},\mathbf{h})=0$ if and only if $\mathbf{f} \perp\!\!\!\perp \mathbf{h}$. The detailed mathematical form of HSIC can be found in [97, 295] (or see the Section 5.6.1.3 of the supplementary material) . For the middle branch, the goal is to learn a discriminative and unbiased feature **f** by minimizing

$$\mathcal{L}(\theta_f, \phi_f) = \mathcal{L}_{EDL}(\mathbf{y}, \mathbf{e}; \theta_f, \phi_f) + \lambda \sum_{\mathbf{h}\in\Omega} \text{HSIC}(\mathbf{f}, \mathbf{h}; \theta_f), \tag{5.4}$$

where θ_f and ϕ_f are parameters of neural networks to produce unbiased feature **f** and to predict evidence **e**. **y** is the multi-class label. The second term encourages feature **f** to be independent of the biased feature **h** from the set of features generated by top branch $h_{3D}(\tilde{\mathbf{x}})$ and the bottom branch $h_{2D}(\mathbf{x})$, i.e., $\Omega=\{h_{3D}(\tilde{\mathbf{x}}), h_{2D}(\mathbf{x})\}$.

For the top and bottom branches, the goal is to learn the above two types of biased feature **h** by

$$\mathcal{L}(\theta_h, \phi_h) = \sum_{\mathbf{h}\in\Omega} \{\mathcal{L}_{EDL}(\mathbf{y}, \mathbf{e}_h; \theta_h, \phi_h) - \lambda \text{HSIC}(\mathbf{f}, \mathbf{h}; \theta_h)\} \tag{5.5}$$

where θ_h denotes the network parameters of $h_{3D}(\tilde{\mathbf{x}})$ and $h_{2D}(\mathbf{x})$ to generate biased features **h**, and the ϕ_h denotes the parameters of neural networks to predict corresponding evidence $\mathbf{e}_h \in \{\hat{\mathbf{e}}, \bar{\mathbf{e}}\}$. The first term in (5.5) aims to avoid the biased feature **h** to predict arbitrary evidence, while the second term guarantees that **h** is similar enough to **f** so that **f** has to be pushed faraway from **h** by (5.4).

The two objectives in (5.4) and (5.5) are alternatively optimized so that feature **h** is learned to be biased to guide the debiasing of feature **f**. In practice, we also implemented a joint training strategy that aims to optimize the objective of (5.4) and (5.5) jointly and we empirically found it can achieve better performance.

Discussion. Compared with recent work [42] that leverages adversarial learning to remove scene bias, our method does not rely on object bounding boxes and pseudo-scene labels as auxiliary training input. The representation bias addressed in our paper implicitly encompasses all sources of biases, not just the scene bias. Compared with ReBias [6], our CED module shares a similar idea of removing bias with bias. However, the HSIC in our CED module considers not only the bias-characterizing model (i.e., $h_{2D}(\mathbf{x})$) as in [6], but also the biased feature input by temporal shuffling. This consideration will further encourage the backbone to focus more on temporal dynamics. Besides, our CED is a plug-and-play module and can be flexibly inserted into any state-of-the-art deep learning-based action recognition model with little coding effort.

5.4 Experiments

Dataset. We evaluate the proposed DEAR method on three commonly used real-world video action datasets, including UCF-101 [296], HMDB-51 [159], and MiT-v2 [230]. All models are trained on UCF-101 training split. MiT-v2 has 305 classes and its testing split contains 30,500 video samples, which are about 20 times larger than the HMDB-51 testing set. In testing, we use the UCF-101 testing set as known samples, and the testing splits of HMDB-51 and MiT-v2 datasets as two sources of unknown. Note that there could be a few overlapping classes between UCF-101 and the other two datasets, but for standardizing the evaluation and reproducibility, we do not manually clean the data.

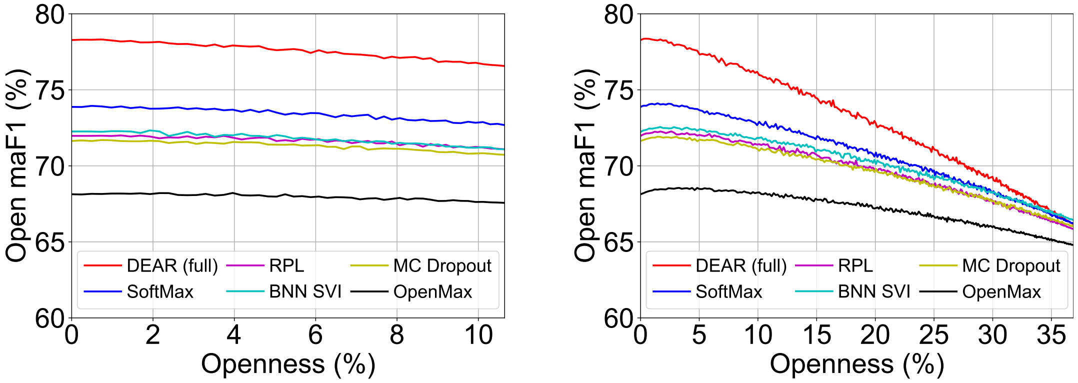
Evaluation Protocol. To evaluate the classification performance on both closed and open set settings, we separately report the Closed Set Accuracy for K-class classification and the Open Set area under ROC curve (AUC) for distinguishing known and unknown (2 classes). Furthermore, to comprehensively evaluate the $(K + 1)$-class classification performance, i.e., the unknown as the $(K + 1)$-th class, we plot the curve of macro-F1 scores by gradually increasing the openness similar to existing literature [291, 372, 302]. For each openness point, i new classes are randomly selected from HMDB-51 (where $i \leq 51$) or MiT-v2 (where $i \leq 305$) test set and we compute the macro-F1 score for each of 10 randomized selections. Since there is no existing quantitative metric to summarize the performance of the F1 curve, in this paper we propose an **Open maF1** score:

$$\text{Open maF1} = \frac{\sum_i \omega_O^{(i)} \cdot F_1^{(i)}}{\sum_i \omega_O^{(i)}} \tag{5.6}$$

Table 5.1 **Comparison with state-of-the-art methods.** Models are trained on the closed set UCF-101 [296] and tested on two different open sets where the samples of unknown class are from HMDB-51 [159] and MiT-v2 [230], respectively. For Open maF1 scores, both the mean and standard deviation of 10 random trials of unknown class selection are reported. Closed set accuracy (CS-Acc) is for reference only.

Models	OSAR	HMDB-51 [159]		MiT-v2 [230]		CS-Acc (%)
		Open maF1 (%)	OS-AUC (%)	Open maF1 (%)	OS-AUC (%)	
I3D [27]	OpenMax [16]	67.85 ± 0.12	74.34	66.22 ± 0.16	77.76	56.60
	MC Dropout	71.13 ± 0.15	75.07	68.11 ± 0.20	79.14	94.11
	BNN SVI [153]	71.57 ± 0.17	74.66	68.65 ± 0.21	79.50	93.89
	SoftMax	73.19 ± 0.17	75.68	68.84 ± 0.23	79.94	94.11
	RPL [34]	71.48 ± 0.15	75.20	68.11 ± 0.20	79.16	94.26
	DEAR (ours)	**77.24** ± 0.18	**77.08**	**69.98** ± 0.23	**81.54**	93.89
TSM [193]	OpenMax [16]	74.17 ± 0.17	77.07	**71.81** ± 0.20	83.05	65.48
	MC Dropout	71.52 ± 0.18	73.85	65.32 ± 0.25	78.35	95.06
	BNN SVI [153]	69.11 ± 0.16	73.42	64.28 ± 0.23	77.39	94.71
	SoftMax	78.27 ± 0.20	77.99	71.68 ± 0.27	82.38	95.03
	RPL [34]	69.34 ± 0.17	73.62	63.92 ± 0.25	77.28	95.59
	DEAR (ours)	**84.69** ± 0.20	**78.65**	70.15 ± 0.30	**83.92**	94.48
SlowFast [74]	OpenMax [16]	73.57 ± 0.10	78.76	72.48 ± 0.12	80.62	62.09
	MC Dropout	70.55 ± 0.14	75.41	67.53 ± 0.17	78.49	96.75
	BNN SVI [153]	69.19 ± 0.13	74.78	65.22 ± 0.21	77.39	96.43
	SoftMax	78.04 ± 0.16	79.16	74.42 ± 0.22	82.88	96.70
	RPL [34]	68.32 ± 0.13	74.23	66.33 ± 0.17	77.42	96.93
	DEAR (ours)	**85.48** ± 0.19	**82.94**	**77.28** ± 0.26	**86.99**	96.48
TPN [360]	OpenMax [16]	65.27 ± 0.09	74.12	64.80 ± 0.10	76.26	53.24
	MC Dropout	68.45 ± 0.12	74.13	65.77 ± 0.17	77.76	95.43
	BNN SVI [153]	63.81 ± 0.11	72.68	61.40 ± 0.15	75.32	94.61
	SoftMax	76.23 ± 0.14	77.97	70.82 ± 0.21	81.35	95.51
	RPL [34]	70.31 ± 0.13	75.32	66.21 ± 0.21	78.21	95.48
	DEAR (ours)	**81.79** ± 0.15	**79.23**	**71.18** ± 0.23	**81.80**	96.30

where $\omega_O^{(i)}$ denotes the openness when i new classes are introduced and it is defined as $\omega_O^{(i)} = 1-\sqrt{2K/(2K+i)}$ according to [278]. $F_1^{(i)}$ is the macro-F1 score by considering the samples from all new classes as unknown. The basic idea of weighting F_1 by ω_O is that the result is essentially the normalized area under the curve of macro-F1 vs. openness. The Open maF1 quantitatively evaluates the performance of $(K+1)$-class classification in an open set setting.



(a) HMDB-51 as Unknown

(b) MiT-v2 as Unknown

Figure 5.6 **Open macro-F1 scores against varying Openness**. The maximum openness is determined by the number of unknown classes, i.e., in $\omega_O^{(i)}$, $i=51$ for HMDB-51 and $i=305$ for MiT-v2.

Implementation Details. Our method is implemented with the PyTorch codebase MMAction2 [45]. The adopted action models are experimented with ResNet-50 backbone pre-trained on Kinetics-400 [27] dataset and fine-tuned on UCF-101 training set. Our proposed EDL loss $\mathcal{L}_{EDL}$ is used to replace the original cross-entropy loss, and our proposed CED module is inserted into the layer before the classification heads of recognition models. During training, we use a base learning rate of 0.001 and it is step-wisely decayed for every 20 epochs with a total of 50 epochs. We set the batch size as 8 during training. The rest of the hyperparameters are kept the same as the default configuration provided by MMAction2. During inference, our CED module is removed. Other implementation details are provided in the supplementary material.

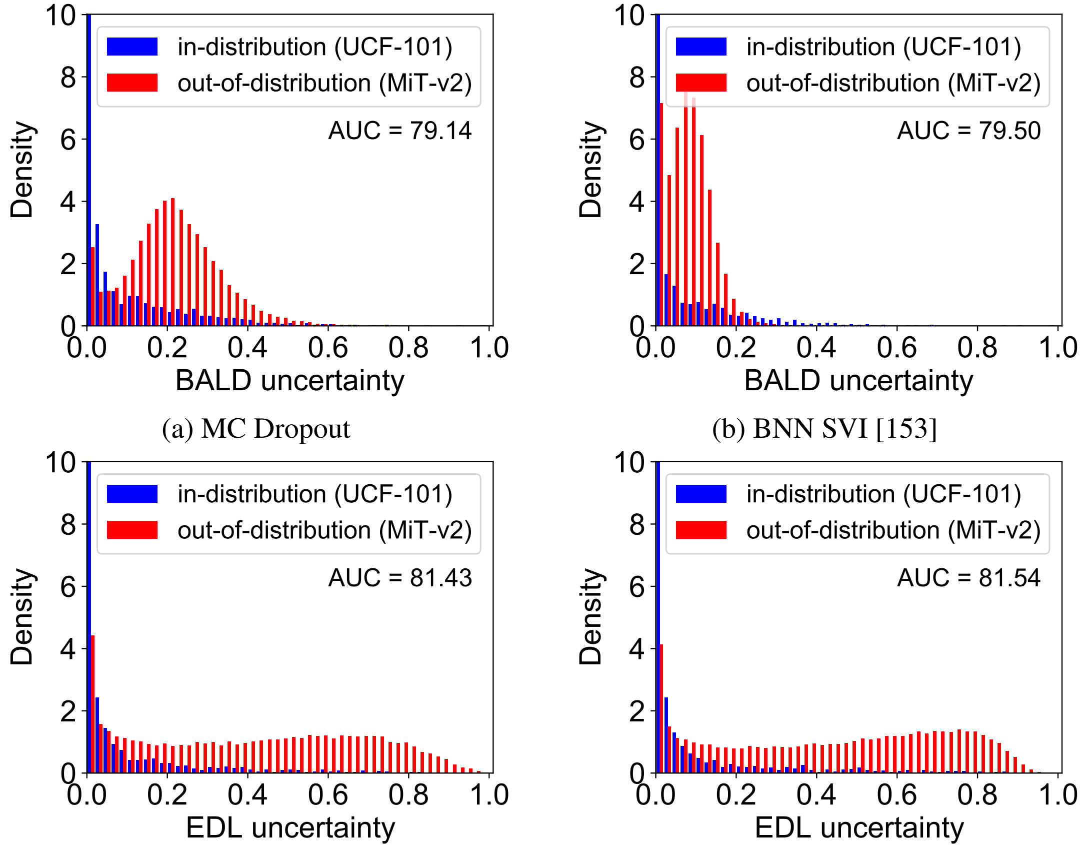
5.4.1 Comparison with State-of-the-art

The proposed DEAR method is compared with baselines as shown in the second column of Table 5.1. The open set performances are also summarized in Fig. 5.1. For these baselines, SoftMax, OpenMax, and MC Dropout share the same trained model since they are only different in the testing phase. For the MC Dropout and BNN SVI which incorporate stochastic sampling in testing, we set the 10 forward passes through the model and adopt the BALD [114] method to quantify the model uncertainty as suggested by [153]. Following [302], the threshold of the scoring function is determined by ensuring 95% training data to be recognized as known.

Open Set Action Recognition. In Table 5.1, we report the results of both closed-set and open set performance. It shows that with different action recognition models, our method consistently and significantly outperforms baselines on Open maF1 score for $(K{+}1)$-class classification and Open Set AUC score for rejecting the unknowns, while only sacrificing less than 1% performance decrease on Closed Set Accuracy. When equipped with the SlowFast model, our method could improve the MC Dropout method by almost 8% of open set AUC and 15% of Open maF1 score. OpenMax and RPL are the recent state-of-the-art OSR methods, however, we find that their performances are far behind our DEAR method on the OSAR task. Note that the closed set accuracy of OpenMax is dramatically lower than other baselines, this is because OpenMax directly modifies the activation layer before softmax and appends the unknown class as output, which could destroy the accurate predictions of known samples. Besides, we also note that with TSM model, the Open maF1 score of DEAR method is slightly inferior to OpenMax on the MiT-v2 dataset. This indicates that for large-scale unknown testing data such as MiT-v2, the 2D convolution-based TSM is not a good choice for the DEAR method as compared to those 3D convolution-based architectures such as I3D, SlowFast, and TPN.

Based on I3D model, as depicted in Fig. 5.6, we plot the average Open maF1 scores against varying openness by incrementally introducing HMDB-51 and MiT-v2 testing sets as unknown. It clearly shows that the proposed DEAR method achieves the best performance. Note that for the large-scale MiT-v2 dataset, as the openness increases, the performances of different methods converge to be close to each other. This is because the macro-F1 is sensitive to class imbalance and it will be gradually dominated by the increasing unknown classes from a total of 305 categories in MiT-v2. Nevertheless, our method DEAR still keeps better than all other baselines.

Out-of-distribution Detection. This task aims to distinguish between the in-distribution samples (known) and out-of-distribution (OOD) samples (unknown). Similar to the baseline MC Dropout and BNN SVI [153], which use uncertainty as a scoring function to identify the unknown, the OOD detection performance can be evaluated by showing the Open Set AUC in Table 5.1 and the histogram statistics in Fig. 5.7. The AUC numbers and figures clearly show that our DEAR method with EDL uncertainty can better detect the OOD samples. Compared with the vanilla



(a) MC Dropout

(b) BNN SVI [153]

(c) DEAR (vanilla)

(d) DEAR (full)

Figure 5.7 **Out-of-distribution Detection by Uncertainty.** The DEAR (vanilla) is the variant of DEAR (full) that only $\mathcal{L}_{EDL}$ is used for model training. We use MiT-v2 as unknown and I3D as the recognition model. Uncertainty values are normalized to [0,1] within each distribution.

DEAR which only uses $\mathcal{L}_{EDL}$ for model training, the estimated uncertainties of OOD samples skew closer to 1.0.

5.4.2 Ablation Study

Contribution of Each Component. In Table 5.2, it shows the OSAR performance of each DEAR variant. The experiments are conducted with the TPN model and evaluated using the HMDB-51 testing set as unknown. The results demonstrate that all the proposed components could contribute to the OSAR performance gain. In particular, the $h_{2D}(\mathbf{x})$ of our CED module contributes the most. Besides, the joint training of CED module shows slightly better than the alternative training. Therefore, by default joint training is adopted throughout other experiments.

Model Calibration. Though the proposed EUC module can improve the performance on

Table 5.2 **Ablation studies.** Based on TPN [360] model, HMDB-51 [159] is used as the unknown. The best results are shown in bold.

$\mathcal{L}_{EUC}$	CED	Joint Train	Open maF1 (%)	OS-AUC (%)
✗	✗	✓	74.95 ± 0.18	77.12
✓	✗	✓	75.88 ± 0.16	77.49
✓	✓	✗	81.18 ± 0.15	79.02
✓	✓	✓	**81.79 ± 0.15**	**79.23**

Table 5.3 **Expected Calibration Error (ECE) results.** Small ECE indicates the model is better calibrated. The numbers in brackets indicate the number of classes involved in the evaluation.

Model variants	Open Set (K+1)	Open Set (2)	Closed Set (K)
DEAR (w/o $\mathcal{L}_{EUC}$)	0.284	0.256	0.030
DEAR (full)	**0.268**	**0.239**	**0.029**

Table 5.4 **Accuracy (%) on Biased and Unbiased dataset.**

Methods	Biased (Kinetics)		Unbiased (Mimetics)	
	top-1	top-5	top-1	top-5
DEAR (w/o CED)	91.18	99.30	26.56	69.53
DEAR (full)	**91.18**	**99.54**	**34.38**	**75.00**

OSAR tasks (as shown in Table 5.2), we further dig into the question of if the performance gain of EUC results from better calibrating a classification model. To this end, we adopt the widely used Expected Calibration Error (ECE) [99] to evaluate the model calibration performance of our full method DEAR (full) and its variant without EUC loss $\mathcal{L}_{EUC}$. Quantitative results are reported in Table 5.3. It shows that $\mathcal{L}_{EUC}$ can reduce the ECE values with both open-set and closed-set recognition settings. In particular, the calibration capability is more significant in an open set setting than in a closed set setting. This validates our claim that the proposed $\mathcal{L}_{EUC}$ could calibrate an OSAR model.

Representation Debiasing. To further validate if the performance gain of our CED module is rooted in the representation debiasing, we use Kinetics [27] as a biased dataset and Mimetics [339] as an unbiased dataset. Similar to [6], we select 10 human action categories from Kinetics for training and biased testing, and select the same categories from Mimetics for unbiased testing.

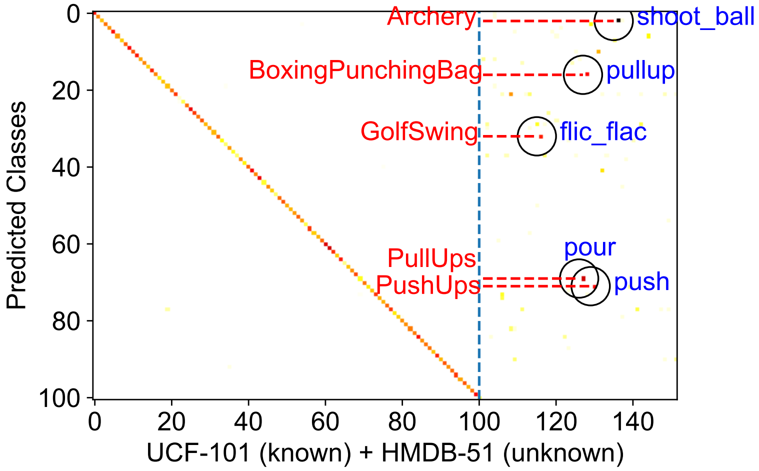


Figure 5.8 **Confusion Matrix for Known and Unknown.** The x-axis shows the ground truth classes of both UCF-101 (known) and HMD-51 (unknown), and the y-axis represents the predicted classes defined by UCF-101. This figure highlights the top 5 unknown classes (blue text) that are misclassified as the known (red text).

Without the pre-trained model from the Kinetics dataset, we apply our DEAR method with and without CED on the TSM model. The top-1 and top-5 accuracy results are reported in Table 5.4. It shows that models trained on biased datasets (Kinetics) are vulnerable to unbiased datasets (Mimetics). However, when equipped with the proposed CED module, the performance on the unbiased dataset can be significantly improved while performance on the biased dataset still keeps minor changes.

What Types of Unknown are Mis-classified? As shown in Fig. 5.8, the confusion matrix is visualized by considering both the known classes from UCF-101 and unknown classes from HMDB-51 datasets. It shows that in spite of high closed set accuracy (the diagonal line), the actions from unknown classes could be easily classified as known categories. For example, *shoot ball* is the top-1 mis-classified unknown class in HMDB-51, which is the most frequently misclassified as the known class *Archery* in UCF-101. It is convincing that the misclassification is caused by their similar background scene, i.e., a large area of grassland, which is static bias as addressed in this paper.

5.5 Conclusion

In this paper, we proposed a Deep Evidential Action Recognition (DEAR) method for the open set action recognition (OSAR) problem. OSAR is more challenging than image OSR problems due to the uncertain nature of temporal action dynamics and the static bias of background scenes. To this end, we conduct Evidential Deep Learning (EDL) to learn a discriminative action classifier with quantified predictive uncertainty, where the uncertainty is used to distinguish between the known and unknown samples. As novel extensions of EDL, an Evidential Uncertainty Calibration (EUC) method and a contrastive evidential debiasing (CED) module are proposed to address the unique challenges in OSAR. Extensive experimental results demonstrate that our DEAR method works for most existing action recognition models in an open set setting.

5.6 Supplementary Material

In this document, additional materials are provided to supplement our main paper. In section 5.6.1, the preliminary knowledge about the evidential deep learning and model calibration are described in detail, which are helpful to understand the methodology of our main paper. In section 5.6.2, additional implementation details are provided, which are useful to reproduce our proposed method.

5.6.1 Detailed Methodology

5.6.1.1 Preliminaries of Evidential Deep Learning

Existing video action recognition models typically use softmax on top of deep neural networks (DNN) for classification. However, the softmax function is heavily limited in the following aspects. First, the predicted categorical probabilities have been squashed by the denominator of softmax. This is known to result in an over-confident prediction for the unknown data, which is even more detrimental to open set recognition problem than the closed set recognition. Second, the softmax output is essentially a point estimate of the multinomial distribution over the categorical probabilities so that softmax cannot capture the uncertainty of categorical probabilities, i.e., second-order uncertainty.

To overcome these limitations, recent evidential deep learning (EDL) [283] is developed from the evidence framework of Dempster-Shafer Theory (DST) [284] and the subjective logic (SL) [130]. For a K-class classification problem, the EDL treats the input $\mathbf{x}$ as a proposition and regards the classification task as to give a multinomial subjective opinion in a K-dimensional domain $\{1, \ldots, K\}$. The subjective opinion is expressed as a triplet $\omega = (\mathbf{b}, u, \mathbf{a})$, where $\mathbf{b} = \{b_1, \ldots, b_K\}$ is the belief mass, u represents the uncertainty, and $\mathbf{a} = \{a_1, \ldots, a_K\}$ is the base rate distribution. For any $k \in [1, 2, \ldots, K]$, the probability mass of a multinomial opinion is defined as

$$p_k = b_k + a_k u, \quad \forall y \in \mathbb{Y} \tag{5.7}$$

To enable the probability meaning of p_k, i.e., $\sum_k p_k = 1$, the base rate a_k is typically set to $1/K$ and the subjective opinion is constrained by

$$u + \sum_{k=1}^{K} b_k = 1 \tag{5.8}$$

Besides, for a K-class setting, the probability mass $\mathbf{p} = [p_1, p_2, \ldots, p_K]$ is assumed to follow a Dirichlet distribution parameterised by a K-dimensional Dirichlet strength vector $\boldsymbol{\alpha} = \{\alpha_1, \ldots, \alpha_K\}$:

$$\mathrm{Dir}(\mathbf{p}|\boldsymbol{\alpha}) = \begin{cases} \dfrac{1}{B(\boldsymbol{\alpha})} \displaystyle\prod_{k=1}^{K} p_k^{\alpha_k - 1}, & \text{for } \mathbf{p} \in \mathcal{S}_K, \\ 0, & \text{otherwise}, \end{cases} \tag{5.9}$$

where $B(\boldsymbol{\alpha})$ is a K-dimensional Beta function, $\mathcal{S}_K$ is a K-dimensional unit simplex. The total strength of the Dirichlet is defined as $S = \sum_{k=1}^{K} \alpha_k$. Note that for the special case when $K = 2$, the Dirichlet distribution reduces to a Beta distribution and a binomial subjective opinion will be formulated in this case.

According to the evidence theory, the term *evidence* is introduced to describe the amount of supporting observations for classifying the data $\mathbf{x}$ into a class. Let $\mathbf{e} = \{e_1, \ldots, e_K\}$ be the evidence for K classes. Each entry $e_k \geq 0$ and the Dirichlet strength $\boldsymbol{\alpha}$ are linked according to the evidence theory by the following identity:

$$\boldsymbol{\alpha} = \mathbf{e} + \mathbf{a}W \tag{5.10}$$

where W is the weight of uncertain evidence. With the Dirichlet assumption, the expectation of the multinomial probability $\mathbf{p}$ is given by

$$\mathbb{E}(p_k) = \frac{\alpha_k}{\sum_{k=1}^{K} \alpha_k} = \frac{e_k + a_k W}{W + \sum_{k=1}^{K} e_k} \tag{5.11}$$

With loss of generality, the weight W is set to K and considering the assumption of the subjective opinion constraint in Eq. (5.8) that $a_k = 1/K$, we have the Dirichlet strength $\alpha_k = e_k + 1$ according to Eq. (5.10). In this way, the Dirichlet evidence can be mapped to the subjective opinion by setting the following equality's:

$$b_k = \frac{e_k}{S} \quad \text{and} \quad u = \frac{K}{S} \tag{5.12}$$

Therefore, we can see that if the evidence e_k for the k-th class is predicted, the corresponding expected class probability in Eq. (5.7) (or Eq. (5.11)) can be rewritten as $p_k = \alpha_k/S$. From Eq. (5.12), it is clear that the predictive uncertainty u can be determined after α_k is obtained.

Inspired by this idea, the EDL leverages deep neural networks (DNN) to directly predict the evidence $\mathbf{e}$ from the given data $\mathbf{x}$ for a K-class classification problem. In particular, the output of the DNN is activated by a non-negative evidence function. Considering the Dirichlet prior, the DNN is trained by minimizing the negative log-likelihood:

$$\begin{aligned}\mathcal{L}_{EDL}^{(i)}(\mathbf{y}, \mathbf{e}; \theta) &= -\log\left(\int \prod_{k=1}^{K} p_{ik}^{y_{ik}} \frac{1}{B(\boldsymbol{\alpha}_i)} \prod_{k=1}^{K} p_{ik}^{\alpha_{ik}-1} d\mathbf{p}_i\right) \\ &= \sum_{k=1}^{K} y_{ik} \left(\log(S_i) - \log(e_{ik} + 1)\right)\end{aligned} \tag{5.13}$$

where $\mathbf{y}_i = \{y_{i1}, \ldots, y_{iK}\}$ is an one-hot K-dimensional label for sample i and $\mathbf{e}_i$ can be expressed as $\mathbf{e}_i = g\left(f(\mathbf{x}_i; \theta)\right)$. Here, f is the DNN parameterized by θ and g is the evidence function such as exp, softplus, or ReLU. Note that in [283], there are two other forms of EDL loss function. In our main paper, we found the Eq. (5.13) achieves better training empirical performance.

5.6.1.2 EDL for Open Set Action Recognition

To implement the EDL method on video action recognition tasks, we removed the Kullback–Leibler (KL) divergence regularizer term defined in [283], because the digamma function involved in the KL divergence is not numerically stable for large-scale video data. Instead, to compensate for the over-fitting risk, we propose the Evidential Uncertainty Calibration (EUC) as a new regularization. Together with the Contrastive Evidence Debiasing module, the complete training objective of our DEAR method can be expressed as

$$\mathcal{L} = \sum_i \mathcal{L}_{EDL}^{(i)} + w_1 \mathcal{L}_{EUC} + w_2 \mathcal{L}_{CED} \tag{5.14}$$

where $\mathcal{L}_{EUC}$ is defined in Eq. (5.3) in our main paper, and $\mathcal{L}_{CED}$ is the sum of (or one of for alternative training) $\mathcal{L}(\theta_f, \phi_f)$ and $\mathcal{L}(\theta_h, \phi_h)$ defined in Eq. (5.4) and Eq. (5.5) respectively in our main paper. The hyperparameters w_1 and w_2 are set to 1.0 and 0.1, respectively.

During the training process, the DEAR model aims to accurately construct the Dirichlet parameters $\boldsymbol{\alpha}$ by collecting the *evidence* from human action video training set. In the inference phase, the probability of each action class is predicted as $\hat{p}_k = \alpha_k / S$ while the predictive uncertainty is simultaneously computed as $u = K/S$. If an input action video is assigned with high uncertainty, which means a vacuity of evidence to support for closed-set classification, the action is likely to be unknown from the open testing set.

Compared with existing DNN-based uncertainty estimation method such as Bayesian neural networks (BNN) or deep Gaussian process (DGP), the advantage of EDL is that the predictive uncertainty is deterministically learned without inexact posterior approximation and computationally expensive sampling. These merits enable the EDL method to be efficient for training recognition models from large-scale vision data such as the human action videos.

5.6.1.3 Hilbert-Schmidt Independence Criterion

Hilbert-Schmidt Independence Criterion (HSIC) is a commonly-used dependency measurement of two high-dimensional variables. In practice, we used the unbiased HSIC estimator in [295] with m samples:

$$\text{HSIC}^{k,l}(U,V) = \frac{1}{m(m-3)}\left[\text{tr}(\tilde{U}\tilde{V}^T) + \frac{\mathbf{1}^T\tilde{U}\mathbf{1}\mathbf{1}^T\tilde{V}\mathbf{1}}{(m-1)(m-2)} - \frac{2}{m-2}\mathbf{1}^T\tilde{U}\tilde{V}^T\mathbf{1}\right], \tag{5.15}$$

where $\tilde{U}$ is the kernelized matrix of U with RBF kernel k by $\tilde{U}_{ij} = (1-\delta_{ij})k(u_i, u_j)$, $\{u_i\} \sim U$ and the $(1-\delta_{ij})$ sets the diagonal of $\tilde{U}$ to zeros. $\tilde{V}$ is defined similarly with kernel l, and $\mathbf{1}$ is an all-one vector. The HSIC value is equal to zero if and only if the two variables are independent.

5.6.1.4 Evaluation of Model Calibration

In our main paper, we used the expected calibration error (ECE) to quantitatively evaluate the model calibration performance of our proposed EUC method. According to [235, 99], the basic idea of model calibration is that, if the confidence estimation $\hat{p}$ (probability of correctness) is well calibrated, we hope $\hat{p}$ represent the true probability of the case when the predicted label $\hat{y}$ is correct. Formally, this can be expressed as

$$\mathbb{P}(\hat{y} = y | \hat{p} = p) = p \tag{5.16}$$

Since perfect calibration is infeasible due to the finite sample space, a practical way is to group all predicted confidence $\hat{p}$ into M bins in the range of [0,1] such that the width of each bin is $1/M$. Therefore, for the m-th bin, the accuracy can be estimated by

$$\mathrm{acc}(B_m) = \frac{1}{|B_m|} \sum_{i \in B_m} \mathbb{I}(\hat{y}_i = y_i) \tag{5.17}$$

where B_m is the set of indices of prediction $\hat{p}$ when it falls into the m-th bin. $\hat{y}_i$ and y_i are predicted and ground truth labels. Besides, the average confidence for the m-th bin can be expressed as

$$\mathrm{conf}(B_m) = \frac{1}{|B_m|} \sum_{i \in B_m} \hat{p}_i \tag{5.18}$$

To evaluate the mis-calibration error, the ECE is defined as the expectation of the gap between the accuracy and confidence in M bins for all N samples:

$$\mathrm{ECE} = \sum_{m=1}^{M} \frac{|B_m|}{N} |\mathrm{acc}(B_m) - \mathrm{conf}(B_m)| \tag{5.19}$$

A perfect calibrated model means that ECE=0 and higher ECE value indicates that the model is less calibrated.

5.6.2 Implementation Details

Network Architecture. As presented in our main paper, the proposed DEAR method as well as all other baselines are implemented on top of the four recent video action recognition models, i.e., I3D, TSM, SlowFast, and TPN. For simplicity, these models use ResNet-50 as the backbone architecture and the network weights are initialized with the pre-trained model from the Kinetics-400 benchmark. To avoid the impact of the validation experiments on the Kinetics and Mimetics datasets, the pre-trained model is not used and we train the model from scratch using the same hyperparameters.

Specifically, for the **I3D** model, it is straightforward to implement our method by replacing the cross-entropy loss with the proposed EUC regularized EDL loss, and inserting the proposed CED module before the recognition head (fully-connected layers). For the **TSM** model, since the architecture of TSM is based on 2D convolution where the output feature embedding is with the size (B, MC, H, W), we recover the number of video segments M as the temporal dimension such that the 5-dimensional tensor with size (B, C, M, H, W) could be compatible with our proposed CED module for contrastive debiasing. For the **SlowFast** model, our CED module is inserted after the *slow* pathway because the feature embedding of slow pathway is more likely to be biased since it captures the static cues of video content. For the **TPN** model, we used the ResNet-50-like SlowOnly model as the recognition backbone and the auxiliary cross-entropy loss in the TPN head is kept unchanged.

Training and Inference. In the training phase, we choose the exp function as the evidence function because we empirically found exp is numerically more stable when using the proposed EDL loss $\mathcal{L}_{EDL}$. We set the hyperparameter λ_0 to 0.01 in EUC loss $\mathcal{L}_{EUC}$ and set λ to 1.0 in the two CED losses. The weight of $\mathcal{L}_{EUC}$ is set to 1.0 and the weight of the sum of the two CED losses is empirically set to 0.1. In practice, we found the model performance is robust to these hyperparameters. We used mini-batch SGD with nesterov strategy to train all the 3D convolution models. For all models, weight decay is set to 0.0001 and momentum factor is set to 0.9 by default. Our experiments are supported by two GeoForce RTX 3090 and two Tesla A100 GPUs. Since no additional parameters are introduced during inference, the inference speed of existing action recognition models is not affected.

Dataset Information. For the UCF-101 and HMDB-51 datasets, we used the *split1* for all experiments. For the MiT-v2 dataset, we only use the testing set for evaluation. To validate the proposed CED module, we refer to [6] and select 10 action categories which are included in both Kinetics and Mimetics dataset. These categories are *canoeing or kayaking*, *climbing a rope*, *driving car*, *golf driving*, *opening bottle*, *playing piano*, *playing volleyball*, *shooting goal (soccer)*, *surfing water*, and *writing*. The recognition model is trained from scratch on the 10 categories of Kinetics training set, and tested on these categories of both Kinetics and Mimetics testing set.

CHAPTER 6

OPEN-SET TEMPORAL ACTION LOCALIZATION

6.1 Introduction

Temporal Action Localization (TAL) aims to temporally localize and recognize human actions in an untrimmed video. With the success of deep learning in video understanding [27, 74, 148, 316, 35] and object detection [268, 24, 314], TAL has experienced remarkable advance in recent years [30, 388, 356, 192]. However, these works are rooted in the closed-set assumption that testing videos are assumed to contain only the pre-defined action categories, which is impractical in an open world where unknown human actions are inevitable to appear. In this paper, we for the first time step forward the Open Set Temporal Action Localization (OSTAL) problem.

OSTAL aims to not only temporally localize and recognize the known actions but also reject the localized unknown actions. As shown in Fig. 6.1, given an untrimmed video (the top row) from open world, traditional TAL (the middle row) could falsely accept the unknown action clip *HammerThrow* as one of the known actions such as the *LongJump*, while the proposed OSTAL (the bottom row) could correctly reject the clip as the *Unknown*. Besides, both tasks need to differentiate between foreground actions and the *Backgrounds* which are purely background frames.

The proposed OSTAL task is fundamentally more challenging than both the TAL and the closely relevant open set recognition (OSR) [278] problems. On one hand, the recognition and localization of known actions become harder due to the mixture of background frames and unknown foreground actions. Existing TAL methods typically assign the mixture with a non-informative *Background* label or a wrong action label, which are unable to differentiate between them. On the other hand, different from the OSR problem, rejecting an unknown action is conditioned on positively localizing a foreground action so that the localization quality is critical to the OSTAL.

To tackle these challenges, we propose a general framework OpenTAL by decoupling the overall

This chapter is adapted from the following publication:
"Wentao Bao, Qi Yu, and Yu Kong. OpenTAL: Towards open set temporal action localization. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Oral, 2022."

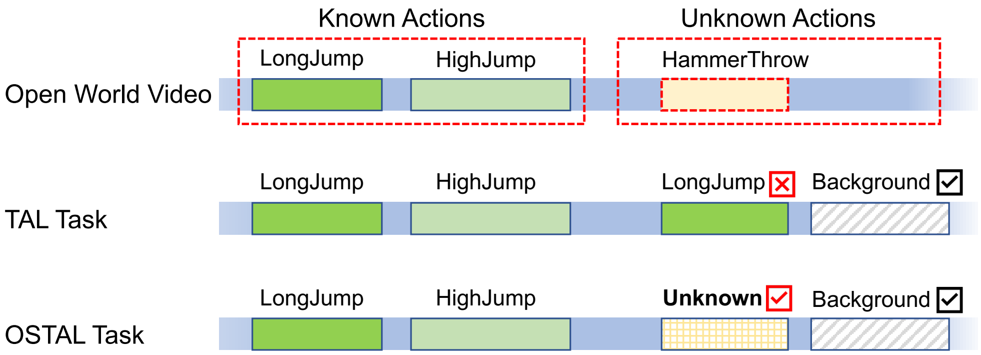


Figure 6.1 **OSTAL and TAL Tasks**. The OSTAL task is different from the TAL in that, there exist unknown actions in untrimmed open-world videos and the OSTAL models need to reject the positively localized action (e.g., *HammerThrow*) as the *Unknown*, rather than falsely assign it a known label such as the *LongJump*.

OSTAL objective into three interconnected components: uncertainty-aware action classification, actionness prediction, and temporal location regression. In essence, the foreground actions are distinguished from the background by the actionness prediction and localized by the temporal localization, while the known and unknown foreground actions are discriminated by the learned evidential uncertainty from the classification module. To achieve these goals, we propose three novel technical approaches as follows.

First, action classification is developed to recognize known actions and quantify the classification uncertainty by recent evidential deep learning (EDL) [283, 401, 4, 316]. To enable this module to learn from important samples, we propose an importance-balanced EDL method by leveraging the magnitude of EDL gradient and evidential features. Second, actionness prediction is to differentiate between foreground actions (positives) and background frames (negatives). In the open set setting, due to the mixture of unknown foreground actions (unlabeled) and background frames, learning from the labeled known actions and the mixture intrinsically reduces to a positive-unlabeled (PU) learning problem [14]. To this end, we propose a PU learning method by selecting the top negative samples from the mixture as the true negatives. Third, the temporal localization module is trained to not only localize the known actions but also calibrate the classification uncertainty. We propose an IoU-aware uncertainty calibration (IoUC) method by using the temporal Intersection-over-Union (IoU) as the localization quality to calibrate the uncertainty.

Based on the existing TAL datasets THUMOS14 [128] and ActivityNet1.3 [22], we set up a new benchmark to evaluate baselines and the proposed OpenTAL method for the OSTAL task, where the *Open Set Detection Rate* is introduced to comprehensively evaluate the OSTAL performance. Experimental results show significant superiority of our method and indicate large room for improvement in this direction. Our main contribution is threefold:

- To the best of our knowledge, this work is the first attempt on open set temporal action localization (OSTAL), which is fundamentally more challenging but highly valuable in open-world settings.
- We propose a general OpenTAL framework to address the unique challenges of OSTAL as compared with existing TAL and OSR problems. It is flexible to enable existing TAL models for open-set scenarios.
- The proposed importance-balanced EDL, PU learning, and IoUC methods are found effective for OSTAL tasks based on the OpenTAL framework.

6.2 Related Work

Temporal Action Localization The goal of Temporal Action Localization (TAL) is to recognize and temporally localize all the action instances in an untrimmed video. Existing TAL methods fall into two dominant paradigms: one-stage and two-stage approaches. The two-stage approaches [356, 388, 297, 354, 210] generate class-agnostic temporal proposals[7, 109, 195, 194] at first and then perform the classification and boundary refinement of each proposal. The heuristic anchor design and the closed-set definition of the pre-trained proposal generation limit their applicability to the open-set problem. One-stage methods [370, 20, 210, 192] do not rely on the action proposal generation and can be typically trained in an end-to-end manner. These methods obtain the temporal boundaries first based on frame-level features and then perform global reasoning by multi-stage refinement or modeling the temporal transitions. Recently, AFSD [192] is proposed following the anchor-free design without actionness and proposals, which is a lightweight and flexible framework.

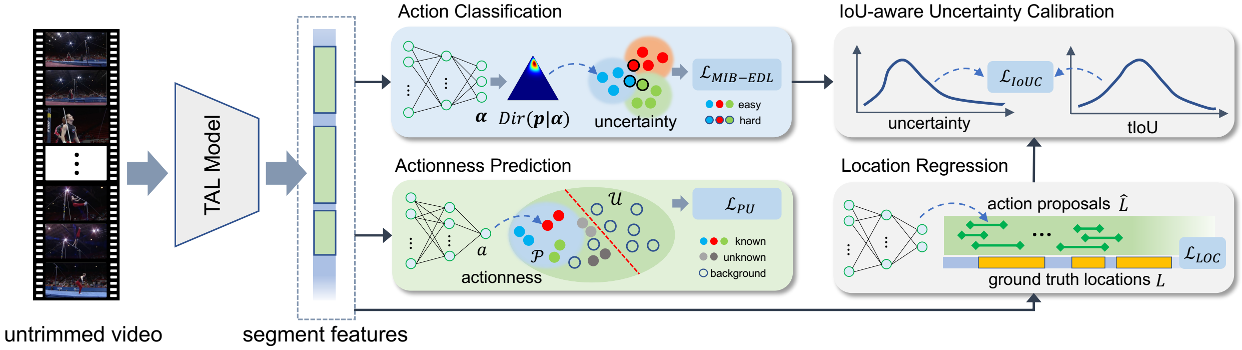


Figure 6.2 **Proposed OpenTAL.** Given untrimmed videos as input, the OpenTAL method is developed on existing TAL models (such as AFSD [192]) toward the OSTAL scenario. It consists of action classification, actionness prediction, and location regression, which are learned by the proposed MIB-EDL loss (Eq. (6.5)), PU learning (Eq. (6.6)), and localization loss (Eq. (6.7)), respectively. Furthermore, the IoU-aware uncertainty calibration is proposed to calibrate the uncertainty estimation by considering localization quality (Eq. (6.8)). In inference, with a two-step decision procedure by leveraging the uncertainty and actionness, video actions from the known and unknown classes, as well as background frames can be distinguished in the OSTAL setting (see Algorithm 6.1).

While a lot of recent methods focus on improving the proposal generation [7, 109, 195, 194, 399] or boundary refinement [192], a few focus on boosting the classification accuracy [402, 288, 413].

The above approaches assume that all of the action instances in untrimmed videos belong to pre-defined categories, which impedes their application to open-world scenarios. Though the open set is considered in [414], their method is designed for efficient annotation in few-shot learning tasks. In this paper, an OSTAL problem is formulated to handle the unknown actions in TAL applications.

Open Set Recognition Open set recognition (OSR) aims to recognize known classes and reject the unknown. The pioneering work by Scheirer et al. [278] formalized the definition of OSR and introduced an “one-vs-set" machine based on binary SVM, which inspired a line of SVM-based OSR methods [279, 123, 133]. Benefited by the deep neural networks (DNNs), Bendale et al. [16] proposed the first DNN-based OSR method OpenMax, which leverages Extreme Value Theory (EVT) to expand the K-class softmax classifier. Recently, Fang et al. [69] theoretically proved the learnability of OSR classifier and the generalization bound. Existing generative OSR methods [83, 64, 257, 147, 33, 407, 381] utilize GAN [92], generative causal model, or mixup augmentation to generate the samples of the unknown. From the reconstruction perspective, some literature [372, 249, 302] leverage VAE [143] or self-supervised learning to reconstruct the representation of known class data to identify the unknown. Prototype learning and metric learning methods [361, 290, 362, 34, 33, 391, 28] aim to identify the unknown by producing large distance to the prototype of known class data. Recently, uncertainty estimation methods [233, 316, 335] by probabilistic and evidential deep learning show promising results on OSR problems.

In this paper, we step further toward the OSTAL problem. We are aware of analogous extensions from OSR to open set object detection [224, 61, 131] and segmentation [258, 246, 333, 120]. However, it is the uniqueness of the localization in an open world that makes the OSTAL problem even more challenging and valuable in practice.

6.3 Approach

Setup Given an untrimmed video, the OSTAL task requires a model to localize all actions with temporal locations $l_i = (s_i, e_i)$, assign the actions with labels $y_i \in \{0, 1, \ldots, K\}$ where $y_i = 0$ indicates the action consisting of background frames, and reject the actions from novel classes as the unknown. In the training, the model only has access to the video data and the annotations of known actions, while the annotations of unknown actions are not given. This setting is different from the OSR problem where both annotations and data of unknown classes are not given because it is impractical in the TAL task to discard video segments of unknown actions.

Overview Fig. 6.2 shows an overview of the proposed OpenTAL. Given an untrimmed video, the features of action proposals are obtained from an existing TAL model such as the AFSD [192]. To fulfill OSTAL, we decouple the objective into three sub-tasks by a trident head, including action classification, actionness prediction, and location regression. The three branches are learned by multi-task loss functions, which will be introduced in detail.

Motivations Existing TAL models typically adopt a $(K+1)$-way action classification by assigning the background video frames with the $(K+1)$-th class *Background*. However, this paradigm is unable to handle the OSTAL case when unknown actions exist in the *Background* class.

To solve this problem, on one hand, one would attempt to append the K known classes with an additional *Unknown* category in an existing TAL system. However, this solution is practically infeasible under the OSTAL setting, because finding the video segments to train a classifier with the class *Unknown* relies on the temporal boundary annotations of unknown actions, which are not available under our OSTAL setting. Though one could relax the OSTAL setting by providing temporal annotations of the unknowns in training, learning a $(K + 1)$-way classifier is nontrivial due to the vague semantics of the *Unknown*, and this relaxation has little practical significance in an open-world where we have nothing about the prior knowledge of unknown actions. On the other hand, one may remove the *Unknown* or the *Background* class from training data, which are both infeasible under the OSTAL setting because (i) we have no temporal annotations of the unknown actions to remove them, and (ii) the pure background frames provide indispensable temporal context for action localization. Therefore, in contrast to the OSR problem, a unique technical challenge of OSTAL lies in distinguishing between actions of known and unknown classes, as well as the background frames.

Moreover, since the unknown actions are mixed with background frames without annotations, learning to distinguish foreground actions essentially reduces to a semi-supervised OSR problem [376, 276], that the model is trained with the labeled “known known" actions and the unlabeled “known unknown" actions while testing with data containing the “unknown unknown" actions[1].

To tackle these unique challenges, we propose to decouple the $(K+1)$-way action classification into K-way uncertainty-aware classification (Sec. 6.3.1) and actionness prediction (Sec. 6.3.2). Thus, we could address the first challenge above by jointly leveraging the uncertainty and actionness in a two-level decision-making (see Table 6.1) and the second challenge by the PU learning (Sec. 6.3.2).

6.3.1 Action Classification

K-way Uncertainty-aware Classification Following the existing Evidential Deep Learning (EDL) [283, 316], which is efficient to quantify the classification uncertainty, we assume a Dirichlet distribution

[1] Refer to [86, 61] for more detailed discussions on these terminologies.

Table 6.1 **Our technical motivations for the OSTAL task.** The notations ↓ and ↑ denote small and large values, repsectively.

	Known Action	Unknown Action	Background
uncertainty (u)	↓	↑	↑
actionness (a)	↑	↑	↓

$\mathrm{Dir}(\mathbf{p}|\boldsymbol{\alpha})$ over the categorical probability $\mathbf{p} \in \mathbb{R}^K$, where $\boldsymbol{\alpha} \in \mathbb{R}^K$ is the Dirichlet strength. The EDL aims to directly predict $\boldsymbol{\alpha}$ by deep neural networks (DNNs). The model is trained by minimizing the following negative log-likelihood of data $\{x_i, y_i\}$:

$$\mathcal{L}_{\text{EDL}}^{(i)}(\boldsymbol{\alpha}_i) = \sum_{j=1}^{K} t_{ij}(\log(S_i) - \log(\alpha_{ij})), \tag{6.1}$$

where t_{ij} is a binary element of the one-hot form of label y_i, and $t_{ij} = 1$ only when $y_i = j$, and $S_i = \sum_j \alpha_{ij}$ is the total strength over K classes.

In testing, given the sample x_i^*, the action classification branch (DNN) produces non-negative evidence output $\mathbf{e}_i \in \mathbb{R}_+^K$. Then, the expectation of the classification probability is obtained by $\mathbb{E}[\mathbf{p}_i] = \boldsymbol{\alpha}_i / S_i$ where $\boldsymbol{\alpha}_i = \mathbf{e}_i + 1$ according to the evidence theory [284] and subjective logic [130]. The classification uncertainty is estimated by $u_i = K/S_i$.

However, the above EDL method is empirically found ineffective in the OSTAL task since Eq. (6.1) gives equal consideration to each sample, which is practically not the case in OSTAL. In this paper, we propose to improve the generalization capability of EDL by encouraging the model to focus more on important samples in a principled way.

Momentum Importance-Balanced EDL Inspired by the recent advances in imbalanced visual classification [146, 254], the sample importance can be measured by the influence function which is determined by the gradient norm. Specifically, let $\mathbf{h}_i \in \mathbb{R}^D$ be the feature input of the last DNN layer, an exponential evidence function is applied to predict the evidence, i.e., $\mathbf{e}_i \triangleq \exp(\mathbf{w}^T \mathbf{h}_i)$ where $\mathbf{w} \in \mathbb{R}^{D \times K}$ are the learnable weights of the DNN layer. The gradient $\mathbf{g}_i$ of the EDL loss $\mathcal{L}_{\text{EDL}}^{(i)}$ w.r.t. the logits $\mathbf{z}_i \triangleq \mathbf{w}^T \mathbf{h}_i$ is derived:

$$g_{ij} = \frac{\partial \mathcal{L}_{\text{EDL}}^{(ij)}}{\partial z_{ij}} = t_{ij}\left[\frac{S_i - K\alpha_{ij}}{S_i \alpha_{ij}}\right] = t_{ij}\left[\frac{1}{\alpha_{ij}} - u_i\right], \tag{6.2}$$

where the chain rule and the equality $u_i = K/S_i$ are used. Since $t_{ij} = 0$ when $j \neq y_i$, it is interesting to see a simple but meaningful gradient form, i.e., $g_{ik} = 1/\alpha_{ik} - u_i$ where $k = y_i$, and in our supplement, we proved that $|g_{ik}| \in [0, 1)$.

Furthermore, inspired by [254], we consider the influence function given by the gradient norm of EDL loss w.r.t. the network parameters $\mathbf{w}$. According to the chain rule of $\mathbf{z}_i = \mathbf{w}^T \mathbf{h}_i$, the influence value ω_i can be derived:

$$\omega_i = \left(\sum_{k=1}^{K} |g_{ik}| \right) \left(\sum_{d=1}^{D} |h_{id}| \right) = \|\mathbf{g}_i\|_1 \cdot \|\mathbf{h}_i\|_1. \tag{6.3}$$

Detailed proof can be found in the supplement. We define the loss weight of sample x_i as the moving mean of influence values within the neighboring region of $\|\mathbf{g}_i\|_1$:

$$\tilde{\omega}_i^{(t)} = \epsilon \cdot \tilde{\omega}_i^{(t-1)} + (1 - \epsilon) \cdot \frac{1}{|\Omega_m|} \sum \Omega_m, \tag{6.4}$$

where Ω_m is a subset of ω_i whose gradient norm $\|\mathbf{g}_i\|_1$ falls into the m-th bin out of total M bins in the region $[0, 1]$, i.e., $\Omega_m = \{\omega_i | \|\mathbf{g}_i\|_1 \in [\frac{m-1}{M}, \frac{m}{M}], m = 1, \ldots, M\}$. The ϵ is a momentum factor within $[0, 1]$, M is a constant, and t is the training iteration. We set the initial weight $\tilde{\omega}_i^{(0)}$ as the 1.0. A larger ϵ means the set of influence values ω_i are less considered, while M controls the granularity of the neighborhood of the gradient norm. Eventually, the proposed Momentum Importance-Balanced (MIB) EDL loss is defined as:

$$\mathcal{L}_{\text{MIB-EDL}} = \frac{1}{N} \sum_{i=1}^{N} \tilde{\omega}_i^{(t)} \mathcal{L}_{\text{EDL}}^{(i)}(\boldsymbol{\alpha}_i). \tag{6.5}$$

The proposed MIB-EDL loss encourages the model to smoothly focus on important samples as the training iteration increases. In practice, to stabilize the training, the re-weighting is applied after T_0 training iterations. Different to [254] that uses the inverse of ω_i to down-weight the influential samples for a balanced closed-set recognition, we use Eq. (6.3) to up-weight these samples for open-set recognition, and (6.4) to achieve a smooth update of the sample weight.

6.3.2 Actionness Prediction

Due to the mixture of unknown actions and pure background frames, it is not sufficient to distinguish between them by the evidential uncertainty over K known classes. Therefore, predicting the actionness that indicates how likely a sample is a foreground action is critical. We notice the fact that data from known classes are *positive* data while the samples from the “background" mixture are *unlabeled*. This intrinsically reduces to a semi-supervised learning problem called positive-unlabeled (PU) learning [14]. In this paper, we propose a simple yet effective PU learning method to predict the actionness.

Let $\hat{a}_i \in [0, 1]$ be the predicted actionness score of the sample x_i, the actionness in a training batch $\hat{\mathcal{A}} = \{\hat{a}_i\}$ can be splitted into the positive set $\hat{\mathcal{P}} = \{\hat{a}_i | y_i \geq 1\}$ and the unlabeled background set $\hat{\mathcal{U}} = \{\hat{a}_i | y_i = 0\}$. In this paper, we propose to ascendingly sort the $\hat{\mathcal{U}}$ and select top-M samples to form the most likely negative set $\hat{\mathcal{N}} = \{\hat{a}_i | \hat{a}_i \in sort(\hat{\mathcal{U}})_{1,...,M}\}$. Then, a binary cross-entropy (BCE) loss could be applied to the $\hat{\mathcal{P}}$ and $\hat{\mathcal{N}}$:

$$\mathcal{L}_{\text{ACT}}(\hat{\mathcal{P}}, \hat{\mathcal{N}}) = -\frac{1}{|\hat{\mathcal{P}}|} \sum_{\hat{a}_i \in \hat{\mathcal{P}}} \log \hat{a}_i - \frac{1}{|\hat{\mathcal{N}}|} \sum_{\hat{a}_i \in \hat{\mathcal{N}}} \log(1 - \hat{a}_i). \tag{6.6}$$

Here, to achieve a balanced BCE training, we set the size of negative set to $M = |\hat{\mathcal{N}}| := \min(|\hat{\mathcal{P}}|, |\hat{\mathcal{U}}|)$ considering that in most training batches we have $|\hat{\mathcal{U}}| \gg |\hat{\mathcal{P}}|$. This BCE loss will push the probably pure background samples far away from positive actions. Though this method is straightforward, the learned actionness scores are found discriminative enough to distinguish between the foreground actions and background frames in the OSTAL setting (see Fig. 6.4a).

6.3.3 Location Regression

To maintain the flexibility of our method on existing TAL models, the temporal location regression follows the design of the TAL models. Take the state-of-the-art TAL model AFSD [192] as an example, it consists of a coarse stage to predict the location proposals $\hat{l}_i = [\hat{s}_i, \hat{e}_i]$ and a refined stage to predict the temporal offset $\hat{\delta}_i = [\hat{\delta}_i^{(s)}, \hat{\delta}_i^{(e)}]$ with respect to the $\hat{l}_i$. The coarse stage is learned by temporal Intersection-over-Union (tIoU) loss, while the refined stage is learned by an L_1 loss:

$$\begin{cases} \mathcal{L}_{\text{LOC}}(\{\hat{l}_i\}) = \dfrac{1}{N_C} \displaystyle\sum_i \mathbb{I}[y_i \geq 1] \left(1 - \dfrac{|\hat{l}_i \cap l_i|}{|\hat{l}_i \cup l_i|}\right) \\ \mathcal{L}_{\text{LOC}}(\{\hat{\delta}_i\}) = \dfrac{1}{N_R} \displaystyle\sum_i \mathbb{I}[y_i \geq 1](|\hat{\delta}_i - \delta_i|), \end{cases} \tag{6.7}$$

where N_C and N_R are the corresponding number of samples that are matched with the ground truth action locations by an IoU threshold. The indicator function $\mathbb{I}[y_i \geq 1]$ filters out the unmatched samples which are treated as the "background" data. In testing, the predicted location is recovered by $l_i^* = [\hat{s}_i + 0.5(\hat{e}_i - \hat{s}_i)\hat{\delta}_i^{(s)}, \hat{e}_i + 0.5(\hat{e}_i - \hat{s}_i)\hat{\delta}_i^{(e)}]$. Note that our OpenTAL framework is not limited to specific TAL models but is general in design.

6.3.4 IoU-aware Uncertainty Calibration

Though the loss functions defined by Eqs. (6.5)(6.6)(6.7) is sufficient for a complete OSTAL task, the learned uncertainty in the classification module is not calibrated by considering the localization performance. Intuitively, an action proposal of high temporal overlap with the ground truth location should contain more evidence and thus low uncertainty. To this end, we propose a novel IoU-aware uncertainty calibration method:

$$\mathcal{L}_{\text{IoUC}}^{(i)}(\hat{l}_i, u_i) = -w_{\hat{l}_i, l_i} \log(1 - u_i) - (1 - w_{\hat{l}_i, l_i}) \log(u_i) \tag{6.8}$$

where the weight w is a clipped form of the temporal IoU between the predicted and ground truth locations:

$$w_{\hat{l}_i, l_i} = \max\left(\gamma, \text{IoU}(\hat{l}_i, l_i)\right) \tag{6.9}$$

where the γ is a small non-negative constant. The cross-entropy form in Eq. (6.8) and (6.9) will encourage the model to produce high uncertainty ($u_i \rightarrow 1$) for action proposals with low localization quality ($w \rightarrow \gamma$).

The motivation behind the clipping by $\max()$ is that given the ground truth of known actions, both the proposals of background frames and unknown actions are not overlapped with the ground truth such that $\text{IoU}(\hat{l}_i, l_i) \leq 0$, the clipping could avoid reversing the loss value from positive to negative, while still maintaining a low localization quality γ. Besides, it is reasonable to encourage high uncertainty u_i by small γ for the location proposals of the background and the unknown actions in the OSTAL setting.

Algorithm 6.1 Inference Procedure

```
Require: Untrimmed test video.
Require: Trained OpenTAL model.
Require: Threshold $\tau$ from training data by Eq. (6.11)
Data pre-processing (if applicable).
Predict proposals $\mathcal{G}=\{l_i^*, \hat{y}_i, u_i, \hat{a}_i\}|_{i=1}^{N}$ by OpenTAL.
Post-processing (if applicable).
for each proposal $\mathcal{G}_i \in \mathcal{G}$ do
    if $\hat{a}_i < 0.5$ then                                  ▷ Background
        $\mathcal{G}_i$ is a Background; continue.
    end if
    if $u_i > \tau$ then                                       ▷ Unknown Action
        $\mathcal{G}_i$ is Unknown.
    else                                                       ▷ Known Action
        $\mathcal{G}_i$ is Known by $\hat{y}_i = \arg\max_j \mathbb{E}[\mathbf{p}_{ij}]$.
    end if
end for
```

6.3.5 Training and Inference

The training procedure is to minimize the weighted sum of losses defined by Eqs. (6.5)(6.6)(6.7)(6.8):

$$\mathcal{L} = \mu\mathcal{L}_{\text{MIB-EDL}} + \mathcal{L}_{\text{ACT}} + \mathcal{L}_{\text{LOC}} + \mathbb{E}[\mathcal{L}_{\text{IoUC}}^{(i)}], \tag{6.10}$$

where μ is a hyperparameter, and $\mathbb{E}[\cdot]$ is to take the mean loss values over the input samples.

In inference, the untrimmed video input is fed into a TAL model, and our OpenTAL method trained on the TAL model could produce multiple action locations $\{l_i^*\}$, classification labels $\hat{y}_i = \arg\max_{j\in[1,\dots,K]} \mathbb{E}[\mathbf{p}_{ij}]$, classification uncertainty u_i, and actionness score $\hat{a}_i$. Together with the u_i and $\hat{a}_i$, a positively localized foreground action x_i, i.e., $a_i > 0.5$, can be accepted as known class $\hat{y}_i$, or rejected as the unknown by the following simple scoring function:

$$P(x_i|a_i > 0.5) = \begin{cases} unknown, & \text{if } u_i > \tau, \\ \hat{y}_i, & \text{otherwise.} \end{cases} \tag{6.11}$$

The complete inference procedure is shown in Algorithm 6.1. In addition to this two-level decision, one-level decisions by the functional formulas of $P(x_i)$ w.r.t. to u_i and a_i are also plausible (see Table 6.5). However, we empirically found that Eq. (6.11) is the most effective formula while maintaining the explainable nature of decision-making.

6.4 Experiments

6.4.1 Implementation Details

Our method is implemented on the AFSD [192] model[2], which is a state-of-the-art TAL model. Pre-trained I3D [27] backbone is used in AFSD. The proposed OpenTAL is applied to both the coarse and refined stages of AFSD. Specifically, the proposed MIB re-weighting is applied after 10 training epochs. We empirically set the momentum ϵ to 0.99 and the number of bins M to 50. The small constant γ in Eq. (6.9) is set to 0.001. The loss weight μ in Eq. (6.10) is set to 10. We trained the model 25 epochs to ensure full convergence. The rest settings in AFSD are kept unchanged.

6.4.2 Datasets

THUMOS14 [128] and ActivityNet1.3 [22] are two commonly-used datasets for TAL evaluation. The THUMOS14 dataset contains 200 training videos and 212 testing videos. ActivityNet1.3 dataset contains about 20K videos with 200 human activity categories. Since our method is not limited by data modality, we use RGB videos for training and testing by default. To enable OSTAL evaluation, we randomly select 3/4 THUMOS14 categories of the training videos as the known data. This random selection is repeated to generate three THUMOS14 open set splits between the known and the unknown. Considering that ActivityNet1.3 is newer and covers most THUMOS14 categories, ActivityNet1.3 is not suitable to be the closed-set training data when the model is tested on THUMOS14. Therefore, we train models on the THUMOS14 known split and use the THUMOS14 unknown split and the disjoint categories of ActivityNet1.3 as two sources of open set testing data. To get the disjoint categories from ActivityNet1.3, we manually removed 14 semantically overlapping categories by referring to the THUMOS14 categories. Detailed dataset information could be found in our supplement.

6.4.3 Evaluation Protocols

The mean Average Precision (**mAP**) is typically used for the evaluation of closed-set TAL performance. To enable OSTAL performance evaluation, the Area Under the Receiver Operating Characteristic (**AUROC**) curve and the Area Under the Precision-Recall (**AUPR**) are introduced to

[2] AFSD: https://github.com/TencentYoutuResearch/ActionDetection-AFSD

evaluate the performance of detecting the unknown from the known actions for positively localized actions. To address the operational meaning in practice, we additionally report the False Alarm Rate at a True Positive Rate of 95% (**FAR@95**), by which a smaller value indicates better performance. However, we noticed the metrics above are insufficient for the OSTAL task because the multi-class classification performance of the known classes in the OSTAL setting is ignored. Inspired by the Open Set Classification Rate [62, 240, 33], we propose the Open Set Detection Rate (**OSDR**), which is defined as the area under the curve of Correct Detection Rate (CDR) and False Positive Rate (FPR). Given an operation point τ of the scoring function $P(x)$ for detecting the unknown and an operation point t_0 of the tIoU for localizing the foreground actions, CDR and FPR are defined as:

$$\begin{cases} \text{CDR}(\tau, t_0) = \dfrac{|\{x|(x \in \mathcal{F}_k) \wedge (\hat{f}_{x|y} = y) \wedge P(x) < \tau\}|}{|\mathcal{F}_k|} \\ \text{FPR}(\tau, t_0) = \dfrac{|\{x|(x \in \mathcal{F}_u) \wedge P(x) < \tau\}|}{|\mathcal{F}_u|} \end{cases} \tag{6.12}$$

where $\mathcal{F}_k$ is the set of positively localized known actions, i.e., $\mathcal{F}_k = \{x|(\text{tIoU} > t_0) \wedge (y \in [1, \ldots, K])\}$, and $\mathcal{F}_u$ is the set of positively localized unknown actions, i.e., $\mathcal{F}_u = \{x|(\text{tIoU} > t_0) \wedge (y = 0)\}$. The CDR indicates the fraction of known actions that are positively localized and correctly classified into their known classes, while the FPR denotes the fraction of unknown actions that are positively localized but falsely accepted as an arbitrary known class. Higher OSDR indicates better performance for the OSTAL task.

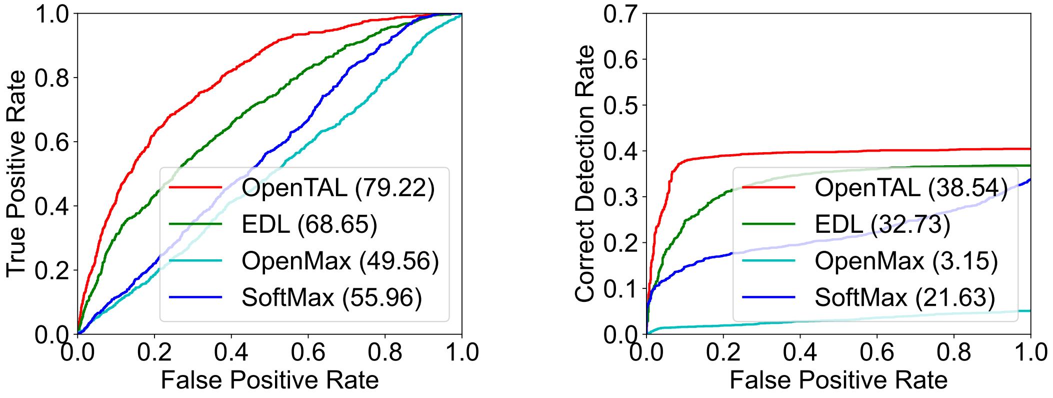
For stable evaluation, all results are reported by averaging the results of each evaluation metric over the three THUMOS14 splits. Results are reported at the tIoU threshold 0.3 for THUMOS14 and 0.5 for ActivityNet1.4, and results by other thresholds are in the supplement.

6.4.4 Comparison with State-of-the-arts

The OpenTAL method is compared with the following baselines based on the AFSD: (1) **SoftMax**: use the softmax confidence score to identify the unknown. (2) **OpenMax**: use OpenMax [16] in testing to append the softmax scores with unknown class. (3) **EDL**: similar to [316], the vanilla EDL method is used to replace the traditional cross-entropy loss for uncertainty quantification. Models are tested using both the THUMOS14 unknown spits and the ActivityNet1.3

Table 6.2 **OSTAL Results (%).** Models trained on the THUMOS14 closed set are tested on the open sets by including the unknown classes from THUMOS14 and ActivityNet1.3, respectively. The mAP is provided as the reference of the TAL results on the THUMOS14 closed set.

Methods	THUMOS14 as the Unknown				ActivityNet1.3 as the Unknown				mAP
	FAR@95 (↓)	AUROC	AUPR	OSDR	FAR@95 (↓)	AUROC	AUPR	OSDR	
SoftMax	85.58	54.70	31.85	23.40	85.05	56.97	53.54	27.63	55.81
OpenMax [16]	90.34	53.26	33.17	13.66	91.36	51.24	54.88	15.73	36.36
EDL [316]	81.42	64.05	40.05	36.26	84.01	62.82	53.97	38.56	52.24
OpenTAL	**70.96**	**78.33**	**58.62**	**42.91**	**63.11**	**82.97**	**80.41**	**50.49**	55.02



(a) ROC Curves

(b) OSDR Curves

Figure 6.3 **ROC and OSDR curves on one THUMOS14 split.** Numbers in the brackets are AUROC or OSDR values.

disjoint subset. Results are reported in Table 6.2.

The results show that the OpenTAL outperforms the baselines by large margins on all OSTAL metrics, while still keeping comparable closed set TAL performance (less than 1% mAP decrease). The results also show that OpenMax does not work well on the OSTAL task, especially when the large-scale ActivityNet1.3 dataset is used as the unknown. The EDL works well but is still far behind the proposed OpenTAL. Fig. 6.3 shows the detailed evaluation by the curves of AUROC and OSDR on one THUMOS14 split. They clearly show that the proposed OpenTAL on different operation points of scoring values and different open set splits is consistently better than the baselines.

Table 6.3 **Ablation Results (%).** The proposed EDL re-weighting method (MIB), the actionness prediction (ACT), and the IoUC loss are individually ablated from the OpenTAL.

Variants	MIB	ACT	IoUC	FAR@95 (↓)	AUROC	AUPR	OSDR
(1)		✓	✓	77.20	76.41	56.65	12.10
(2)	✓		✓	82.85	58.12	31.80	37.89
(3)	✓	✓		79.64	62.73	37.86	39.39
OpenTAL	✓	✓	✓	**70.96**	**78.33**	**58.62**	**42.91**

6.4.5 Ablation Study

Component Ablation. By individually removing the major components of OpenTAL, three model variants are compared. (1) Without MIB: the proposed MIB re-weighting is removed so that the vanilla EDL loss (Eq. (6.1)) is used. (2) Without ACT: the actionness prediction is removed so that the $(K+1)$-way classification in $\mathcal{L}_{\text{MIB-EDL}}$ (Eq. (6.5)) is adopted. (3) Without IoUC: the loss $\mathcal{L}_{\text{IoUC}}$ (Eq. (6.8)) is removed from the training. Results are reported in Table 6.3. They show that OpenTAL achieves the best performance. Specifically, the MIB re-weighting strategy contributes the most to the OSDR performance gain by around 30%. The actionness prediction (ACT) contributes the most to the FAR@95, AUROC, and AUPR metrics. Besides, the proposed IoUC loss also leads to significant performance gains on all metrics. These observations demonstrate the effectiveness of the three components for the OSTAL task.

Choices of Re-weighting Methods. We compare the proposed MIB re-weighting method (MIB (soft)) with the MIB (hard) and existing literature on sample re-weighting in Table 6.4. The results show that the focal loss (Focal) [197] does not work well with the OpenTAL framework. GHM [170] and IB [254] methods could achieve comparable FAR@95, AUROC, and AUPR performance, but their OSDR results are still largely far behind ours. Note that these methods are all designed for closed-set recognition, thus the proposed MIB is more suitable for open-set scenarios. Besides, the hard version of MIB, i.e., the momentum mechanism is removed by setting the ϵ to 0, could improve about 4% FAR@95 while sacrificing the AUROC, AUPR, and OSDR.

Table 6.4 **Results of Different Re-weightings (%).** MIB (hard) means the momentum factor $\epsilon = 0$ in Eq. (6.4) such that the sample weight is updated in a hard manner, while the MIB (soft) sets the ϵ to 0.99 to enable a soft update, and wo. Re-weight means $\epsilon = 1.0$.

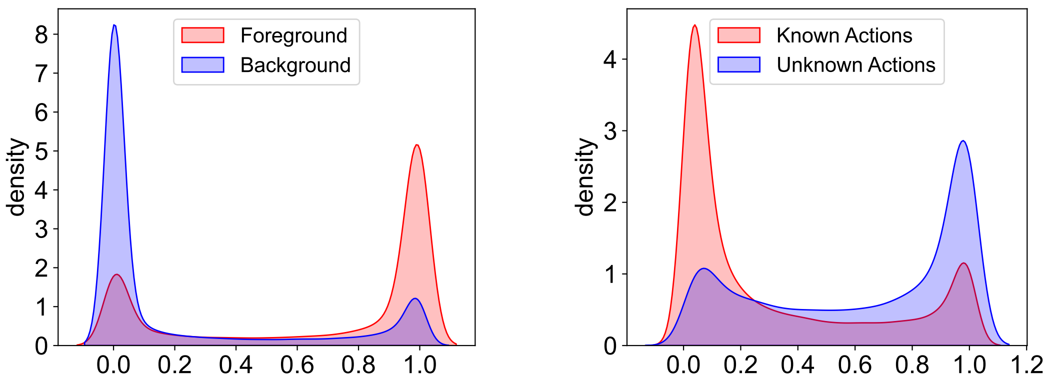
Methods	FAR@95 (↓)	AUROC	AUPR	OSDR
wo. Re-weight.	77.20	76.41	56.65	12.10
Focal [197]	91.05	56.67	35.55	2.04
GHM [170]	78.33	73.52	54.03	1.41
IB [254]	80.23	75.91	58.00	2.18
MIB (hard)	**66.34**	78.16	57.66	38.90
MIB (soft)	70.96	**78.33**	**58.62**	**42.91**

Choices of Scoring Function. The scoring function is critical to identify the known and unknown actions, as well as the background frames in model inference. In addition to the proposed two-level decision by (6.11), we compare it with four reasonable one-level decision methods by utilizing actionness a_i and uncertainty u_i. The results in Table 6.5 show that using the maximum classification confidence (the 1st row) or other compositions of u_i and a_i (the 2nd and 3rd rows) cannot achieve favorable performance. The proposed method (the last row) is slightly better than the product between u_i and a_i (the 4-th row) with comparable FAR@95 performance. Though there are certainly other alternatives, our scoring function achieves the best performance while maintaining a good decision-making explanation, which means that the foreground actions are identified first by a_i, based on which the known and unknown actions are further distinguished by u_i.

Distributions of Actionness and Uncertainty. To show the quality of the learned actionness and uncertainty, we visualized their distributions on the test set in Fig. 6.4. Specifically, the dominant modes in Fig. 6.4a show that foreground actions are majorly assigned with high actionness while the background frames are with low actionness, and the dominant modes in Fig 6.4b show that the actions of known classes are majorly assigned with low uncertainty while those of the unknowns are with high uncertainty. These observations align well with the expectations of our OpenTAL method.

Table 6.5 **Scoring Functions.** It shows when conditioned on $a_i > 0.5$, uncertainty u_i is the best scoring function for the OSTAL task.

Scoring Functions	FAR@95 (↓)	AUROC	AUPR	OSDR
$P(x_i) = 1 - \max_j(\alpha_i/S_i)$	77.90	59.50	35.82	31.38
$P(x_i) = u_i/(1 - a_i)$	79.16	61.94	38.52	30.64
$P(x_i) = a_i/(1 - u_i)$	90.39	72.71	56.19	38.24
$P(x_i) = u_i \cdot a_i$	**70.64**	77.52	58.17	42.44
$P(x_i \mid a_i > 0.5) = u_i$	70.96	**78.33**	**58.62**	**42.91**



(a) Actionness

(b) Uncertainty

Figure 6.4 **Distributions of Actionness and Uncertainty.** The two figures show significant separation between the foreground actions and background frames by actionness score, as well as the separation between the known and unknown actions by uncertainty.

Qualitative Results. Fig. 6.5 shows the qualitative results of the proposed OpenTAL and baseline approaches. The three video samples are from the THUMOS14 dataset. The results clearly show that OpenTAL is superior to baselines in terms of both recognizing the known actions (colored segments in the 1st video), and rejecting the unknown actions (black segments in the 2nd and 3rd videos).

Limitations. We note that all those methods do not show remarkably high OSDR performance, which indicates the challenging nature of the OSTAL task and there exists large room for improvement in the OpenTAL.

6.5 Conclusion

In this paper, we introduce the Open Set Temporal Action Localization (OSTAL) task. It aims to simultaneously localize and recognize human actions, and to reject the unknown actions from

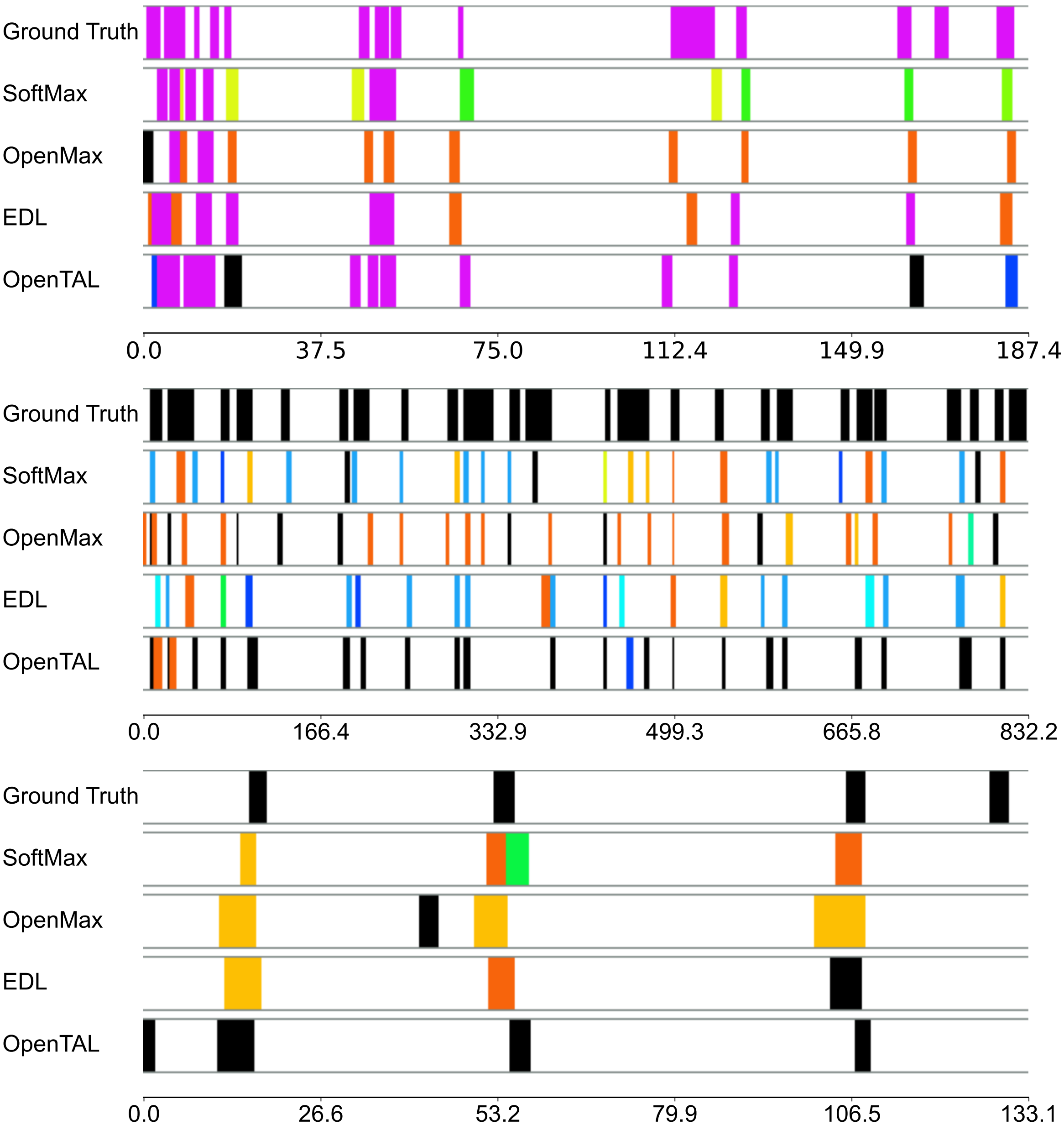


Figure 6.5 **Qualitative Results.** We show the actions of unknown classes with black color, while the rest colors are actions of known classes. The x-axis represents the timestamps (seconds).

untrimmed videos in an open world. The unique challenge lies in discriminating between known and unknown actions as well as background video frames. To this end, we propose a general OpenTAL framework to enable existing TAL models for the OSTAL task. The OpenTAL predicts the locations, classifications with uncertainties, and actionness to jointly achieve the goal. For comprehensive OSTAL evaluation, the Open Set Detection Rate is introduced. The OpenTAL is empirically demonstrated to be effective and significantly outperforms existing baselines. We believe the generality of the OpenTAL design could inspire relevant research fields such as spatiotemporal action detection, video object detection, and video grounding toward open set scenarios.

6.6 Supplementary Material

In this section, we provide the detailed proof of the gradient of the EDL loss (Sec. 6.6.1), the dataset description of the open set setting (Sec. 6.6.2), implementation details (Sec. 6.6.3), additional results and discussions (Sec. 6.6.4).

6.6.1 Gradient of EDL

Given the DNN logits $\mathbf{z}_i \in \mathbb{R}^K$ of sample x_i, an evidence function defined by exp is applied to the logits to get the class-wise evidence prediction, i.e., $\mathbf{e}_i = \exp(\mathbf{z}_i)$. Following the maximum likelihood loss form of Evidential Deep Learning (EDL) [283], we have the EDL loss:

$$\mathcal{L}_{\text{EDL}}^{(i)}(\boldsymbol{\alpha}_i) = \sum_{j=1}^{K} t_{ij}(\log(S_i) - \log(\alpha_{ij})), \tag{6.13}$$

where $t_{ij} = 1$ iff. the class label $y_i = j$, otherwise $t_{ij} = 0$. The total Dirichlet strength $S_i = \sum_j \alpha_{ij}$ and the class-wise strength $\boldsymbol{\alpha}_i = \mathbf{e}_i + 1$. Therefore, according to the simple chain rule, we have the partial derivative:

$$\frac{\partial \alpha_{ij}}{\partial z_{ij}} = \frac{\partial \alpha_{ij}}{\partial e_{ij}} \cdot \frac{\partial e_{ij}}{\partial z_{ij}} = e_{ij} \tag{6.14}$$

Then, the gradient of the j-th entry in Eq. (6.13), i.e., $\mathcal{L}_{\text{EDL}}^{(ij)}$, w.r.t. the logits z_{ij} can be derived as follows:

$$\begin{aligned} g_{ij} = \frac{\partial \mathcal{L}_{\text{EDL}}^{(ij)}}{\partial z_{ij}} &= t_{ij}\left[\frac{1}{S_i}\frac{\partial S_i}{\partial z_{ij}} - \frac{1}{\alpha_{ij}}\frac{\partial \alpha_{ij}}{\partial z_{ij}}\right] \\ &= t_{ij}\left[\frac{1}{S_i}\sum_{k=1}^{K}\frac{\partial \alpha_{ik}}{\partial z_{ij}} - \frac{e_{ij}}{\alpha_{ij}}\right] \\ &= t_{ij}\left[\frac{1}{S_i}\sum_{k=1}^{K} e_{ik} - \frac{e_{ij}}{\alpha_{ij}}\right] \end{aligned} \tag{6.15}$$

Consider that $S_i = \sum_k \alpha_{ik} = \sum_j e_{ik} + K$, and the evidential uncertainty $u_i = K/S_i$, we further simplify the g_{ij} as follows:

$$\begin{aligned} g_{ij} &= t_{ij} \left[\frac{S_i - K}{S_i} - \frac{\alpha_{ij} - 1}{\alpha_{ij}} \right] \\ &= t_{ij} \left[\frac{S_i - K\alpha_{ij}}{S_i \alpha_{ij}} \right] \\ &= t_{ij} \left[\frac{1}{\alpha_{ij}} - u_i \right], \end{aligned} \tag{6.16}$$

which has proved the equation of g_{ij} in our main paper. From this conclusion, when considering that $\alpha_{ij} \in (1, \infty)$ and $u_i \in (0, 1)$, we have the property $|g_{ij}| \in [0, 1)$.

Furthermore, consider the last DNN layer parameters $\mathbf{w} \in \mathbb{R}^{D \times K}$ such that $\mathbf{z}_i = \mathbf{w}^T \mathbf{h}_i$ where $\mathbf{h}_i \in \mathbb{R}^D$ is the high-dimensional feature of x_i, we can derive the gradient of EDL loss w.r.t. parameters $\mathbf{w}$:

$$\nabla_w \mathcal{L} = \frac{\partial \mathcal{L}_{\text{EDL}}^{(ij)}}{\partial w_{dk}} = \frac{\partial \mathcal{L}_{\text{EDL}}^{(ik)}}{\partial z_{ik}} \cdot \frac{\partial z_{ik}}{\partial w_{dk}} = g_{ik} \cdot h_{id}, \tag{6.17}$$

where w_{dk} and h_{id} are elements of the matrix $\mathbf{w}$ and the vector $\mathbf{h}_i$. Similar to [254], we consider the influence function [146] by ignoring the inverse of Hessian and using the magnitude (L_1 norm) of the gradient:

$$\begin{aligned} \omega_i = \|\nabla_w \mathcal{L}\|_1 &= \sum_{k=1}^{K} \sum_{d=1}^{D} |g_{ik} \cdot h_{id}| \\ &= \left(\sum_{k=1}^{K} |g_{ik}| \right) \left(\sum_{d=1}^{D} |h_{id}| \right) \\ &= \|\mathbf{g}_i\|_1 \cdot \|\mathbf{h}_i\|_1, \end{aligned} \tag{6.18}$$

which has proved the equation of ω_i in our main paper.

6.6.2 Dataset Details

To enable the existing Temporal Action Localization (TAL) datasets such as THUMOS14 [128] and ActivityNet1.3 [22] for the open set TAL setting, a subset of action categories has to be reserved as the unknown used in open set testing. In practice, we randomly splitted the THUMOS14 three times into known and unknown subsets of categories. For each split, a model will be trained on the closed set (which only contains known categories), and tested on the open set that contains both

Table 6.6 **THUMOS14 Splits for Open Set TAL.** For each split, five out of twenty action categories are randomly selected as the unknown (**U**) used in open set testing, while the rest fifteen categories are the known (**K**) used in model training.

	Split 1	Split 2	Split 3
BaseballPitch	K	K	K
BasketballDunk	K	K	K
Billiards	K	K	K
CricketBowling	K	U	K
CricketShot	K	K	U
FrisbeeCatch	K	K	K
GolfSwing	K	K	K
HammerThrow	K	U	K
HighJump	K	K	K
JavelinThrow	K	U	U
PoleVault	K	K	U
Shotput	K	K	U
TennisSwing	K	K	K
ThrowDiscus	K	K	K
VolleyballSpiking	K	K	K
CleanAndJerk	U	K	K
CliffDiving	U	U	K
Diving	U	U	K
LongJump	U	K	U
SoccerPenalty	U	K	K

known and unknown categories. Table 6.6 shows the detailed information of the three dataset splits from THUMOS14.

To further increase the openness in testing, we incorporate activity categories from ActivityNet1.3 that are non-overlapped with THUMOS14 into the open set testing. Specifically, the following 14 overlapping activity categories are removed: *Table soccer, Javelin throw, Clean and jerk, Springboard diving, Pole vault, Cricket, High jump, Shot put, Long jump, Hammer throw, Snatch, Volleyball, Plataform diving, Discus throw*. Note that we did not use ActivityNet1.3 for similar model training as the THUMOS14, e.g., train a model on multiple random splits of ActivityNet1.3, due to the limited computational resource.

Table 6.7 **AUROC Results (%) vs. Different tIoU Thresholds**. Models trained on the THUMOS14 closed set are tested by including the unknown classes from THUMOS14 and ActivityNet1.3, respectively. Results are averaged over the three dataset splits.

Methods	THUMOS14 as the Unknown						ActivityNet1.3 as the Unknown			
	0.3	0.4	0.5	0.6	0.7	Avg.	0.5	0.75	0.95	Avg.
SoftMax	54.70	55.46	56.41	57.12	57.11	56.16	56.97	58.41	55.97	57.77
OpenMax [16]	53.26	52.1	52.13	51.89	52.53	52.38	51.24	52.39	49.13	51.59
EDL [316]	64.05	64.27	65.13	66.21	66.81	65.29	62.82	66.23	67.92	65.69
OpenTAL	**78.33**	**79.04**	**79.30**	**79.40**	**79.82**	**79.18**	**82.97**	**83.21**	**83.38**	**83.22**

Table 6.8 **AUPR Results (%) vs. Different tIoU Thresholds**. Models trained on the THUMOS14 closed set are tested by including the unknown classes from THUMOS14 and ActivityNet1.3, respectively. Results are averaged over the three dataset splits.

Methods	THUMOS14 as the Unknown						ActivityNet1.3 as the Unknown			
	0.3	0.4	0.5	0.6	0.7	Avg.	0.5	0.75	0.95	Avg.
SoftMax	31.85	31.81	31.11	29.78	27.99	30.51	53.54	44.15	34.54	44.77
OpenMax [16]	33.17	31.61	30.59	29.15	28.45	30.60	54.88	48.37	40.07	48.48
EDL [316]	40.05	39.45	38.05	37.58	36.35	38.30	53.97	47.22	45.59	48.46
OpenTAL	**58.62**	**59.40**	**58.78**	**57.54**	**55.88**	**58.04**	**80.41**	**74.20**	**73.92**	**75.54**

Table 6.9 **OSDR Results (%) vs. Different tIoU Thresholds**. Models trained on the THUMOS14 closed set are tested by including the unknown classes from THUMOS14 and ActivityNet1.3, respectively. Results are averaged over the three dataset splits.

Methods	THUMOS14 as the Unknown						ActivityNet1.3 as the Unknown			
	0.3	0.4	0.5	0.6	0.7	Avg.	0.5	0.75	0.95	Avg.
SoftMax	23.40	25.19	27.43	29.97	32.08	27.61	27.63	33.73	31.59	32.01
OpenMax [16]	13.66	14.58	15.91	17.71	20.41	16.45	15.73	21.49	18.07	19.35
EDL [316]	36.26	37.58	39.16	41.18	42.99	39.43	38.56	43.72	42.20	42.18
OpenTAL	**42.91**	**46.19**	**49.50**	**52.50**	**56.78**	**49.57**	**50.49**	**59.87**	**62.17**	**57.89**

6.6.3 Implementation Details

Detailed Architecture The proposed OpenTAL is primarily implemented on the AFSD [192] framework. It uses a pre-trained I3D [27] as the feature extraction backbone and a 6-layer temporal FPN architecture is applied to the I3D for action classification and localization. Each level consists of a coarse stage, a saliency-based proposal refinement module, and a refined stage. The first two pyramid levels use 3D convolutional (Conv3D) block while the rest four levels use 1D convolutional

(Conv1D) block. Group Normalization and ReLU activation are utilized in each block. The temporal localization head and action classification head are implemented by a shared Conv1D block across all 6 levels. To implement OpenTAL method, the $(K + 1)$-way classification head is replaced with K-way evidential neural network head, while the localization head is kept unchanged. We additionally add an actionness prediction branch which consists of a Conv1D block for both the coarse and the refined stages.

Training and Testing In training, the proposed classification loss $\mathcal{L}_{\text{MIB-EDL}}$ and actionness prediction loss $\mathcal{L}_{\text{ACT}}$ are applied to both the coarse and refined stages in AFSD, while the IoU-aware uncertainty calibration loss $\mathcal{L}_{\text{IoUC}}$ is only applied to the refined stage because this loss function is dependent of the pre-computed temporal IoU using the predicted action locations in the coarse stage. Similar to AFSD, we used temporal IoU threshold 0.5 in the training to identify the foreground actions from the proposals. Besides, we reduced the weight of triplet loss in AFSD to 0.001 since the contrastive learning loss would not work well when there are unknown action clips in the background. The whole model is trained by Adam optimizer with base learning rate 1e-5 and weight decay 1e-3. All models are trained with 25 epochs to ensure full convergence and the model snapshot of the last epoch is used for testing and evaluation.

In testing, the actionness score is multiplied to the confidence score before the soft-NMS post-processing module. The σ and top-N hyperparameters are set to 0.5 and 5000, which are recommended by the AFSD.

6.6.4 Additional Results

Impact of tIoU Thresholds Since the proposed OSTAL task cares not only the classification but also the temporal localization, we present the experimental results under different temporal IoU (tIoU) thresholds. Following existing TAL literature, we set five tIoU thresholds [0.3 : 0.1 : 0.7] when the unknown classes are from THUMOS14 and ten tIoU thresholds [0.5 : 0.05 : 0.95] when the unknown classes are from ActivityNet1.3, respectively. Evaluation results by AUROC, AUPR, and OSDR are reported in Table 6.7, 6.8, and 6.9, respectively. The results show that AUROC performances are stable across different tIoU thresholds, while the AUPR and OSDR performances vary significantly as the tIoU threshold changes. Besides, as the tIoU threshold increasing, AUROC and OSDR values would increase accordingly. For all those tIoU thresholds and evaluation metrics, the proposed OpenTAL could consistently outperform baselines.

CHAPTER 7

COMPOSITIONAL ZERO-SHOT LEARNING

7.1 Introduction

Compositional visual recognition is a fundamental characteristic of human intelligence [164] but it is challenging for modern deep learning systems. For example, humans can easily recognize unseen `sliced tomatoes` after seeing `sliced potatoes` and `red tomatoes`. Such a compositional zero-shot learning (CZSL) capability is valuable in that, novel visual concepts from a huge combinatorial semantic space could be recognized without "seeing" any of their training data. For example, the C-GQA [234] dataset contains 413 states and 674 objects. This implies a total of at least 278K compositional classes in an open world while only 2% of them are accessible in training. Therefore, CZSL can significantly reduce the need for large-scale training data.

Traditional vision-based methods either directly learn the visual feature of compositions, or try to first decompose the visual data into representations of simple primitives, i.e., , states and objects, and then learn to re-compose the compositions [226, 5, 415, 119, 135, 321, 234, 396, 220, 174]. Thanks to the recent large pre-trained vision-language models (VLM) such as CLIP [261], state-of-the-art CZSL methods have been developed [239, 214, 352, 116]. For instance, CSP [239] inherits the hard prompt template of the CLIP, i.e., , *a photo of* `[state][object]` where only the embeddings of the states and objects are trained. The following methods [214, 352, 116] use soft prompt introduced in CoOp [409], where the embeddings of the prompt template are jointly optimized, leading to a better CZSL performance. The impressive performance of CLIP-based CZSL methods benefits from the sufficiently good feature alignment between the image and text modalities, and the prompting techniques for adapting the aligned features to recognizing compositional classes.

Despite the success of existing CLIP-based methods, we find several key considerations to

This chapter is adapted from the following publication:
"Wentao Bao, Lichang Chen, Heng Huang, and Yu Kong. Prompting language-informed distribution for compositional zero-shot learning. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2024."

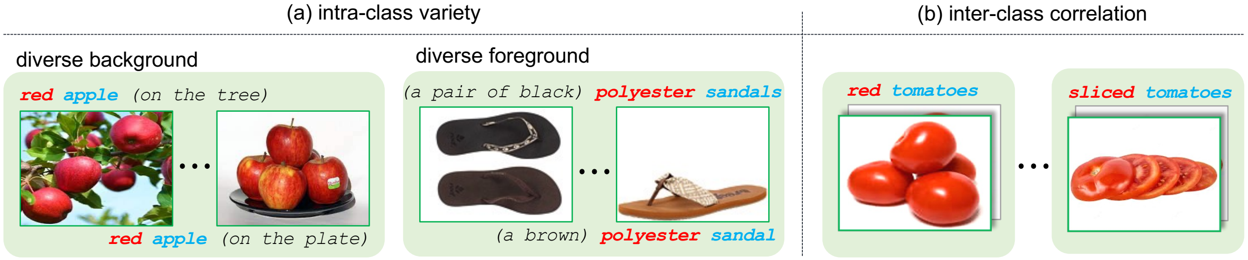


Figure 7.1 **Challenges of compositional recognition.** **(a)** images of the same compositional class appear differently due to diverse visual backgrounds or foregrounds. **(b)** `red tomatoes` and `sliced tomatoes` are visually correlated because 1) both are `tomatoes` object, and 2) the object `tomatoes` is inherently entangled with the state `red`, resulting in the need of primitive decomposition.

prompt the pre-trained CLIP for better CZSL modeling. First, the *diversity* and *informativeness* of prompts are both important to distinguish between compositional classes. CZSL can be treated as zero-shot learning on fine-grained categories, which requires a fine-grained context to prompt the CLIP model [261, 215]. However, to contextualize a class with fine granularity, the hard prompt in [261] suffers from the heuristic design of prompt templates, and a single prompt for each class lacks diversity to capture the intra-class variance of visual data (Fig. 7.1a). Though the ProDA [215] proposes to learn a collection of prompts that formulate class-specific distribution to address the diversity, the lack of *language informativeness* in their prompts limits their performance on fine-grained compositional categories. Second, the entanglement between visual primitives, e.g., `red` and `tomatoes` in Fig. 7.1b, incurs difficulty in learning decomposable visual representations that are useful for compositional generalization [209, 135], while such a capability is missing in [239, 352]. Though the more recent work [214, 116] learn to decompose the primitives and considers the re-composed compositional predictions, their language-only decomposition and probability-level mixup potentially limit the generalizability in the open-world.

In this paper, we propose a novel CLIP-based method for the CZSL task by prompting the language-informed distributions ($\mathbb{PLID}$) over both the compositional and primitive categories. To learn the diverse and informative textual class representations, the $\mathbb{PLID}$ leverages off-the-shelf large language models (LLM) to build the class-specific distributions and to enhance the class embeddings. Furthermore, we propose a visual language primitive decomposition (VLPD) module to decompose the image data into simple primitives for recognition of state and objects. Eventually, the compositional classification is performed by fusing the decisions from both the compositional and primitive spaces. The proposed PLID shows state-of-the-art performance on CZSL benchmarks such as MIT-States [121], UT-Zappos [374], and C-GQA [234].

Note that our method is orthogonal to the existing hard prompt [261], soft prompt tuning [409], and prompt distribution learning [215, 161, 208, 60]. We advocate prompting the distribution of informative LLM-based class descriptions. From a classification perspective, this is grounded on the classification-by-description [223, 222, 358, 107], that LLM-generated text enables more informative class representations. Compared to the deterministic soft or hard prompt aforementioned, our distribution modeling could capture the intra-class diversity and inter-class correlation for better zero-shot generalization. Compared to the existing prompt distribution learning approaches, the class context is more linguistically interpretable and provides fine-grained descriptive information about the class. Our method is also parameter-efficient without the need to optimize a large collection of prompts. Specific to the CZSL task, the enhanced class embeddings by LLM descriptions enable visual language primitive decomposition and decision fusion in both compositional and primitive space, which eventually benefits the generalization to the unseen.

In summary, the contributions are as follows. (*i*) We develop a PLID method that advocates prompting the language-informed distribution for compositional zero-shot learning, which is orthogonal to existing soft or hard prompting and distributional prompt learning. (*ii*) We propose primitive decomposition with stochastic logit mixup to fuse the classification decision from compositional and primitive predictions. (*iii*) We empirically show that PLID could achieve superior performance to prior arts in both the closed-world and open-world settings on MIT-States, UT-Zappos, and C-GQA datasets.

7.2 Related Work

Prompt Learning in VLM. Vision-Language Models (VLM) such as the CLIP [261] pre-trained on web-scale datasets recently gained substantial attention for their strong zero-shot recognition capability on various downstream tasks. Such a capability is typically achieved by performing prompt engineering to adapt pre-trained VLMs. Early prompting technique such as the hard prompt in CLIP uses the heuristic template "*a photo of* `[CLS]`" as the textual input. Recently, the soft prompt tuning method in CoOp [409], CoCoOp [408], and ResPT [266] that uses learnable embedding as the textual context of class names significantly improved the model adaptation performance. This technique is further utilized in MaPLe [138] that enables multi-modal prompt learning for both image and text. However, the prompts of these methods are deterministic and lack the diversity to capture the appearance variety in fine-grained visual data, so they are prone to overfitting the training data. To handle this issue, ProDA [215] explicitly introduces a collection of soft prompts to construct the class-specific Gaussian distribution, which results in better zero-shot performance and inspires the recent success of PPL [161] in the dense prediction task. Similarly, the PBPrompt [208] uses neural networks to predict the class-specific prompt distribution and utilizes optimal transport to align the stochastically sampled soft prompts and image patch tokens. The recent work [60] assumes the latent embedding of prompt input follows a Gaussian prior and adopts variational inference to learn the latent distribution. In this paper, in order to take the merits of the *informativeness* of hard prompt and the *diversity* of distributional modeling, we adopt the soft prompt to adapt the distributions supported by LLM-generated class descriptions.

Compositional Zero-Shot Learning (CZSL). For a long period, the CZSL task has been studied from a vision-based perspective in literature. They either directly learn the compositional visual features or disentangle the visual features into simple primitives, i.e., , states and objects. For example, [237, 185, 234] performs a direct classification by projecting the compositional visual features into a common feature space, and [213, 226, 5, 119, 415, 135, 209] decompose the visual feature into simple primitives so that the compositional recognition can be achieved by learning to recompose from the primitives. Though the recent large-scale pre-trained CLIP model shows impressive zero-shot capability, it is found to struggle to work well for compositional reasoning [217, 382, 169]. Thanks to the recent prompt learning [409], the CZSL task has been dominated by CLIP-based approaches [239, 214, 352, 116, 186, 404]. The common idea is to prompt the frozen CLIP model to separately learn the textual embeddings of simple primitives, which empirically show strong compositionality for zero-shot generalization. Different to [404, 186] that develop primitive adapters and [214, 352, 116] that use learnable prompts for deterministic vision-language alignment, our method takes the benefit of learnable prompt and LLM-generated text for distributional alignment, addressing the importance of diversity and informativeness for zero-shot generalization.

7.3 Preliminaries

CZSL Task Formulation. The CZSL task aims to recognize images of a compositional category $y \in \mathcal{C}$, where the semantic space $\mathcal{C}$ is a Cartesian product between the state space $\mathcal{S} = \{s_1, \ldots, s_{|\mathcal{S}|}\}$ and object space $\mathcal{O} = \{o_1, \ldots, o_{|\mathcal{O}|}\}$, i.e., , $\mathcal{C} = \mathcal{S} \times \mathcal{O}$. For example, as shown in Fig. 7.1, a model trained on images of `red apple` and `sliced tomatoes` needs to additionally recognize an image of `sliced apple`. In training, only a set of **seen** compositions is available. In closed-world testing, the model needs to recognize images from both the **seen** compositions in $\mathcal{C}^{(s)}$ and the **unseen** compositions in $\mathcal{C}^{(u)}$ that are assumed to be feasible, where the cardinality $|\mathcal{C}^{(s)} \cup \mathcal{C}^{(u)}| \ll |\mathcal{C}|$ since most of the compositions in $\mathcal{C}$ are practically not feasible. In open-world testing, the model needs to recognize images given any composition in $\mathcal{C}$.

VLMs for CZSL. Large pre-trained VLMs such as CLIP [261] have recently been utilized by CSP [239] for the CZSL task. The core idea of CSP is to represent the text embeddings of states in $\mathcal{S}$ and objects in $\mathcal{O}$ as learnable parameters and contextualize them with the hard prompt template "*a photo of* `[s][o]`" as the input of the CLIP text encoder, where `[s]` $\in \mathcal{S}$ and `[o]` $\in \mathcal{O}$. Given an image $\mathbf{x}$, by using the cosine similarity (`cos`) as the logit, the class probability of the composition y is defined as $p_{\boldsymbol{\theta}}(y|\mathbf{x}) = \texttt{softmax}(\texttt{cos}(\mathbf{v}, \mathbf{t}_y))$, where $\boldsymbol{\theta}$ are the $|\mathcal{S}| + |\mathcal{O}|$ learnable parameters, $\mathbf{v}$ and $\mathbf{t}_y$ are the image feature and class text embedding, respectively.

In training, the prediction $p_{\boldsymbol{\theta}}(\hat{y}|\mathbf{x})$ is supervised by multi-class cross-entropy loss. In CZSL testing, a test image is recognized by finding the compositional class $c \in \mathcal{C}$ which has the maximum $\texttt{cos}(\mathbf{v}, \mathbf{t}_c)$. The CSP method is simple, parameter-efficient, and it largely outperforms traditional approaches. However, due to the lack of diversity and informativeness in prompting, the zero-shot capability of CLIP is not fully exploited by CSP for the CZSL task.

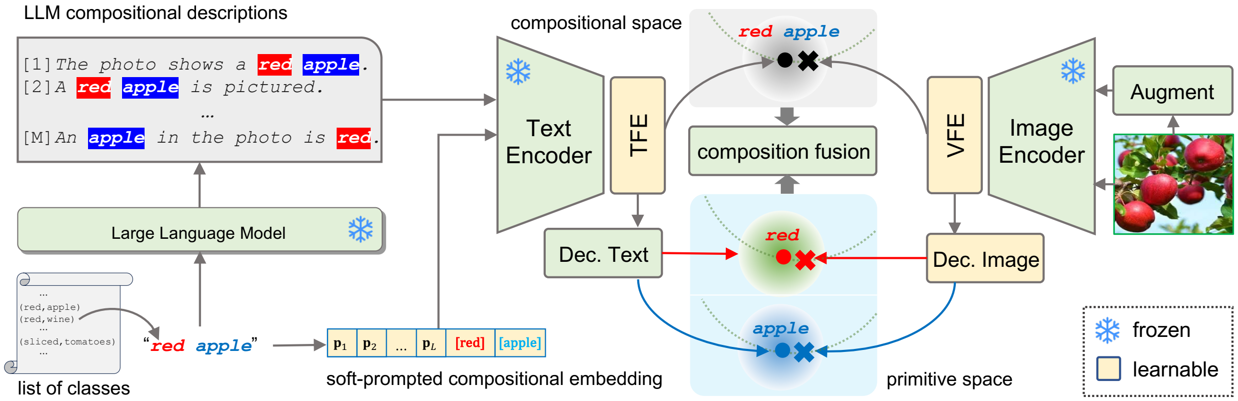


Figure 7.2 Overview of PLID. The model is developed for the CZSL task by aligning the semantics of image **x** (e.g., , image on the right) and compositional class $y = (s, o)$ (e.g., , "red apple") via a frozen CLIP [261]. It constructs language-informed text distributions in both compositional and primitive (attribute and object) spaces (middle part) by soft prompting and LLM-generated class descriptions (left part). The features of the image and text are enhanced by text and visual feature enhancement (TFE and VFE). Eventually, the compositional decisions from the two spaces are fused as the prediction.

7.4 Proposed Method

Overview. Fig. 7.2 shows an overview of the PLID. The basic idea is to use LLMs to generate sentence-level descriptions for each compositional class, and learn to prompt the class-wise text distributions (supported by the descriptions) to be aligned with image data. Besides, we introduce visual language primitive decomposition (VLPD) and stochastic logit mixup (SLM) to enable recognition at both compositional and primitive levels. In testing, an image is recognized by fusing the decisions from the directly predicted and the recomposed compositions.

7.4.1 Prompting Language-Informed Distribution

Motivation. To adapt the large pre-trained CLIP [261] to downstream tasks, recent distributional prompt learning [215, 161, 208, 60] shows the importance of *context diversity* by distribution modeling for strong generalization. Motivated by the inherent fine-granularity of compositional recognition in the CZSL task, we argue that not only the context diversity but also the *context informativeness* by language modeling, are both important factors to adapt CLIP to the zero-shot learning task. The insight behind this is that the sentence-level descriptions could contextualize compositional classes in a more fine-grained manner than the prior arts. Therefore, we propose to address the two factors by learning to **P**rompt the **L**anguage-**I**nformed **D**istributions (PLID) for the

CZSL task.

Compositional Class Description. To generate diverse and informative text descriptions for each compositional class, we adopt a similar way as [223] by prompting an LLM that shows instruction-following capability. An example below shows the format of the LLM instruction.

```
Keywords: sliced, potato, picture
Output: The picture features a beautifully arranged plate of thinly sliced
    potatoes.
```

For each composition $y = (s, o)$, we generate M descriptions denoted as $S^{(y)} = \{S_1^{(y)}, \dots, S_M^{(y)}\}$ where $S_m^{(y)}$ is a linguistically complete sentence. Different to [223] that aims to interpret the zero-shot recognition by attribute phrases from LLMs, we utilize the LLM-based sentence-level descriptions in the CZSL task for two benefits: (*i*) provide diverse and informative textual context for modeling the class distributions, and (*ii*) enhance the class embedding with fine-grained descriptive information.

Language-Informed Distribution (LID). For both the image and text modalities, we use the frozen CLIP model and learnable feature enhancement modules to represent the visual and language features, which are also adopted in existing CZSL literature [214, 116].

Specifically, for the text modality, each composition y is tokenized and embedded by CLIP embedding layer and further prompted by concatenating with learnable context vectors, i.e., , "$[\mathbf{p}_1] \dots [\mathbf{p}_L] [\mathbf{s}] [\mathbf{o}]$", where $\mathbf{p}_{1:L}$ is initialized by "`a photo of`" and shared with all classes. Followed by the frozen CLIP text encoder $\mathcal{E}_T$, the embedding of class y is $\mathbf{q}_y = \mathcal{E}_T\left([\mathbf{p}_1] \dots [\mathbf{p}_L] [\mathbf{s}] [\mathbf{o}]\right)$ where $\mathbf{q}_y \in \mathbb{R}^d$. Following the CZSL literature [352, 214], here the soft prompt $\mathbf{p}_{1:L}$ and primitive embeddings $[\mathbf{s}] [\mathbf{o}]$ are learnable while $\mathcal{E}_T$ is frozen in training.

To simultaneously address the lack of diversity and informativeness of the soft prompts, we propose to formulate the class-specific distributions supported by the texts $S^{(y)}$ and learn to prompt these distributions. Specifically, we encode $S^{(y)}$ by the frozen CLIP text encoder: $\mathbf{D}^{(y)} = \mathcal{E}_T(S^{(y)})$, where $\mathbf{D}^{(y)} \in \mathbb{R}^{M \times d}$. Then, we use $\mathbf{D}^{(y)}$ to enhance $\mathbf{q}_y$ by $\mathbf{t}_y = \Psi_{\text{TFE}}(\mathbf{q}_y, \mathbf{D}^{(y)})$ where Ψ_{TFE} is the text feature enhancement (**TFE**) implemented by a single-layer cross attention Transformer [328].

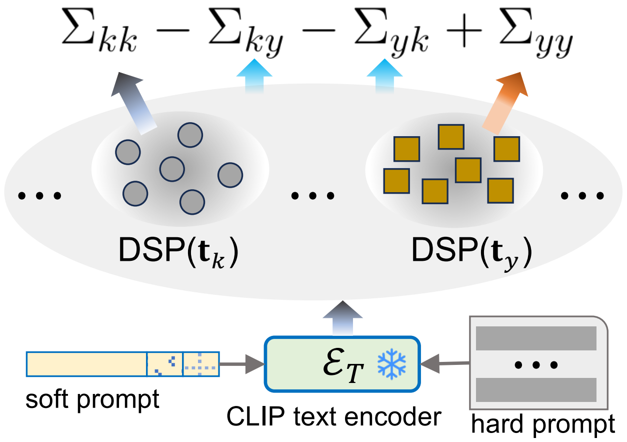


Figure 7.3 Prompting for intra- and inter-class covariance optimization.

Similarly, given an image $\mathbf{x}$, to mitigate the loss of fine-grained cues, we augment it with N views to be $\mathbf{X} = \{\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(N)}\}$. Followed by the frozen CLIP visual encoder $\mathcal{E}_V$, the feature of $\mathbf{x}$ is enhanced by $\mathbf{v} = \Psi_{\text{VFE}}(\mathcal{E}_V(\mathbf{x}), \mathcal{E}_V(\mathbf{X}))$ where Ψ_{VFE} is the visual feature enhancement (**VFE**) by cross attention [328], implemented with the same structure as TFE for simplicity.

We treat the enhanced text feature $\mathbf{t}_y$ of class y as the class mean and $\mathbf{t}_y + \mathbf{D}^{(y)}$ as the distribution support points (**DSP**) that follow the Gaussian $\mathcal{N}(\mathbf{t}_y, \mathbf{\Sigma}_y)$ where $\mathbf{\Sigma}_y$ is the text variance of the class y. The motivation of $\mathbf{t}_y + \mathbf{D}^{(y)}$ is to enable the flexibility of DSP to traverse around in the d dimensional space in training since $\mathbf{t}_y$ is trainable while $\mathbf{D}^{(y)}$ are pre-trained. For all $|C^{(s)}|$ (denoted as C) seen compositional classes, we build joint Gaussians $\mathcal{N}(\boldsymbol{\mu}_{1:C}, \mathbf{\Sigma}_{1:C})$ similar to ProDA [215], where the means $\boldsymbol{\mu}_{1:C} \in \mathbb{R}^{C \times d}$ are given by $\mathbf{t}_y$ over C classes, and the covariance $\mathbf{\Sigma}_{1:C} \in \mathbb{R}^{d \times C \times C}$ is defined across C classes for each feature dimension from DSP.

Discussions. Compared to the ProDA [215] that learns a collection of non-informative prompts, our DSPs are language-informed by $\mathbf{D}^{(y)}$ that provides more fine-grained descriptive information to help recognition and decomposition. Besides, our method is more parameter-efficient than ProDA since we only have a single soft prompt to learn. This is especially important for the CZSL task where there is a huge number of compositional classes. Lastly, we highlight the benefit of performing the intra- and inter-class covariance optimization induced by the learning objective of distribution modeling, which will be introduced below.

Learning Objective. Given the visual feature $\mathbf{v} \in \mathbb{R}^d$ of image $\mathbf{x}$ and the text embeddings $\mathbf{t}_{1:C}$

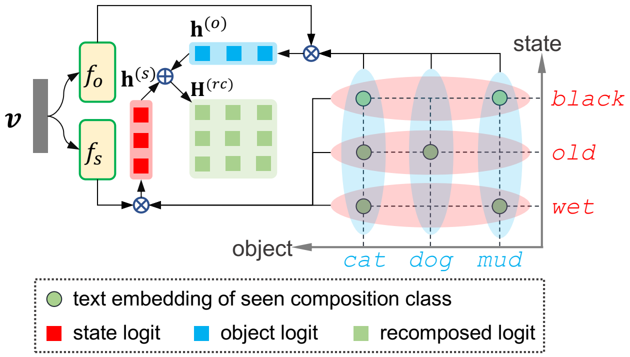


Figure 7.4 VLPD for recomposing.

from class-wise joint distributions $\mathcal{N}(\boldsymbol{\mu}_{1:C}, \boldsymbol{\Sigma}_{1:C})$, minimizing the cross-entropy loss is equivalent to minimizing the upper bound of negative log-likelihood (NLL):

$$-\log \mathbb{E}_{\mathbf{t}_{1:C}} p(y|\mathbf{v}, \mathbf{t}_{1:C}) \leq -\log \frac{\exp(h_y/\tau)}{\sum_{k=1}^{C} \exp((h_k + h_{k,y}^{(m)})/\tau)} := \mathcal{L}_y(\mathbf{x}, y), \tag{7.1}$$

where the compositional logit $h_y = \texttt{cos}(\mathbf{v}, \mathbf{t}_y)$, the pairwise margin $h_{k,y}^{(m)} = \mathbf{v}^\top \mathbf{A}_{k,y} \mathbf{v}/(2\tau)$ and $\mathbf{A} \in \mathbb{R}^{d \times C \times C}$ is given by $\mathbf{A}_{k,y} = \boldsymbol{\Sigma}_{kk} + \boldsymbol{\Sigma}_{yy} - \boldsymbol{\Sigma}_{ky} - \boldsymbol{\Sigma}_{yk}$. The covariance $\mathbf{A}_{k,y}$ indicates the correlation between the k-th out of C classes and the target class y on each of d feature dimensions. The insight of minimizing $\mathcal{L}_y(\mathbf{x}, y)$ is illustrated in Fig. 7.3, which encourages minimizing intra-class variance by $\boldsymbol{\Sigma}_{yy}$ and $\boldsymbol{\Sigma}_{kk}$, and maximizing inter-class separability indicated by $\boldsymbol{\Sigma}_{ky}$ and $\boldsymbol{\Sigma}_{yk}$.

7.4.2 Primitives Decomposition for Fused Recognition

Motivation. Considering the fundamental challenge in the CZSL task, that the visual primitives are inherently entangled in an image, an unseen composition in testing can be hardly identified if its object (or its state) embedding is overfitted to the visual data of seen compositions. To this end, it is better to inherit the benefits of the decompose-recompose paradigm [415, 135, 209] by decomposing visual features into simple primitives, i.e., , states and objects, from which the recomposed decision can be leveraged for zero-shot recognition. Thanks to the compositionality of CLIP [342, 325], such motivation can be achieved by the visual-language primitive decomposition (**VLPD**). See Fig. 7.4 and we explain it below. Based on VLPD, we propose the stochastic logit mixup to fuse the directly learned compositions and the recomposed ones.

VLPD. Specifically, we use two parallel neural networks f_s and f_o to decompose $\mathbf{v}$ into the state visual feature $f_s(\mathbf{v})$ and object visual feature $f_o(\mathbf{v})$, respectively. To get the primitive-level supervisions, given the training compositions $C^{(s)}$ (see the circle dots in Fig. 7.4), we group their enhanced embeddings $\{\mathbf{t}_y\}$ over the subset $\mathcal{Y}_o$, in which all compositions share the same given object o (see vertical ellipses in Fig. 7.4), and group $\{\mathbf{t}_y\}$ over the subset $\mathcal{Y}_s$, in which all compositions share the same given state s (see horizontal ellipses in Fig. 7.4). Thus, given a state s and an object o, the predicted object logit h_s and state logit h_o are computed by

$$h_s = \cos\left(f_s(\mathbf{v}), \frac{1}{|\mathcal{Y}_s|}\sum_{y\in\mathcal{Y}_s}\mathbf{t}_y\right), \quad h_o = \cos\left(f_o(\mathbf{v}), \frac{1}{|\mathcal{Y}_o|}\sum_{y\in\mathcal{Y}_o}\mathbf{t}_y\right). \tag{7.2}$$

Different from DFSP [214] that only decomposes text features, we additionally use f_s and f_o to decompose visual features $\mathbf{v}$ and empirically show the superiority of performing both visual and language decomposition (see Table 7.6).

Following the spirit of distribution modeling, we also introduce the distributions over state and object categories, where the corresponding DSP, denoted as $\mathbf{D}^{(s)}$ and $\mathbf{D}^{(o)}$, are obtained by grouping $\mathbf{D}^{(y)}$ over $\mathcal{Y}_s$ and $\mathcal{Y}_o$, respectively. This leads to the following upper-bounded cross-entropy losses:

$$\begin{aligned}\mathcal{L}_s(x, s) &= -\log\frac{\exp(h_s/\tau)}{\sum_{k=1}^{|S|}\exp((h_k + h_{k,s}^{(m)})/\tau)},\\ \mathcal{L}_o(x, o) &= -\log\frac{\exp(h_o/\tau)}{\sum_{k=1}^{|O|}\exp((h_k + h_{k,o}^{(m)})/\tau)},\end{aligned} \tag{7.3}$$

where $h_{k,s}^{(m)}$ and $h_{k,o}^{(m)}$ are determined the same way as $h_{k,y}^{(m)}$ in Eq. (7.1). By this way, the merits of language-informed distribution modeling, i.e., , the inter- and intra-class covariance optimization constraints, can be introduced into primitive space for fused recognition as introduced below.

Composition Fusion. With the individually supervised f_s and f_o, we have $p(y|\mathbf{v}) = p(s|\mathbf{v}) \cdot p(o|\mathbf{v})$ according to conditional independence, that induces $p(y|\mathbf{v}) \propto \exp((h_s + h_o)/\tau)$. Therefore, the recomposed logit matrix $\mathbf{H}^{(rc)} \in \mathbb{R}^{|S|\times|O|}$ is a Cartesian (element-wise combinatorial) sum between $\mathbf{h}^{(s)} \in \mathbb{R}^{|S|}$ and $\mathbf{h}^{(o)} \in \mathbb{R}^{|O|}$, i.e., , $\mathbf{H}^{(rc)} = \mathbf{h}^{(s)} \oplus \mathbf{h}^{(o)\top}$, where $\mathbf{h}^{(s)}$ contains all state logits and $\mathbf{h}^{(o)}$ contains all object logits. See the red and blue squares in Fig. 7.4.

Given the recomposed logit $h_y^{(rc)} \in \mathbf{H}^{(rc)}$ and the directly learned compositional logit h_y by Eq. 7.1, we propose to stochastic fusion method in training by sampling a coefficient λ from a Beta prior distribution:

$$\tilde{h}_y = (1 - \lambda)h_y + \lambda h_y^{(rc)}, \quad \lambda \sim \text{Beta}(a, b), \tag{7.4}$$

where (a, b) are hyperparameters indicating the prior preference for each decision. In training, we replace the h_y and h_k of Eq. (7.1) with the mixed logit $\tilde{h}_y$ and $\tilde{h}_k$, respectively. In testing, no stochasticity is needed so that we use the Beta expectation of λ which is $a/(a + b)$ to get the fused logit $\tilde{h}_y$.

The insights behind the stochasticity are that the Beta distribution indicates a prior to h_y or $h_y^{(rc)}$. It provides the flexibility of which compositional decision to trust in, and the stochasticity of the coefficient λ inherently introduces a regularization effect in training [26]. Moreover, compared to softmax probability mixup [116], our logit mixup avoids the limitation of softmax normalization over a huge number of compositional classes, that rich information of class relationship is lost after softmax normalization according to [8]. Such class relationships are even more important in the CZSL problem as indicated in [234].

7.5 Experiments

Datasets. We perform experiments on three CZSL datasets, i.e., , MIT-States [121], UT-Zappos [374], and C-GQA [234], following the standard splitting protocols in CZSL literature [259, 239, 214]. MIT-States consists of 115 states and 245 objects, with 53,753 images in total. Following [259, 239, 214], it is split into 1,262 seen and 300/400 unseen compositions for training and validation/testing, respectively. UT-Zappos contains 16 states and 12 objects for 50,025 images in total, and it is split into 83 seen and 15/18 unseen compositions for training and validation/testing. C-GQA contains 453 states and 870 objects for 39,298 images, and it is split into 5,592 seen and 1,040/923 unseen compositions for training and validation/testing, respectively, resulting in 7,555 and 278,362 target compositions in closed- and open-world settings.

Evaluation. We report the metrics in both closed-world (**CW**) and open-world (**OW**) settings, including the best seen accuracy (**S**), the best unseen accuracy (**U**), the best harmonic mean (**H**) between the seen and unseen accuracy, and the area under the curve (**AUC**) of unseen versus seen accuracy. For OW evaluation, following the CSP [239], we adopt the feasibility calibration by

Table 7.1 CZSL results of Closed- and Open-World settings on three datasets. Baseline results are from published literature except for ProDA. Note that "–" indicates no results reported by the PCVL paper or not applicable by ProDA for more than 278K compositional classes on the C-GQA dataset.

	Method	*MIT-States*				*UT-Zappos*				*C-GQA*			
		S	U	H	AUC	S	U	H	AUC	S	U	H	AUC
Closed	CLIP [261]	30.2	46.0	26.1	11.0	15.8	49.1	15.6	5.0	7.5	25.0	8.6	1.4
	CoOp [409]	34.4	47.6	29.8	13.5	52.1	49.3	34.6	18.8	20.5	26.8	17.1	4.4
	ProDA[1] [215]	37.4	51.7	32.7	16.1	63.7	60.7	47.6	32.7	–	–	–	–
	CSP [239]	46.6	49.9	36.3	19.4	64.2	66.2	46.6	33.0	28.8	26.8	20.5	6.2
	PCVL [352]	48.5	47.2	35.3	18.3	64.4	64.0	46.1	32.2	–	–	–	–
	HPL [330]	47.5	50.6	37.3	20.2	63.0	68.8	48.2	35.0	30.8	28.4	22.4	7.2
	DFSP [214]	46.9	52.0	37.3	20.6	66.7	**71.7**	47.2	36.0	38.2	32.0	27.1	10.5
	PLID	**49.7**	**52.4**	**39.0**	**22.1**	**67.3**	68.8	**52.4**	**38.7**	**38.8**	**33.0**	**27.9**	**11.0**
Open	CLIP [261]	30.1	14.3	12.8	3.0	15.7	20.6	11.2	2.2	7.5	4.6	4.0	0.3
	CoOp [409]	34.6	9.3	12.3	2.8	52.1	31.5	28.9	13.2	21.0	4.6	5.5	0.7
	ProDA[1] [215]	37.5	18.3	17.3	5.1	63.9	34.6	34.3	18.4	–	–	–	–
	CSP [239]	46.3	15.7	17.4	5.7	64.1	44.1	38.9	22.7	28.7	5.2	6.9	1.2
	PCVL [352]	48.5	16.0	17.7	6.1	64.6	44.0	37.1	21.6	–	–	–	–
	HPL [330]	46.4	**18.9**	19.8	6.9	63.4	48.1	40.2	24.6	30.1	5.8	7.5	1.4
	DFSP [214]	47.5	18.5	19.3	6.8	66.8	**60.0**	44.0	30.3	38.3	7.2	10.4	2.4
	PLID	**49.1**	18.7	**20.4**	**7.3**	**67.6**	55.5	**46.6**	**30.8**	**39.1**	**7.5**	**10.6**	**2.5**

GloVe [256] to filter out infeasible compositions.

Implementation Details. We implement the PLID based on the CSP codebase in PyTorch. The CLIP architecture ViT-L/14 is used by default. On the MIT-States, we generate $M = 64$ texts and augment an image with $N = 8$ views, and adopt $\text{Beta}(1, 9)$ as prior. The dropout rates of cross-attention layers in TFE and VFE are set at 0.5, and the dropout rate of 0.3 for the learnable state and object embeddings. For the soft prompt embeddings, we set the context length of text encoder to 8 for all datasets. Following [214], we use Adam optimizer with base learning rate 5e-5 and weight decay 2e-5, and step-wise decay it with the factor of 0.5 every 5 training epochs for a total of 20 epochs.

Table 7.2 **Ablation study**. (a): the baseline that uses mean pooling of text embeddings from T5-generated sentences. (b): add language-informed distribution (LID). (c): use text and visual feature enhancement module (FE). (d): change the LLM from T5-base to the OPT-1.3B. (e): apply primitive decomposition for fused decisions (PDF).

	LID	FE	OPT	PDF	H_{cw}	AUC_{cw}	H_{ow}	AUC_{ow}
(a)					35.41	18.56	17.37	5.56
(b)	✓				37.06	20.43	18.65	6.50
(c)	✓	✓			37.87	21.09	19.70	6.95
(d)	✓	✓	✓		38.80	21.67	19.61	7.01
(e)	✓	✓	✓	✓	**38.97**	**22.12**	**20.41**	**7.34**

7.5.1 Main Results

The results are reported in Table 7.1. We compare with the CZSL baselines that are developed on the same frozen CLIP model. The table shows that under both the closed-world and open-world test settings, our proposed $\mathbb{PLID}$ method achieves the best performance in most metrics on the three datasets. Note that ProDA [215] also formulates the class-wise Gaussian distributions to address the intra-class diversity, but it can only outperform CLIP and CoOp on all metrics. This indicates the importance of both diversity and informativeness for the CZSL task. On the UT-Zappos dataset, the $\mathbb{PLID}$ outperforms the DFSP in terms of S, H, and AUC by 0.6%, 5.2%, and 2.7% respectively, while inferior to the DFSP on the best unseen metric. The potential reason is that DFSP fuses the text features into the image images, which better preserves the generalizability of CLIP for the small downstream UT-Zappos dataset. Note that the HPL method uses prompt learning and recognition at both compositional and primitive levels, but it performs only slightly better than CSP and way worse than our method, indicating that traditional prompt learning helps but is not enough to adapt the CLIP model to the CZSL task.

7.5.2 Model Analysis

To comprehensively analyze the proposed $\mathbb{PLID}$, we perform extensive ablation study and design analysis on the middle-sized MIT-States dataset in this section.

Major Components. In Table 7.2, we show the contribution of the major components in the $\mathbb{PLID}$ model. It is clear that they are all beneficial. We highlight some important observations: (1) The LID method in row (b) significantly improves the performance compared to the baseline (a) that

Table 7.3 Effect of LID on states ($\mathcal{N}_s$), objects ($\mathcal{N}_o$), and compositions ($\mathcal{N}_y$). The first row: the model w/o LID.

$\mathcal{N}_s$	$\mathcal{N}_o$	$\mathcal{N}_y$	H_{cw}	AUC_{cw}	H_{ow}	AUC_{ow}
			38.44	21.67	19.53	6.99
✓	✓		38.30	21.62	19.49	6.95
		✓	38.49	21.90	19.93	7.20
✓	✓	✓	**38.97**	**22.12**	**20.41**	**7.34**

Table 7.4 Effect of LLMs. Note that GPT-3.5 is not open-sourced so that we use its API call to get text descriptions.

LLMs	H_{cw}	AUC_{cw}	H_{ow}	AUC_{ow}
Mistral-7B	37.22	20.78	19.22	6.74
GPT-3.5	37.38	20.61	19.38	6.80
T5-base	38.41	21.53	**20.46**	7.34
OPT-1.3B	**38.97**	**22.12**	20.41	**7.34**

does not formulate Gaussian distribution in training, and they are much better than ProDA (20.43% vs 16.1% of AUC_{cw}) when referring to Table 7.1. This implies that addressing the context **diversity** by modeling the Gaussian distribution like the ProDA is not sufficient, but context **informativeness** is critical and preferred for the CZSL task. (2) Rows (c)(d) show that feature enhancement (FE) and the better LLM OPT-1.3B can also brings performance gains. (3) Rows (e) show that the primitive decomposition for fused decision (PDF) could further improve the CZSL performance in both closed- and open-world settings. In the following paragraphs, we further validate the effect or design choices of these components in detail.

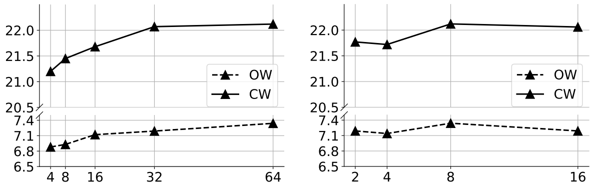
Effect of LID. In Table 7.3, we investigate at which semantic level the language-informed distribution (LID) should be applied. Denote the Gaussian distribution on state, object, and composition as $\mathcal{N}_s$, $\mathcal{N}_o$, and $\mathcal{N}_y$, respectively. The Table 7.3 results clearly show the superiority of applying LID on all three semantic levels. This indicates the generality of LID towards many potential zero-shot or open-vocabulary recognition problems.

Effect of LLM. In Table 7.4, we analyze the choice of LLMs by comparing $\mathbb{PLID}$ variants using different LLMs, including the T5-base [262], OPT-1.3B [395], GPT-3.5 [247], and Mistral-7B [126]. It shows the performance varies across different LLMs. Note that the capacity of GPT-3.5 and Mistral-7B on general language processing tasks is much better than T5-base and OPT-1.3B. However, we do not see improvements by using these generally larger and better LLMs, but a small OPT-1.3B is sufficient to achieve the best performance.

TFE and VFE. In Table 7.5, we explore the design choices of the text and visual feature enhancement (TFE and VFE) modules. The results show that using one layer of randomly initialized

Table 7.5 Design choices of feature enhancement (FE). We explore the use of text or visual feature enhancement (TFE/VFE) and the number of their cross-attention layers.

TFE	VFE	layers	$\mathrm{H_{cw}}$	$\mathrm{AUC_{cw}}$	$\mathrm{H_{ow}}$	$\mathrm{AUC_{ow}}$
✓		1	37.89	21.07	19.37	6.78
	✓	1	37.48	21.04	19.43	6.72
✓	✓	3	37.46	20.65	19.15	6.70
✓	✓	1	**38.97**	**22.12**	**20.41**	**7.34**



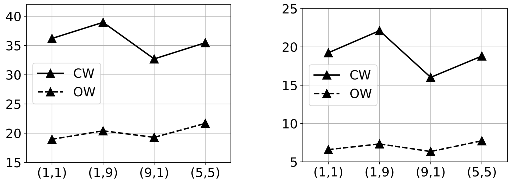
(a) AUC vs. M

(b) AUC vs. N

Figure 7.5 Impact of M and N. We set $N = 8$ for the Fig. 7.5a, while we set $M = 64$ for the Fig. 7.5b.

Table 7.6 Effect of VLPD and fusion strategies. We explore the modalities (text or image) of the decomposition, and whether deterministic (det.) or stochastic (stoc.) compositional fusion.

VLPD		fusion		$\mathrm{H_{cw}}$	$\mathrm{AUC_{cw}}$	$\mathrm{H_{ow}}$	$\mathrm{AUC_{ow}}$
text	image	det.	stoc.				
			✓	37.94	20.98	19.67	6.98
✓			✓	38.40	21.31	19.99	7.13
✓	✓			38.42	21.69	20.24	7.31
✓	✓	✓		38.67	21.90	19.99	7.15
✓	✓		✓	**38.97**	**22.12**	**20.41**	**7.34**



(a) best HM

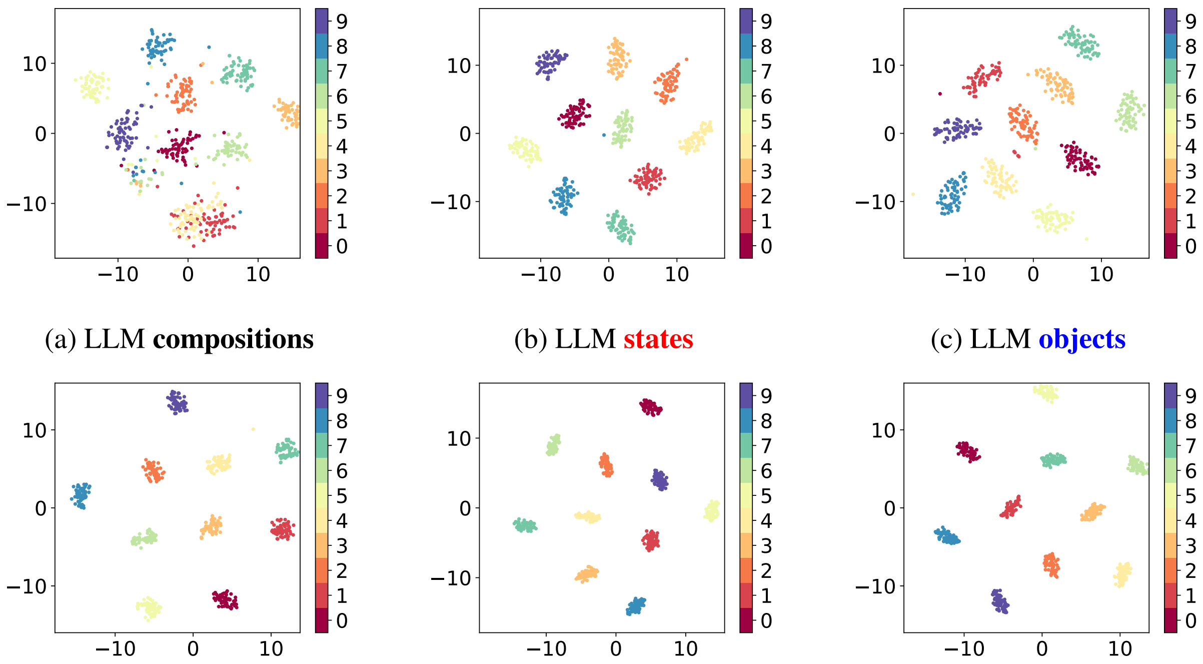
(b) best AUC

Figure 7.6 Impact of (a, b). Here $(1, 1)$ implies random sampling while $(5, 5)$ implies equally trusted.

cross-attention for both TFE and VFE performs the best. Using more cross-attention layers will causes significant performance drop (see the 3rd row). We attribute the cause to the overfitting issue when more learnable parameters introduced.

VLPD and Fusion. In Table 7.6, we validate the design choices of visual language primitive decomposition (VLPD) and the stochastic compositional fusion. Comparing with the results of the first two rows, it show clear advantages of primitive decomposition over both image and text modalities. Note that DFSP [214] also has primitive decomposition but only on text modality. Our better performance than DFSP and the results in Table 7.6 thus tell the need for decomposition on both visual and image. Besides, to validate our stochastic compositional fusion, we compare with the model without fusion in the 3rd row and the model with only deterministic fusion (weighted average without Beta sampling) in the 4th row. They also show the benefit of fusion with stochasticity.

Hyperparameters. In Fig. 7.5, we show the impact of the number of generated text descriptions



(a) LLM **compositions**

(b) LLM **states**

(c) LLM **objects**

(d) Learned **composition** DSP

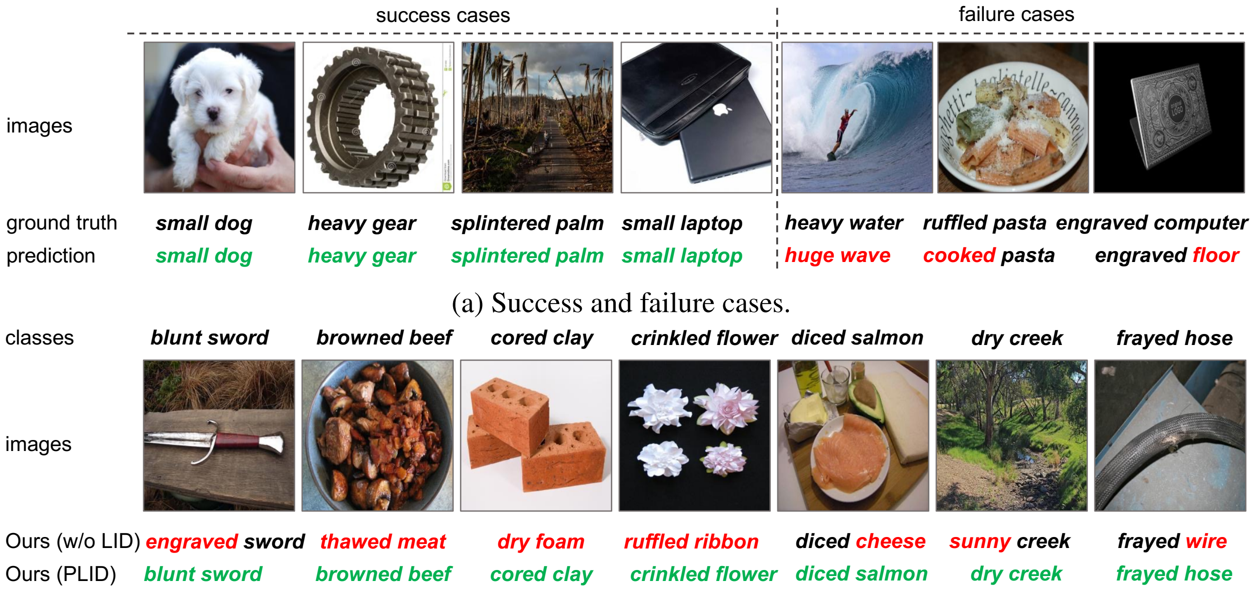
(e) Learned **state** DSP

(f) Learned **object** DSP

Figure 7.7 tSNE visualization of the text embeddings with (the 2nd row) and without (the 1st row) learnable distribution modeling over compositions (the 1st column), states (the 2nd column), and objects (the 3rd column). This figure clearly shows that our method achieves good performance by distribution modeling.

M and the number of augmented image views N. It shows that the best performance is achieved when $M = 64$ and $N = 8$. We note that more augmented image views slightly decrease the performance, which could be attributed to the overfitting of the seen compositions. In Fig. 7.6, we show the impact of the Beta prior parameters (a, b). We set them to $(1, 1)$ for random sampling, $(1, 9)$ for preference to the composition, $(9, 1)$ for preference to re-composition, and $(5, 5)$ for equal preference, respectively. It reveals that trusting more of the directly learned composition by $\text{Beta}(1, 9)$ achieves the best results.

Class Distributions. We use the tSNE to visualize the generated text embeddings $\mathbf{D}$ and the learned DSP from or $\mathbb{PLID}$ model in Fig. 7.7, where the same set of 10 compositional (or state/object) classes are randomly selected from MIT-States dataset. It shows that by learning the distribution of each composition, state, and object from LLM-generated texts using Eq. (7.1) and (7.3) and TFE module, class embeddings can be distributed more compactly in each class (small



(a) Success and failure cases.

(b) Comparison with model without LID.

Figure 7.8 Case studies. In Fig. 7.8a, we show the cases from the MIT-States dataset that our method succeeds or fails. In Fig. 7.8b, we compare the proposed method with and without language-informed distribution (LID) modeling. Correct predictions are in **green** color, while incorrect predictions on state or object part are marked in **red**.

intra-class variance), and better separated among multiple classes (large inter-class distance). This clearly show why our proposed language-informed distribution modeling works in the CZSL task.

Case Study. In Fig. 7.8a, we show some success and failure cases of our PLID model. They reveal PLID still could make mistakes on the state prediction (`cooked pasta`) and object prediction (`engraved floor`), which indicates there are still rooms for improvement. In Fig. 7.8b, we show that PLID could work much better than the model without LID. For example, the `sunny creek` and `frayed wire` are incorrect potentially due to the lack of handling (*i*) intra-class variety, as the `dry creek` images can be `sunny` and the `frayed hose` class could contain `wire` images, and (*ii*) inter-class correlation, as the `sunny` is correlated to both the `dry creek` images and other `sunny` images.

7.6 Conclusion

In this work, we propose a novel CLIP-based compositional zero-shot learning (CZSL) method named PLID. It leverages the generated text description of each class from large language models to formulate the class-specific Gaussian distributions. By softly prompting these language-informed distributions, PLID could achieve diversified and informative class embeddings for fine-grained compositional classes. Besides, we decompose the visual embeddings of image data into states and objects, from which the re-composed predictions are derived to calibrate the prediction by our proposed stochastic logit mixup strategy. Experimental results show the superiority of the PLID on multiple CZSL datasets.

7.7 Supplementary Material

7.7.1 Broader Impact and Limitations

Broader Impact. The method in this work can be broadly extended to more multi-modality applications, such as general zero-shot learning, cross-modality compositional retrieval and generation, etc. Besides, the central idea of LLM-based modality alignment is not limited to text and image, but any modality that could reveal the semantic categories in practice is promising to explore in the future. The potential negative societal impact is that, the developers should be cautious by carefully examining the societal biases indicated by the generated class descriptions, though the LLMs we used are publicly accessible.

Limitations. One limitation is that the primitive decomposition could be difficult to learn when the states are non-visual concepts like `smelly`, `hot`, etc., even by the pre-trained CLIP model. Another limitation is that the generated descriptions by LLMs are not grounded to the image such that some distraction from generated descriptions could be introduced.

7.7.2 Generating Compositional Class Descriptions

In this work, we choose T5-base, OPT-1.3B, GPT-3.5, and Mistral-7B models as the LLMs for compositional class description generation. For the T5 model, we follow the same setting as [107] that uses the T5-base model for word-to-sentence generation. The T5-base model was pre-trained on the Colossal Clean Crawled Corpus dataset [262] and finetuned on the CommonGen dataset [190]. Take the `painted ceiling` as an example, the results from T5-base model are:

```
- A very old but beautifully decorated ceiling.
- A remodeled interior with a painted ceiling.
- A painted ceiling at a restaurant.
- Stained glass windows and a carved pattern on the ceiling.
- Painted ceilings and a fireplace.
- This apartment has a painted ceiling.
- A painted ceiling was a huge hit.
- A chandelier is painted in the middle of the ceiling.
- A stained glass window in a bathroom with a white painted ceiling.
- The ceiling of a hotel is painted.
```

For the OPT model, we adopt the pre-trained OPT-1.3B model which is small but still could generate sentences of sufficiently good quality. Considering that it is a decoder-only LLM that relies on a few examples as the prompt for word-to-sentence generation, we design the prompt template with 4 examples as shown below.

```
Keywords: sliced, potato, picture
Output: The picture features a beautifully arranged plate of thinly sliced
    potatoes.
###
Keywords: red, apple, photo
Output: In the photo, a bright red apple is the central focus, captured in
    stunning detail.
###
Keywords: leather, shoes, image
Output: The image showcases a sleek pair of leather shoes, meticulously
    designed to impress.
###
Keywords: broken, car, photo
Output: The photo captures the aftermath of a car accident with a broken
    vehicle lying on the road.
###
```

where the `Keywords` is followed by the words of the state, object, and the word randomly picked from the set {`photo`, `image`, `picture`}. Using the same example *painted ceiling* as T5-base model, the generated sentences are:

```
- The painting of the ceiling features an intricate pattern of intricate
   gold-colored paint and is framed by a white background.
- The ceiling has been painted with the pictures of these three characters,
   all arranged together.
- In the picture, the ceiling is covered in bright, colorful paintwork that
   has been painted on by hands painted white. The colors have been
   selected carefully.
- In the picture, the ceiling features painted decoration. The decoration
   resembles the surface of the sea, and has been painted in shades of blue.
- The photograph captures both the bright colors of the painting atop the
   ceiling and the subtle shades of light reflecting off of it.
- The large picture shows a large pattern painted onto the ceiling. The
   blue line shows paint dripping down.
- The wall behind the picture shows three different painted ceilings, in
   bright contrasting colors. A vibrant sky and blue skies are depicted
   against the dark brick wall.
- The ceiling of the room depicted in the painting could very well be
   painted in a few hours. The details of each object are clearly defined
   in its placement and position.
- Another photo of the same scene, this time featuring a ceiling painted in
   a stunning, white color.
- A painted ceiling is shown, painted according to a specific design. this
   is a typical design that can also include decorative or functional
   elements.
```

It is clear that the generated class descriptions are much more diverse and informative than

Table 7.7 Effect of covariance sharing on MIT-States dataset. All methods use the same batch size of 64 for a fair comparison of GPU memory.

Variants	Mem.(GB)	H_{cw}	AUC_{cw}	H_{ow}	AUC_{ow}
ProDA [215]	32.5	32.71	16.11	17.30	5.11
PLID (w. ShareCov)	**17.6**	38.50 (-0.47%)	21.69 (-0.43%)	19.81 (-0.60%)	7.04 (-0.30%)
$\mathbb{PLID}$ (full)	22.2	**38.97**	**22.12**	**20.41**	**7.34**

those of the OPT model.

7.7.3 Covariance Sharing

For the CZSL task, the spatial complexity of computing the covariance matrix $\mathbf{\Sigma}_{1:C}$ is $O(|C^{(s)}|^2 d)$ which could be too heavy to compute if the number of the compositions is too large. For example, the C-GQA dataset contains 278K seen compositions which result in around 6×10^{13} floating elements of $\mathbf{\Sigma}_{1:C}$ for 768-dim text features. To handle this issue, we instead implement the $\mathbf{\Sigma}_{1:C}$ by sharing the covariance across attributes given the same object. This implies that the model is encouraged to learn the object-level distributions.

Specifically, similar to the VLPD module of the main paper, we compute the mean $\boldsymbol{\mu}_{1:|O|}$ and covariance $\mathbf{\Sigma}_{1:|O|}$ over the objects by grouping $\mathbf{t}_y$ and $\mathbf{D}^{(y)}$ with object labels:

$$\mathbf{t}_o = \frac{1}{|\mathcal{Y}_o|} \sum_{y \in \mathcal{Y}_o} \mathbf{t}_y, \quad \mathbf{D}^{(o)} = \frac{1}{|\mathcal{Y}_o|} \sum_{y \in \mathcal{Y}_o} \mathbf{D}^{(y)}, \tag{7.5}$$

where $\mathcal{Y}_o$ is the subset of compositions in $\mathcal{Y}$ that contains the same object as y. Then, all the pairwise margins $\mathbf{H}_o^{(m)} \in \mathbb{R}^{|O| \times |O|}$ in object space can be mapped back to $\mathbf{H}^{(m)} \in \mathbb{R}^{C \times C}$ in a compositional space by sharing it with all compositions in $\mathcal{Y}_o$. This could significantly reduce the computation load of the covariance while compromising the accuracy of distribution modeling.

Since the distribution modeling for both our $\mathbb{PLID}$ and ProDA is not applicable to the C-GQA dataset, we use the MIT States dataset to show the negative impact of sharing the covariance (see Table 7.7). It shows that the covariance sharing can significantly save the GPU memory (17.6 vs 32.5 GB), while still performing much better than ProDA.

Table 7.8 Hyperparameters of model implementation.

Hyperparameters	MiT-States	UT-Zappos	C-GQA
max epochs	20	25	20
base learning rate	0.00005	0.0001	0.00001
weight decay	0.00002	0.00001	0.00001
number of text descriptions	64	32	64
number of image views	8	8	8
attention dropout	0.5	0.1	0.1
weights of primitive loss	0.1	0.01	0.01

7.7.4 Primitive-level Gaussian Modeling

To formulate the Gaussian distributions over the state classes and the object classes, we group the text embeddings of composition descriptions $\mathbf{D}$ by Eq. (7.5), resulting in the distribution support points (DSP) $\mathbf{t}_o + \mathbf{D}^{(o)}$ and $\mathbf{t}_s + \mathbf{D}^{(s)}$ for a given object class o and state class s, respectively. The DSPs are assumed to follow the state distribution $\mathcal{N}(\mathbf{t}_s, \mathbf{\Sigma}_s)$ or the object distribution $\mathcal{N}(\mathbf{t}_o, \mathbf{\Sigma}_o)$, where the covariances $\mathbf{\Sigma}_s$ and $\mathbf{\Sigma}_o$ are determined by $\mathbf{D}^{(s)}$ and $\mathbf{D}^{(o)}$, respectively.

Eventually, given the decomposed state visual features $f_s(\mathbf{v})$ and object visual features $f_o(\mathbf{v})$, the logit margin terms are defined as

$$h_{k,s}^{(m)} = f_s(\mathbf{v})^\top \mathbf{A}_{k,s} f_s(\mathbf{v}), \quad \text{and} \quad h_{k,o}^{(m)} = f_o(\mathbf{v})^\top \mathbf{A}_{k,o} f_o(\mathbf{v}), \tag{7.6}$$

where the index k ranges within $[1, |\mathcal{S}|]$ for computing the state classification loss $\mathcal{L}_s$, and ranges within $[1, |\mathcal{O}|]$ for computing the object classification loss $\mathcal{L}_o$, respectively.

7.7.5 More Implementation Details and Results

Implementation. The training hyperparameters of our final model on each dataset are listed in Table 7.8.

More Ablation Analysis. In Table 7.9, we show more ablation study results on the design choices of our model. The first is to answer: *Should we learn both the compositional and primitive feature space?* This is interesting because if the primitive space can be learned by the proposed VLPD, intuitively the original compositional space is redundant. In the first line of Table 7.9, we show that if we remove the compositional space but only learn primitive space to recompose,

Table 7.9 More ablation study results.

model variants		H_{cw}	AUC_{cw}	H_{ow}	AUC_{ow}
recompose only		30.02	13.88	15.46	4.35
w/o soft prompt		38.57	21.67	20.00	7.17
3-layers FE	TFE only	36.89	19.93	18.77	6.42
	VFE only	36.55	19.80	19.06	6.51
	TFE+VFE	37.46	20.65	19.15	6.70
full model		38.97	22.12	20.41	7.34

the performance experiences a large drop in all metrics. This can be explained by the intuition that, without a direct compositional recognition, the merits of *explicitly* learned separatability and *implicitly* learned compositionality will be totally lost. These are the keys to the success of the pioneering CZSL method CSP [239].

Besides, in Table 7.9 line 2, we investigate whether the soft prompt is still useful or not based on our model, though it has been validated in prior CZSL literature [214]. It shows that without the soft prompt, the performance decreases but not too much. However, it is still necessary as it drives the LLM text distributions to align with visual features in training.

Lastly, in Table 7.9 lines 3-5, we further analyze the impact of TFE and VFE modules if they are implemented with the three-layer cross-attention Transformers. The two modules still show contributions to the performance gain. Moreover, compared to the default one-layer setting, using more Transformer layers does not improve the performance, even performing worse.

CHAPTER 8

OPEN-VOCABULARY ACTION DETECTION

8.1 Introduction

Action Detection (AD) aims to recognize actions and spatially and temporally localize the actors in videos. It plays a vital roles in various applications like video surveillance [365, 383, 51], autonomous driving [293], and sport event analysis [184], and it thus draws increasing attentions in recent literature [134, 299, 150, 250, 38, 345, 398, 347, 37].

Existing AD methods are mostly developed in a closed-set setting where the models are trained and tested on videos from the same fixed set of action categories.While significant progress has been made over the past few years [38, 398, 347, 37], the assumption that the training and test videos are from the same action categories limits their application to the real world, where test videos could contain actions beyond the pre-defined training categories. For example, a video surveillance system may be able to detect `fighting`, but other dangerous or suspicious actions like `shooting` and `chasing` will not be detected if the system has not been trained with annotated videos from these action categories. In addition, being able to detect actions in an open world facilitates a comprehensive understanding of videos and opens doors to high-level video understanding tasks, like reasoning [386], forecasting [300], etc., that usually require detecting various actions in videos.

This motivates us to investigate Open-Vocabulary Action Detection (**OVAD**), a task aiming to detect any actions in videos, including both seen categories contained in the training set and unseen categories absent in the training set. However, OVAD is challenging as it requires understanding the human motion dynamics across frames. While motion dynamics modeling has been well studied by the conventional closed-set action detection [74, 38, 398, 347] that takes advantages of full supervision in training, it is challenging in the open-vocabulary setting since there is no

This chapter is adapted from a paper in submission, which was mostly done during the internship of Wentao Bao and Yuxiao Chen at NEC Labs America and partly supported by Dr. Yu Kong at MSU.
"Wentao Bao, Kai Li, Yuxiao Chen, Deep Patel, Martin Renqiang Min, and Yu Kong. Exploiting VLM localizability and semantics for open vocabulary action detection. in *submission*, 2024."

supervision for the unseen action categories.

Recently, harvesting the strong generalization capability of pre-trained large visual-language foundation models (VLMs) [261], various open-vocabulary approaches have been proposed for image recognition [392], object detection [9, 140, 348, 191, 160], and image segmentation [411, 188]. However, these methods are designed for images and do not consider temporal dynamics among video frames. In addition, image VLMs such as the CLIP [261] are struggling to capture the action verbs in text and human motion in videos [229]. This inevitably requires learning the temporal dynamics [132] or fully fine-tuning [264] for recognizing the actions on downstream tasks, which take the risk of poor generalizability to the unseen.

There are a few seminal works that leverage VLMs for open-vocabulary video understanding, including action recognition [243, 340, 204, 264] and temporal action localization [265, 236, 359]. However, for the region-level action detection by VLM, there exists a representation gap between video-level pre-training and the region-level adaptation, which is analogous to the representation gap issue discussed in image-based open-vocabulary object detection literature [9, 140, 348]. Specific to the OVAD task, the representation gap stems from the holistic video-action alignment in pre-training and the downstream region-level sub-tasks, i.e., , region-action alignment and action-relevant person localization. The cause of the representation gap can be attributed to their intrinsically different adaptation goals from pre-trained video VLMs, i.e., , transferring the *semantics* and *localizability* of VLMs from video to regions for the two sub-tasks, respectively.

Re-thinking the Transformer-like design of VLMs, we found that the way of using VLM semantic features and the undervalued localizability of VLMs are both critical to the OVAD task. First, to transfer the video-level semantics to each region, we propose to learn a set of region-wise queries to decode the temporal dynamics from videos, by using the pre-trained video-level features as adaptive semantics conditions. The updated queries and video-level features are further dynamically fused and aligned with the textual semantics for recognition. Second, to exploit the video VLM localizability for region-wise localization, we learn a set of queries to decode the person boxes starting from the prior locations revealed by the VLM visual attention.

Specifically, we develop a query-based open-vocabulary action detector, OpenMixer, to detect any video actions in an open vocabulary. It fits in the family of the detection transformers (DETR) [24, 271, 82, 398, 347]. The basic idea is to decouple the action recognition and localization by learning two sets of queries and corresponding decoding modules. Our OpenMixer consists of a spatial OpenMixer Block (S-OMB) for person localization, a temporal OpenMixer Block (T-OMB) for capturing the region-level temporal motion, and a dynamically fused alignment (DFA) for open-vocabulary action recognition. The S-OMB inherits the localizability of VLMs by the text-patch cross-attention, the T-OMB exploits the visual semantic features of VLMs to better capture the temporal dynamics, and the DFA dynamically fuses the pre-trained semantics into learnable region-level queries for generalizable recognition. Eventually, our model enjoys the merits of semantics and localizability from VLMs as well as the end-to-end detection capability from the DETR pipeline.

Besides, we set up the first OVAD benchmark based on popular action detection datasets, and evaluate technically viable baselines and our proposed model. Empirical results show that our OpenMixer is superior to these baselines. In summary, the main contributions are three-fold:

- We formulate the task of open-vocabulary action detection (OVAD) for the first time, which is valuable while challenging even by foundation models.
- We develop the OpenMixer model that exploits the semantics and localizability of pre-trained video-language models toward the OVAD task.
- We empirically reveal the effectiveness of the proposed modules that show strong generalizability on multiple video action detection datasets.

8.2 Related Work

Spatio-temporal Action Detection. This task aims to localize human actions spatially and temporally in videos and recognize their actions, which has been a fundamental video understanding topic [343, 74, 73, 275]. A line of recent works [299, 150, 306, 250, 36, 403] adopts the two-backbone design to separately extract features of the keyframes and the entire video for actor localization and actor-context relation modeling, respectively. Though they are flexible by taking advantage of both image and video backbones for achieving promising performance, the model parameters are redundant in design and heavy to optimize [37]. With the recent advances in detection transformer (DETR) [24], end-to-end action detection by a single backbone shows impressive performance and thus becomes a more popular design choice [89, 38, 398, 347, 37]. The basic idea is to use a single video transformer to get features of all video frames, and then introduce learnable queries to mix with video features for actor localization and action recognition. Specifically, WOO [38] follows the Sparse RCNN [301] for localization, TubeR [398] learns the action tubes following the classical DETR [24], and STMixer [347] follows the AdaMixer [82] design that achieves the sate-of-the-art performance. The query-based design is advantageous in modeling the interaction between actors and actor context while simplifying the architecture as a single-stage design. Therefore, to further overcome the limitation in an open world, we introduce large-scale pre-trained vision-language models to achieve query-based open-vocabulary action detection.

Open-vocabulary Visual Understanding. The basic idea is to replace the traditional fixed classifier weights with the pre-trained textual representation of class categories. Thanks to the strong alignment capability of pre-trained visual language models (VLM), visual data from unseen classes in an open world can be recognized by the alignment between the visual feature and the text feature of the class names [346, 261]. This motivates a series of works in object detection [385, 206, 384, 405, 140, 191, 160, 348], action recognition [332, 243, 198, 132, 264, 338], and temporal action localization [132, 265, 236, 204]. For video action detection, the iCLIP [117] that tackles the zero-shot action detection is the most relevant work to ours. However, it skipped learning to localize actions by off-the-shelf person detectors [267] and only learns to recognize the unseen actions, which lacks the adaptability to localize action-relevant persons. Besides, the recent image-based open-vocabulary object detectors OV-DETR [384] and CORA [348] share a common spirit with ours by injecting VLM semantics into the learnable queries. However, the query conditions in OV-DETR are class-specific such that they are not adaptive to test-time samples, and the two-stage training in CORA limits its flexibility in video domains. To the best of our knowledge, we are the

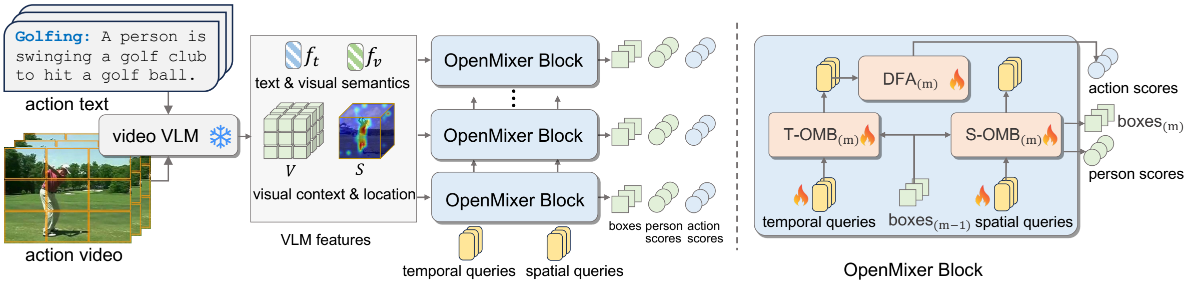


Figure 8.1 **Framework (left) and the OpenMixer Block (right).** Given a video and an open vocabulary of actions, we use prompted classes and a pre-trained video VLM to obtain all kinds of VLM features. With a stack of cascaded OpenMixer blocks and spatial-temporal queries, the model predicts the action scores, person boxes, and their associated person scores.

first to systematically investigate the open vocabulary action detection (OVAD) task and we develop the first query-based OVAD model that can be learned in an end-to-end way.

8.3 Method

In contrast to the closed-set video action detection [299, 134], open-vocabulary action detection (OVAD) aims to recognize and spatiotemporally localize any human actions in videos, including action categories seen and unseen in training. Concretely, an OVAD model is learned from a training set of N_{train} samples $\{(\mathbf{X}, \mathbf{Y})_i | i = 1, \ldots, N_{train}\}$ where $\mathbf{X}$ denotes the training video and $\mathbf{Y}$ denotes the bounding box annotations on the keyframe that consists of box coordinates $\mathbf{b}$ and action category y. In training, an action y is drawn from a fixed set of base action categories C_B. In testing, the learned action detector could detect "any" actions in a given video from the open vocabulary $C_B \cup C_N$, where C_N contains any novel action categories.

8.3.1 OpenMixer

In this paper, we propose the OpenMixer to solve the OVAD task. The OpenMixer model is developed within the family of query-based action DEtection TRansformers (DETR) [347, 38, 398]. Basically, DETR-style models treat the action detection task as a set-to-set prediction problem, i.e., , learning a sparse set of query features from videos to match with the ground truth boxes and action classes. Specific to the OVAD task, the action classes are predicted from an open vocabulary that contains both the base and novel actions.

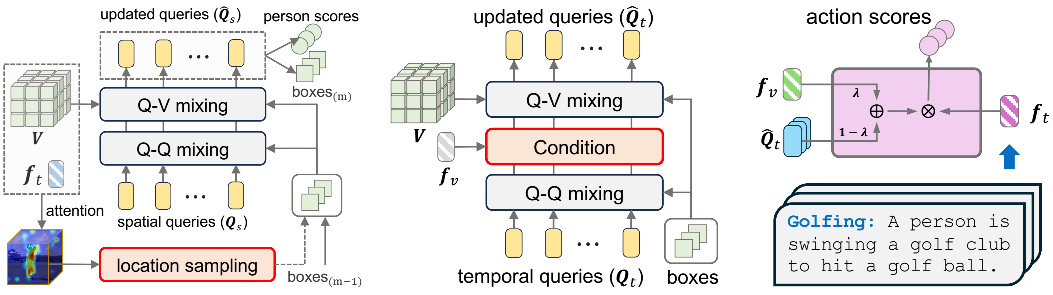
The OpenMixer is shown in Fig. 8.1 (left), given a video $\mathbf{X}$ and a list of text prompted action class as input, we leverage the visual and text encoders Ψ_{VE} and Ψ_{TE} of a pre-trained video VLM to obtain all features of the video and action text, i.e., , $\mathbf{V}, \mathbf{f}_v, \mathbf{S} = \Psi_{\text{VE}}(\mathbf{X})$ and $\mathbf{f}_t = \Psi_{\text{TE}}(y)$. Here, $\mathbf{V}$, $\mathbf{f}_v$ and $\mathbf{S}$ are the 4D patch-level video feature, video-level feature, and video attention, respectively, and $\mathbf{f}_t$ is the text feature of class y. Then, we build M cascaded OpenMixer Blocks (**OMB**) to learn a set of N spatial queries $\mathbf{Q}_s$ and N temporal queries $\mathbf{Q}_t$ from $(\mathbf{V}, \mathbf{S}, \mathbf{f}_v, \mathbf{f}_t)$ for person detection and action classification, respectively. The OMB takes as input all the features from VLM and the $\mathbf{Q}_s$ and $\mathbf{Q}_t$ to predict person boxes, person scores, and action scores.

For the m-th OMB, as shown in Fig. 3.2 (right), it consists of a Temporal OpenMixer Block (T-OMB) Ψ_{α}, a Spatial OpenMixer Block (S-OMB) Ψ_{θ}, and a dynamically fused alignment (DFA). The S-OMB consists of prior location sampling, query-query (Q-Q) mixing by self-attention [328] and query-video (Q-V) mixing by AdaMixer [82], while the T-OMB sequentially consists of Q-Q mixing, query conditioning, and Q-V mixing (see Fig. 8.2a and 8.2b for reference). The DFA module recursively updates the $\mathbf{Q}_s$, $\mathbf{Q}_t$, and person boxes from the $(m-1)$-th OMB, and predict person scores and action scores. These three modules are developed for the OVAD task with the consideration of VLM *semantics* and *localizability*, which will be introduced in the following sections.

8.3.2 Localizability Prior for Spatial OMB

A major challenge for one-stage query-based detectors is the low convergence of localization. One of the causes is the lack of prior knowledge of object locations. Specific to the action detection, recent two-stage action detectors [377, 117, 72, 37] address location prior by an off-the-shelf person detector and RoIAlign [105] cropping, but the feature cropping lacks the spatiotemporal context and suffer from representation gap when a pre-trained VLM is introduced. For recent query-based action detectors [38, 398, 347], the prior knowledge of the person locations is missing in their design. Therefore, when it comes to the OVAD task by VLMs, a natural question is that, *Can we obtain the prior locations of actors from pre-trained VLMs in a cheap way?* Motivated by these considerations, we resort to the visual attention from a pre-trained VLM.

Prior Locations from VLM Attention. Visual attention maps are traditionally represented by



(a) **S-OMB** (b) **T-OMB** (c) **DFA**

Figure 8.2 **Spatial and Temporal OMB, and DFA.** In Fig. 8.2a and Fig. 8.2b, the Q-Q and Q-V mixing modules aim to mix information among queries and across query-visual features, respectively. S-OMB is in Sec. 8.3.2 where the dashed arrow is only used at the 1st stage. T-OMB is in Sec. 8.3.3 and DFA is in Sec. 8.3.4.

the class activation map (CAM) to visually explain recognition models [406, 281]. In the era of ViT [65] and VLM [261], recent works [25, 32, 178] propose to use the multi-head self-attention (MHSA) of the last ViT layer, or the gradient-weighted accumulative product over multi-layer self-attention. However, MHSA is not visually faithful due to the high redundancy of video tokens, and the gradient-based methods suffer from a huge computational cost on video VLMs and ad-hoc implementation for different VLMs. Moreover, due to the lack the token-level video-text correlation, their attention map does not closely relevant to the action specified by the vocabulary. Therefore, an efficient and structure-agnostic CAM is preferable to large video VLMs, which motivates us to use patch-text correlation as VLM attention to encode the location priors.

Specifically, with the D-dimensional 4D video feature $\mathbf{V} \in \mathbb{R}^{T\times hw\times D}$ where T is the number of frames and hw is the number of visual tokens in each frame, the holistic video feature $\mathbf{f}_v \in \mathbb{R}^D$, and the text features of C classes as $\mathcal{F}_t = [\mathbf{f}_t^{(1)}, \dots, \mathbf{f}_t^{(C)}]^\top$. The features are L_2 normalized. We first get the pre-matched text feature $\mathbf{f}_t$ by maximum similarity: $\mathbf{f}_t = \arg\max_{\mathbf{f}_t \in \mathcal{F}_t} \mathbf{f}_v^\top \otimes \mathbf{f}_t$, since we do not have access to the class label in testing. Thus, the inner-product between $\mathbf{f}_t$ and $\mathbf{V}$ determines the patch-text correlation: $\mathbf{S} = \mathbf{V} \otimes \mathbf{f}_t$. Furthermore, as discussed in [179, 178], the q-v attention in self-attention layers shows an opposite heatmap where the foreground regions are associated with low attention value. In practice, we also observed this issue so that similar to [179], our CAM is determined by the reversed patch-text similarity: $\hat{\mathbf{S}} = 1 - \mathbf{S}$. By reshaping and spatial interpolation over $\hat{\mathbf{S}}$, the attention map is obtained for prior location sampling. We treat the $\hat{\mathbf{S}}$ as the prior distribution of person locations indicated by the VLM, thus the top-N positions are sampled as the initial boxes centers: $\{(u, v)_i | i = 1, \ldots, N\} \sim \hat{\mathbf{S}}(u, v, k)$ where (u, v) are 2D coordinates on the keyframe k and N is the number of queries.

Spatial OMB. With the sampled prior locations, the S-OMB (see Fig. 8.2a) that consists of Q-Q and Q-V mixing modules takes as input the video patch features $\mathbf{V}$ and the box prediction $\hat{\mathbf{b}}_{m-1}$ of the previous $(m-1)$-th stage, to update the spatial queries by $\hat{\mathbf{Q}}_s = \Psi_{\theta_m}(\mathbf{V}, \mathbf{Q}_s, \hat{\mathbf{b}}_{m-1})$. The updated spatial queries $\hat{\mathbf{Q}}_s$ are used to predict the person scores $\hat{\mathbf{o}}_m$ and person box offsets $\Delta\hat{\mathbf{b}}_m$ by MLP. Then, the predicted boxes at stage m are updated by $\hat{\mathbf{b}}_m = \hat{\mathbf{b}}_{m-1} + \Delta\hat{\mathbf{b}}_m$, where initial box queries $\hat{\mathbf{b}}_0$ consist of the sampled prior locations and the video spatial range.

The technical intuition behind the design is to encourage the proposed Spatial OMB to learn the box offset $\Delta\mathbf{b}$ starting from the prior locations inherited from the pre-trained VLM. Besides, compared to [347] that uses the fixed non-informative frame centers as prior locations, our VLM attention-based prior locations are adaptive to the test-time video content and vocabulary, which improves not only the seen action localization but also the generalization to the unseen (see Table 8.1 ZSR+TL section).

8.3.3 Adaptive Semantics for Temporal OMB

For the query-based OVAD models, temporal queries are expected to be discriminative for both base and novel actions. This requires a strong capability of content decoding for the query-video (Q-V) mixing module. The pioneering work DETR [24] uses cross-attention while [82, 347] adopt the MLP-Mixer [322]. However, without VLM semantics, these approaches inevitably overfit the seen class data and are unable to detect the unseen. Recent works [384, 348] rightly address the importance of VLM semantics for the query features, but they lack the adaptability to the test-time visual content due to the class-wise semantic condition in [384] and the region prompting in [348]. These motivate us to propose the Temporal OMB that exploits adaptive semantics from pre-trained VLMs.

Temporal OMB. As dipicted in Fig. 8.2b, with the temporal queries $\mathbf{Q}_t$ and the predicted boxes

$\hat{\mathbf{b}}_m$ at the current stage m, the queries are updated by interacting with the video features $\mathbf{V}$ and $\mathbf{f}_v$ by the function $\hat{\mathbf{Q}}_t = \Psi_{\alpha_m}(\mathbf{V}, \mathbf{Q}_t, \mathbf{f}_v, \hat{\mathbf{b}}_m)$. To achieve our motivation of using adaptive semantics, we propose a query update:

$$\hat{\mathbf{Q}}_t = \Psi_{qv}\left(\Psi_{qq}(\mathbf{Q}_t, \mathbf{b}) \oplus \mathbf{f}_v, \mathbf{V}, \mathbf{b}\right), \tag{8.1}$$

where Ψ_{qq} and Ψ_{qv} are Q-Q mixing and Q-V mixing modules by self-attention [328] and AdaMixer [82], respectively. Here, $\mathbf{f}_v$ is the adaptive semantic condition by the pre-trained VLM video feature, which is broadcastly added (denoted as $\oplus$) to the output of Q-Q mixing.

Remark. Note that the adaptiveness of the semantic condition stems from the test-time video feature $\mathbf{f}_v$. Alternatively, when the semantic condition $\mathbf{f}_v$ is changed to $\mathbf{f}_t$ over C classes, it is equivalent to the way in [384]. However, we empirically show this leads to inferior performance (see Table 8.4) especially for the seen action detection. The inferiority can be attributed to the lack of adaptability to test-time video content. Besides, as another alternative, the post-condition that places the condition $\mathbf{f}_v$ after the Q-V mixing, i.e., , $\hat{\mathbf{Q}}_t = \Psi_{qv}\left(\Psi_{qq}(\mathbf{Q}_t, \mathbf{b}), \mathbf{V}, \mathbf{b}\right) \oplus \mathbf{f}_v$, the module Ψ_{qv} is thus to learn the residual of $\mathbf{f}_v$. We empirically found that our pre-condition by Eq. 8.1 is superior to the post-condition, potentially because of the better query features used to learn the important Q-V mixing module.

8.3.4 Dynamically Fused Alignment

To recognize both seen and unseen actions, the model needs to learn discriminative region-wise visual features to align with seen actions, while keeping the generalizable knowledge of the pre-trained VLMs to align with the unseen actions. Dealing with the two goals is challenging. A line of recent approaches uses model adaptation by prompt tuning [132, 67, 236, 338, 348, 117], adapters [251, 81], and gradient preserving [340, 412]. However, these methods either struggle in generalization to novel categories or need to back-propagate through the large VLM that incurs huge computational costs, especially for long videos. Therefore, we resort to a dynamically fused alignment (DFA) for open-vocabulary action recognition, which is lightweight in design and works well for both seen and unseen actions.

Specifically, as shown in Fig. 8.1 and Fig. 8.2c, the DFA is formulated to learn the action classification at each stage m, i.e., , $\hat{\mathbf{y}}_m = \Psi_{\lambda_m}(\hat{\mathbf{Q}}_t, \mathbf{f}_v, \mathbf{f}_t)$, where $\hat{\mathbf{y}}_m$ are the predicted actions for all queries $\hat{\mathbf{Q}}_t$ and the λ_m are the learnable parameters. This module includes *dynamic feature fusion* and *query-text alignment*.

Dynamic Feature Fusion. This step aims to fuse the video-level feature $\mathbf{f}_v$ into each of the queries $\hat{\mathbf{Q}}_t$ dynamically. Specifically, we first repeat N times of the $\mathbf{f}_v$ to be $\mathbf{F}_v \in \mathbb{R}^{N \times D}$. Then, the fusion between $\mathbf{F}_v$ and $\hat{\mathbf{Q}}_t$ is achieved by $\tilde{\mathbf{F}}_v = \lambda \odot \mathbf{F}_v + (1 - \lambda) \odot \hat{\mathbf{Q}}$, where $\lambda \in \mathbb{R}^{N \times 1}$ are learnable in training. The intuition behind the query-specific learnable λ is that, it allows the dynamic contributions of the video-level knowledge from $\mathbf{f}_v$ to the different learnable queries in the set-matching training.

Query-Text Alignment. To make the classification decision by $\tilde{\mathbf{F}}_v$ and open vocabulary of actions, for the action category, we leverage GPT-4 [248] to generate multiple visually descriptive action prompts for each category. With VLM text encoder, the aggregated text features of C classes are represented as $\mathbf{F}_t \in \mathbb{R}^{C \times D}$, where C is the number of classes. Eventually, we use the softmax of visual-text cosine similarity to represent the multi-class classification probability: $P(\hat{y}|\hat{\mathbf{Q}}) = \text{softmax}(\tilde{\mathbf{F}}_v \otimes \mathbf{F}_t^{\top}/\tau)$ where τ is the VLM temperature. In testing, the open-vocabulary action recognition for all queries is achieved by finding the maximum visual-text cosine similarity: $\hat{y} = \arg\max_{y \in C}(\tilde{\mathbf{F}}_v \otimes \mathbf{F}_t^{\top})$.

Note that we do not include the spatial queries $\mathbf{Q}_s$ as the input of our DFA module. This makes the T-OMB Ψ_α and the S-OMB Ψ_θ to be decoupled in training such that the person localization is class-agnostic, which is essential for open-vocabulary tasks according to [236, 348].

8.3.5 Training and Inference

In training, for action localization, we adopt the regular set matching loss following the DETR literature [24, 271, 82]: $\mathcal{L}_{set} = \mathcal{L}_{bce} + \mathcal{L}_{L_1} + \mathcal{L}_{giou}$, where $\mathcal{L}_{bce}$ is a binary cross-entropy loss for person score prediction, $\mathcal{L}_{L_1}$ and $\mathcal{L}_{giou}$ are the coordinate distance and GIoU distance [269] between predicted and ground truth boxes, respectively. Then, we use the Hungarian matching [24] to find the optimal bipartite matching between the predicted and ground truth boxes for each video. For action recognition, we use a multi-class cross-entropy loss $\mathcal{L}_{act}$ so that the total loss for training is $\mathcal{L}_{total} = w_1\mathcal{L}_{set} + w_2\mathcal{L}_{act}$ where the hyperparameters w_1 and w_2 are used to balance between the two subtasks.

During inference, the thresholded person scores determine the kept person boxes, while the action scores assign the action categories to boxes from input class categories.

8.4 Experiments

8.4.1 Experimental Setup

Datasets. Our method is implemented on two commonly-used action detection datasets, i.e., , J-HMDB [124] and UCF101-24 [296]. J-HMDB dataset contains per-frame annotated bounding boxes of persons along with 21 action classes. Similar to [132, 117], with 50% of actions as the novel classes, we randomly split it into 10 base classes for training and 11 novel classes for open-world testing, which results in 10,570 training samples and 9,139 testing samples. UCF101-24 dataset is a subset of UCF101 [296]. It is also per-frame annotated for action detection and contains 24 action classes. With the same 50% splitting strategy, we split it into 12 base classes for training and 12 novel classes for open-world testing.

Evaluation criteria. Following the standard paradigm in action detection literature [150, 72, 38, 398, 347], the model performance is evaluated by video mAP. It evaluates the spatiotemporal action tubes of the detected bounding boxes over the classification and 3D intersection-over-union (IoU). Following [72], the 3DIoU threshold is set to 0.5 for J-HMDB and 0.2 for UCF101-24, respectively.

Implementation details. We experiment with two VLMs including the image pre-trained OpenAI CLIP [261] and video pre-trained CLIP-ViP [357]. We use the same ViT-B/16 architecture for the two VLMs. The VLMs are kept frozen in training. For the image CLIP, we get video-level semantic features by temporal mean pooling. We obtain the patch token features of the last ViT layer and use them to construct the 4D pyramid feature $\mathbf{V}$ by multi-scale residual convolutions. By default, we set the number of queries and OMB stages to 100 and 3, respectively. In training, we set the mini-batch size to 16 and frame sampling by 16×1. The weight of the set loss $\mathcal{L}_{set}$ and action loss $\mathcal{L}_{act}$ are set to 2.0 and 48.0, respectively. Following [24, 38, 347], each intermediate stage is individually supervised by the loss $\mathcal{L}_{set}$ and $\mathcal{L}_{act}$. We set the base learning rate to 1e-5

Table 8.1 **OVAD results.** For all methods use the same pre-trained CLIP-ViP [357] as the frozen VLM and evaluated by video mAP. For the ZSR+ZSL setting, we use Mask RCNN [105] as the ZSL person localizer and use either handcrafted (**HC**) or GPT-generated (**GPT**) prompts for either video- or region-level zero-shot recognition.

Settings	Models	J-HMDB			UCF101-24		
		Mean	Base	Novel	Mean	Base	Novel
ZSR+ZSL	Region + GPT	31.86	30.06	33.51	19.92	21.54	18.29
	Video + HC	54.40	49.89	58.51	31.04	31.43	30.64
	Video + GPT	66.73	64.61	68.66	35.01	34.59	35.43
ZSR+TL	STMixer [347]	63.53	58.27	68.31	36.66	45.26	28.07
	Spatial OpenMixer	74.06	68.04	79.53	40.32	48.80	31.85
E2E	STMixer [347]	49.16	73.06	27.44	33.72	60.91	6.54
	STMixer [347] (w. CoOp [409])	42.27	75.66	11.91	36.12	60.42	11.81
	OpenMixer (w. CoOp [409])	86.86	94.18	80.20	45.11	62.48	27.75
	OpenMixer	86.34	90.75	82.33	47.71	61.18	34.23

and use the AdamW [211] optimizer to train models for 12 epochs on 4 RTX 6000Ada or 2 A100 (80G) GPUs. In testing, the person detection threshold is set to 0.6. We individually test the base and novel classes and report their video-mAP results and the mean on all categories. Our model inference speed is 0.23 s/video per A6000 GPU, with 587M parameters based on CLIP-ViP/B16 VLM.

OVAD task settings. To benchmark methods on OVAD task, three settings are presented considering if the localization and classification are trained or not.

- **ZSR+ZSL** (zero-shot action recognition and actor localization): without any training, we only use pre-trained person detectors such as Mask RCNN [105] to detect persons, and use pre-trained video VLMs such as CLIP-ViP [357] to perform region- or video-level open-vocabulary recognition.

- **ZSR+TL** (zero-shot action recognition and trainable actor localization): we use pre-trained CLIP-ViP [357] to perform video-level action recognition while training the localization modules to detect persons on the training set.

- **E2E** (end-to-end learning): we train and test models in an end-to-end way by using raw videos and vocabulary as input. In this setting, with the same CLIP-ViP [357] backbone, we compare with STMixer [347] using different prompting methods, i.e., , handcraft prompts (HC), soft prompt by CoOp [409], and our recommended GPT-generated prompts (GPT).

8.4.2 Comparative Results

The main results are reported in Table 8.1. To analyze the baseline performance, we summarize the discussion below.

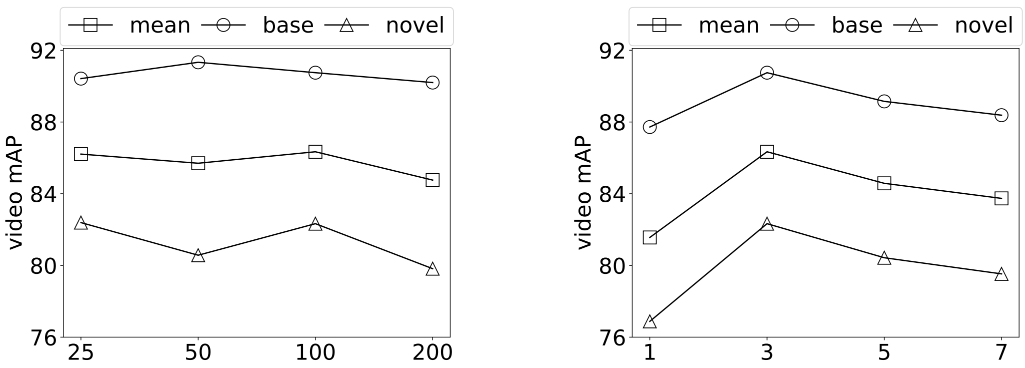
Zero-shot recognition and localization. In the ZSR+ZSL setting, the findings are as follows. First, region-level features (the 1st row) by RoIAlign [105] perform significantly worse than the video-level features (the 3rd row). This indicates that the RoI-cropped features from VLM suffers from a large representation gap between the video-level pre-training and downstream region-level recognition. Second, the descriptive **GPT**-generated prompts (the 3rd row) achieve a better performance than the handcrafted (**HC**) prompt such as the "`a video of person [CLS]`" (the 2nd row). This can be explained by the more transferable knowledge in the GPT prompts than the handcraft ones.

Table 8.2 **Zero-shot action detection**. Following the same 75%-25% split as [117], we report both the frame- and video-level mAP (f-mAP and v-mAP) on novel classes of J-HMDB.

Models	f-mAP	v-mAP
iCLIP [117]	65.41	–
OpenMixer	**77.06**	**81.20**

Zero-shot recognition with learnable localization. Under the ZSR+TL setting, we observe a significant superiority of Spatial OpenMixer to the STMixer baseline, with more than 10% performance gain on the J-HMDB dataset. Since the training is only encouraged to localize actors in videos, the outperformance suggests a good exploitation of the localizability in pre-trained VLMs.

End-to-end learnable OVAD. For the E2E setting, the OpenMixer (the last row) outperforms the simple STMixer baseline (STMixer+VLM) by large margins, with 7.69% and 54.89% of video mAP gains on base and novel categories of the J-HMDB dataset, respectively. Besides, we explored



(a) number of queries

(b) number of stages

Figure 8.3 **Hyperparameters.** We show the video mAP with respect to different numbers of learnable queries and OMB stages.

Table 8.3 **Ablation study.** We show the contribution of each proposed component.

S-OMB	DFA	T-OMB	Mean	Base	Novel
✗	✓	✓	81.77	86.32	77.64
✓	✗	✓	74.06	68.04	79.53
✓	✓	✗	83.47	86.01	81.18
✓	✓	✓	**86.34**	**90.75**	**82.33**

the widely-used VLM adaptation method CoOp [409] that optimizes the context of class names, i.e., , prompt tuning. From Table 8.1, we observe that CoOp improves the base class performance with sacrifice on the novel classes, while the GPT prompted OpenMixer achieves much better performance on novel classes. Lastly, we notice the relatively smaller numbers on UCF101-24 than those on J-HMDB. This reflects the challenging aspects of UCF101-24 dataset such as the long duration (~ 10× longer), heavy background bias, and multi-person scenarios.

Zero-shot action detection. We note the iCLIP [117] defines the zero-shot action detection (ZSAD) task which is different from our OVAD task. The ZSAD only cares about the samples from novel classes while OVAD values both the base and novel classes. Therefore, ZSAD uses all samples from base classes in training and only tests on novel classes. Following the same settings as iCLIP, the results in Table 8.2 show that our method could achieve much better performance than iCLIP, even though iCLIP relies on pre-detected person boxes from YOWO [150].

Table 8.4 **Results of query conditions.** The post/pre and TQ/SQ means that the conditional feature (from video $\mathbf{f}_v$ or from text $\mathbf{f}_t$) is placed after/before the Q-V mixing on the temporal queries (TQ) or spatial queries (SQ).

Methods	Queries	Modalities	Mean	Base	Novel
	w/o condition		83.99	85.86	82.28
post	TQ	video $\mathbf{f}_v$	85.48	88.74	**82.52**
pre	TQ, SQ	video $\mathbf{f}_v$	85.66	90.29	81.45
	TQ	text $\mathbf{f}_t$	76.36	70.25	81.92
		video $\mathbf{f}_v$	**86.34**	**90.75**	82.33

8.4.3 Ablation Study

In this section, we analyze the properties of the OpenMixer model on the J-HMDB dataset. Results of the component-wise ablation are reported in Table 8.3. It shows that all three components could work well. Specifically, without S-OMB which means the attentional location prior is removed, the performance drops significantly especially for the novel classes. If the DFA is removed, we only use the pre-trained VLM feature for zero-shot recognition, it shows that the base class performance is the worst. Without T-OMB which means the semantic condition is removed and both spatial and temporal queries are used for recognition, it shows a decrease of 4.74% and 1.15% on base and novel actions, respectively.

Query condition strategies. Specific to the T-OMB, we further investigate different alternatives of query condition strategies in Table 8.4, which shows the following observations: (1) Without any condition, it performs worse on the base class with 4.89% mAP drop. (2) Pre-condition performs much better than post-condition on base classes (+2%), with negligible performance drop on the novel classes (−0.19%). This can be explained that the pre-condition alleviates the difficulty of content decoding by the following Q-V mixing module. (3) The additional condition on spatial queries (SQ) hurts the performance on both the base and novel classes, because this essentially makes the recognition and localization entangled in training. (4) When using text feature $\mathbf{f}_t$ as a condition, base class performance significantly decreased (−20.50%) and novel class performance also decreased a bit. This is due to the large semantic gap between text feature $\mathbf{f}_t$ and patch-wise

Table 8.5 **Results of fusion strategies.** We explored different strategies to fuse the pre-trained $\mathbf{F}_v$ and learnable queries ($\hat{\mathbf{Q}}_t$ or $\hat{\mathbf{Q}}_s$) within our DFA module.

Methods	Mean	Base	Novel
w/o $\mathbf{F}_v$ ($\lambda = 0$)	68.84	88.94	50.58
w/o $\hat{\mathbf{Q}}_t$ ($\lambda = 1$)	74.06	68.04	79.53
w/o dynamics ($\lambda = 0.5$)	51.48	63.08	40.93
use $\hat{\mathbf{Q}}_s$ by concat. & mlp	85.51	89.19	82.17
DFA (ours)	**86.34**	**90.75**	**82.33**

video token features $\mathbf{V}$, suggesting that the test-time adaptive $\mathbf{f}_v$ is preferable even though $\mathbf{f}_v$ and $\mathbf{f}_t$ are semantically aligned.

Feature fusion strategies. To validate the design choice of our DFA module, we explored different feature fusion strategies, as shown in Table 8.5. The results show that only using the learned query feature $\hat{\mathbf{Q}}_t$ (the 1st row) performs much worse performance on the novel classes, indicating the loss of generalization. If only using the pre-trained feature $\mathbf{F}_v$ (the 2nd row), the model cannot work well on the base classes which indicates an under-fitting to the task. If fusing the features by simple averaging, the performance still lags behind ours as it is not adaptive to the variety in queries. Moreover, we notice that [347] uses both spatial and temporal queries for recognition by MLP layers. Thus, we additionally include the spatial queries $\hat{\mathbf{Q}}_s$ by concatenation with temporal queries $\hat{\mathbf{Q}}_t$ and use MLP layers for dimension reduction. We observe the performance drop, which can be explained as the MLP layers eliminate the benefits of the semantic conditions and makes localization and recognition entangled in training.

Number of queries and OMB stages. In Fig. 8.3a and 8.3b, we show that using 100 queries and 3 OMB stages achieves the best average mAP. The figures also indicate that the number of OMB stages is more important than the number of queries, as the bipartite matching could handle the redundant queries in training. The decreasing trend with more than three OMB stages can be attributed to the risk of overfitting to training data.

Impact of VLMs. We note there is a line of literature [332, 265, 132, 236, 204] built on image CLIP for open-vocabulary video understanding. Therefore, it is interesting to see whether image CLIP also works for the OVAD task. In Table 8.6, we compare OpenMixer with its variants using

Table 8.6 **Effect of VLMs.** We implement the OpenMixer by CLIP-ViP and CLIP with the same ViT-B/16 transformer architecture.

VLMs	Modality	Mean	Base	Novel
CLIP [261]	image	71.60	79.46	64.44
CLIP-ViP [357]	video	**86.34**	**90.75**	**82.33**

Table 8.7 **GPT help temporal localization.** we compute mAP by only using temporal IoU on the J-HMDB dataset.

	Mean(t)	Base(t)	Novel(t)
w/o. GPT	83.57	90.74	77.06
w. GPT	**91.62**	**93.63**	**89.79**

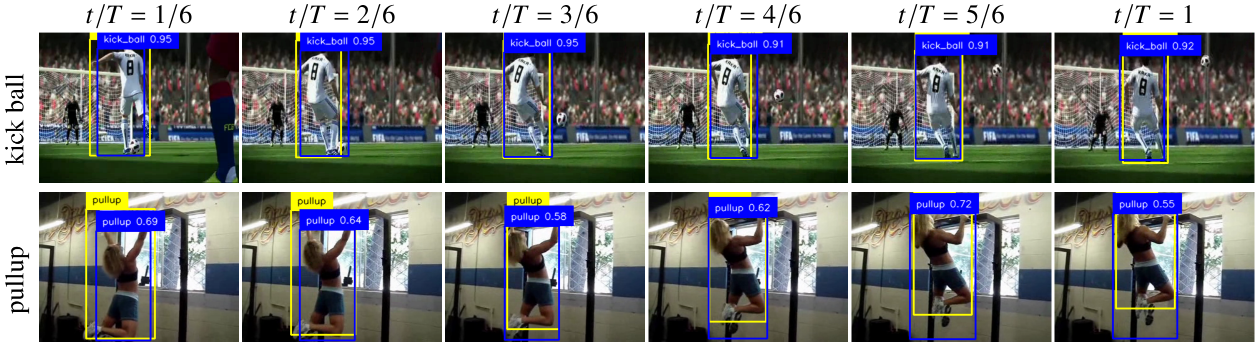


Figure 8.4 **Unseen Action Detection.** We visualize our OpenMixer detections (in blue) and ground truth (in yellow) on two representative videos from **novel** classes. The numbers after class names are confidence scores.

video-based CLIP-ViP [357] and image-based CLIP [261] under the same ViT-B/16 architecture. The results show that the OpenMixer with CLIP performs way worse than the model with CLIP-ViP, because of the limited capacity of image CLIP in capturing video actions.

Impact of person detectors. In Table 8.8, we compare the impact of using external person boxes from off-the-shelf person detectors, i.e., , G-DINO [206] and Mask RCNN [105], in test time on the two best-performed models under the ZSR+ZSL and E2E settings, respectively. It shows that the high-quality boxes from G-DINO could consistently outperform those from Mask RCNN. With the same external test-time boxes, the results of OpenMixer model are consistently better than those of the strongest ZSR+ZSL baseline (Video+GPT). The relatively smaller gains on UCF101-24 than the gains on J-HMDB can be explained by the background bias in UCF videos that restricts VLMs in action recognition.

Can GPT help temporal localization? This question is interesting as how textual prompts from language models like GPT could help temporal localization has not been explored in literature. In Table 8.7, we show that by evaluating the temporal action localization performance, GPT prompts

Table 8.8 **Impact of person detectors.** For E2E setting, predicted boxes from OpenMixer are replaced with boxes from Mask RCNN [105] or G-DINO [206], and their classification scores are assigned by maximum IoU with OpenMixer boxes that have scores.

Models	Test-time person boxes	J-HMDB			UCF101-24		
		Mean	Base	Novel	Mean	Base	Novel
Video + GPT (ZSR+ZSL)	MaskRCNN [105]	66.73	64.61	68.66	35.01	34.59	35.43
	G-DINO [206]	69.72	67.09	72.12	45.43	44.82	46.04
OpenMixer (E2E)	Mask RCNN [105]	83.51	87.45	79.92	42.31	48.48	36.13
	G-DINO [206]	85.06	87.76	82.60	46.56	47.00	46.11

could significantly help.

Qualitative results. We visualize results on the J-HMDB novel categories in Fig. 8.4. They show that OpenMixer could precisely localize and confidently recognize those unseen actions, even though there are multiple persons.

8.5 Conclusion

We present the first work that addresses the open-vocabulary action detection (OVAD) problem. The key challenges are identified as two aspects. The first is how to transfer the semantic knowledge of human motion dynamics from video-level pre-trained VLMs to the spatiotemporal region-level actors in videos for recognizing both seen and unseen actions. The second is how to exploit the prior knowledge of actor location from pre-trained VLMs for action localization. In this paper, we tackle the challenges by developing a query-based detection transformer, OpenMixer, that fully exploits the semantics and localizability of VLMs for action recognition and localization, respectively. Furthermore, we build the first OVAD benchmark that extensively evaluates baselines and our model under various settings, showing the superiority of the OpenMixer while revealing open research questions to explore in the future.

8.6 Supplementary Material

8.6.1 Prompts for Query-Text Alignment

To generate text prompts for each action category, we send a request to GPT [248] by using the template: “`For the action type {CLS}, what are the visual descriptions? Please respond with a list of 16 short sentences.`" where the placeholder “`{CLS}`" is replaced by the action class name from the vocabulary. Thus, we obtained multiple caption-like sentence descriptions of the action. Eventually, the text feature for each class is computed by mean pooling of features from the VLM text encoder given the text prompts.

8.6.2 Explanation of the Reversed Attention

As discussed in the main paper, the seemly counterintuitive phenomenon of reversed visual-text attention has been studied in [179, 178] and we also observed this in our video-based experiments. For CLIP-based models, `[CLS]` token in ViT is aligned to the text semantics so that its attention weight corresponds to the foreground, while the rest L visual token weights are complementary after softmax over $L + 1$ tokens before attention pooling. Therefore, due to the attention pooling, high similarity between text feature (or visual `[CLS]` token feature) and L visual tokens could indicate the background.

8.6.3 Implementation Details

Positional Embedding Interpolation. When using the pre-trained VLM without fine-tuning, an immediate challenge is that the input videos have different spatiotemporal resolutions from the data in VLM pre-training. For example, the CLIP-ViP is pre-trained on input videos with size $12 \times 224 \times 224$ while videos from J-HMDB can be in any resolution after random augmentations in training. A simple solution is to resize the input video size to match with the pre-trained ones. But for the action detection subtask, person localization is sensitive to the input resolution. To handle this challenge, we instead keep the raw resolution as input, but interpolate the pre-trained spatial and temporal positional embeddings. For example, given the CLIP-ViP B16 VLM and an input video with size $T \times H \times W$, we interpolate the 12 temporal positional embeddings $\text{PE}_t \in \mathbb{R}^{12\times D}$ to $\hat{\text{PE}}_t \in \mathbb{R}^{T\times D}$, and interpolate the 196 ($= \frac{224}{16} \times \frac{224}{16}$) spatial positional embeddings $\text{PE}_s \in \mathbb{R}^{196\times D}$ to $\hat{\text{PE}}_s \in \mathbb{R}^{L\times D}$ where $L = \frac{H}{16} \times \frac{W}{16}$. This technique is found useful for the action detection problem.

4D Feature Pyramid. Following the line of detection literature [176, 347], the pre-trained patch token features are transformed into a 4D feature pyramid before the detection head. Let the $\mathbf{H} \in \mathbb{R}^{h\times w\times T\times D}$ be the pre-trained patch token features from the VLM video encoder, where $h \times w$ is the number of patches for each frame, T is the number of video frames, and D is the Transformer

Table 8.9 Results on 50%-50% J-HMDB.

Metrics	(0)	(1)	(2)	(3)	(4)	avg
Mean	86.34	86.29	85.50	86.73	83.40	85.65
Base	90.75	89.89	89.20	87.70	85.36	88.58
Novel	82.33	83.02	82.13	85.85	81.61	82.99

Table 8.10 Results on 50%-50% UCF101-24.

Metrics	(0)	(1)	(2)	(3)	(4)	avg
Mean	46.42	46.28	45.45	47.32	48.30	46.75
Base	59.10	61.11	55.85	62.33	61.25	59.93
Novel	33.73	31.45	35.05	32.31	35.34	33.58

Table 8.11 Results on 75%-25% J-HMDB.

Metrics	(0)	(1)	(2)	(3)	(4)	avg
Mean	75.96	79.43	79.77	81.88	86.56	80.72
Base	74.73	75.21	78.34	82.14	85.46	79.17
Novel	79.03	89.98	83.34	81.23	89.30	84.57

Table 8.12 Results on 75%-25% UCF101-24.

Metrics	(0)	(1)	(2)	(3)	(4)	avg
Mean	55.78	55.83	57.04	57.19	61.85	57.54
Base	64.85	61.83	60.16	58.74	61.82	61.48
Novel	28.55	37.80	47.69	52.55	61.96	45.71

dimension. We use deconvolution or convolution to produce hierarchical feature maps $\tilde{\mathbf{H}}^{(l)}$ by spatial strides $s^{(l)} \in \{1/4, 1/2, 1, 2\}$ where the fractional strides are deconvolutional stides and l indexes the pyramid level. Different from [176, 347] that fully fine-tunes the visual encoder, our VLM visual encoder has to be frozen. Therefore, to allow pre-trained features better utilized by OpenMixer head, we propose to add residual connection at each level of the 4D feature pyramid by spatial interpolation: $\hat{\mathbf{H}}^{(l)} = \phi(\mathbf{H}, s^{(l)}) + \tilde{\mathbf{H}}^{(l)}$. The function ϕ is to spatially interpolate the feature map from the size $h \times w$ to the same resolution of $\tilde{\mathbf{H}}^{(l)}$.

8.6.4 Results on Different Splits

We experiment with five random 50%-50% seen-unseen class splits on both the J-HMDB and UCF101-24 datasets. Full results of video mAP are summarized in Table 8.9 and 8.10. The split (0) is used in all experiments of the main paper. We also experiment with five random 75%-25% seen-unseen class splits on the two datasets, and report results in Table 8.11 and 8.12. As some of human actions are much harder to detect than others and they could be included in either base or novel categories, it is normal that the overall performances on different splits vary significantly. Following the existing literature, we will release all splits.

8.6.5 Limitations and Future Work

As indicated in existing literature, the recent large-scale action detection dataset AVA [98] is not included in this paper, as the AVA human actions are in fine granularity and the person boxes are annotated with multi-label actions. This raises new challenges when adapting video-language foundation models for multi-label action detection problems, which are out of the scope in this paper.

We note the recent success of multi-modal LLMs [200] that uses LLM to re-formulate downstream tasks as a unified generative token prediction problem, which points out a promising direction toward the OVAD task in the future.

CHAPTER 9

CONCLUSIONS AND DISCUSSIONS

9.1 Contribution Summary

In this dissertation, we have made several attempts to endow AI systems to learn from an open-ended visual world. These attempts could handle the major challenges of open-world visual understanding problems, including the open-world visual forecasting that has only limited temporal observations to predict the unseen future or in an unseen environment, open-world visual recognition that data from unknown categories could exist in testing, and open-world vision-language understanding to recognize the unknown from language queries. Exploring these problems is valuable in real-world practice, as human-level intelligence cannot be achieved without the capability of forecasting and detecting the unseen. For this goal, our contributions are summarized below.

First, we empirically found the practicality of Bayesian uncertainty in real-world visual forecasting applications, i.e., the epistemic (model) uncertainty for traffic accident anticipation (Sec. 2) and the aleatoric (data) uncertainty for egocentric 3D hand trajectory prediction (Sec. 4). The deep learning uncertainties, on the one hand, lead to theoretically principled ways to regularize model learning on real-world data, on the other hand, provide trustworthy confidence in downstream decision-making in robotic systems. Moreover, beyond the supervised learning paradigm, we explored deep reinforcement learning (DRL) in traffic accident anticipation (Sec. 3), which naturally mimics the dynamic Markov decision process of human observation and forecasting, resulting in visually explainable and best-performed accident anticipation since unseen driving distractors in an open-world can be suppressed. This indicates that DRL is not only applicable to the extensively studied topics in robotics such as planning and control, but is also potential for video understanding.

Second, we are excited to have found that evidential deep learning (EDL) is powerful in detecting unknown human actions in videos. We technically contributed to several first works such as the open-set action recognition (Sec. 5) and the open-set temporal action localization (Sec. 6). The EDL method is successful because it provides a general Dirichlet assumption on classification problems such that multi-dimensions of classification uncertainties can be applied to regularize the model learning accordingly in diverse downstream tasks. A broader insight of this line of research is that the open-set learning on videos cannot be simply treated the same as the learning on images, due to the implicit challenges and vital importance of modeling the temporal dynamics for complex action understanding.

Lastly, we explore more complex open-world visual understanding problems by vision-language modeling. For image understanding, we contribute to the well-defined compositional zero-shot learning (CZSL) field that aims to recognize unseen compositional concepts from images in an open world. We found that the pre-trained CLIP model can be significantly enhanced for the CZSL task by leveraging LLM-generated compositional class descriptions. For video understanding, we go beyond the traditional closed-set action detection but formulate the first open-vocabulary action detection (OVAD) work that could detect any human actions in videos from an open-ended action vocabulary. On the technical side, we empirically found effective ways to fully utilize the semantics and localizability priors of a video-based CLIP model and LLM descriptions for the OVAD task. In addition to these two topics, there are many other research problems for open-world vision-language understanding to be explored in the future.

9.2 Limitation Discussion

In this part, some unsolved challenges of this dissertation are discussed. We hope these discussions are useful to inspire future following works.

(a) 3D motions and simulation for traffic accident anticipation. In Sec. 2, though it is interesting to formulate the spatio-temporal dynamics of a traffic scene by a sequence of object graphs, the underlying relation between traffic object nodes is from the 2D visual plane. However, in a real-world traffic scenario, the risk of a future accident could be better predicted from the relational motion between cars/pedestrians in a 3D physical world. Moreover, for the ego-vehicle involved accidents in Sec. 2 and Sec. 3, the ego-motion of the camera in 3D space plays an important role in accident forecasting, because an abnormal ego-motion intuitively indicates an out-of-control driving behavior. However, introducing 3D motions of traffic objects or ego-camera is practically infeasible, because the pose information of camera and cars is missing in existing accident video datasets and it can hardly be collected in a real accident, i.e, we cannot take the risk of human lives to collect 3D annotation data. Fortunately, the recent advances in traffic simulation provide the potential. We could use 3D reconstruction from videos/images to represent the traffic objects in 3D [314, 313], import them into traffic simulators such as the CARLA [66] and MetaDrive [173], and simulate any kinds and any amount of traffic accident video data in anywhere. Moreover, through the traffic accident simulation, more data modalities can be created in addition to the visual data, such as high-definition (HD) maps, birds-eye-view (BEV) maps, and IMU sensor data. These modalities are commonly used in autonomous driving, while to the best of our knowledge, they have never been explored for accident anticipation.

(b) Unbiased causal representation learning for open-world vision. As evidenced by many existing literature [87, 122, 42, 183, 139], visual understanding models guided by empirical risk minimization (ERM) suffer from spurious correlation such that the learned visual features are biased toward confounding factors, e.g., background in images, appearance in videos, co-occurrence between visual objects, etc. This may not be bad in a closed-world environment for achieving good fitting performance. But when the model is deployed in an open world where confounding factors exist in unknown data, the model performance shows a dramatic decrease. We have noticed this issue in our work as evidenced in Sec 5. Such an issue also exists in the accident anticipation task. The anticipation model could be biased by visual backgrounds since the backgrounds are typically different for accident and normal videos due to different data collection conditions, i.e., road, weather, and cities. To achieve a debiased open-set video recognition, recent work [389] introduces an adversarial scene reconstruction objective. However, there could be many more tools, e.g., the structured causal model [364], that can be explored for open-world visual understanding.

(c) Grounded language descriptions for vision-language understanding. In our previous works in Sec. 7 and Sec. 8, the LLM-generated class descriptions are class-level free-form sentences, without grounding to the instance-level images or videos. A potential limitation is that the model could heavily rely on the quality of the LLM descriptions. Moreover, performing prompt engineering to generate desired class descriptions could be heuristic in practice. Inspired by the recent

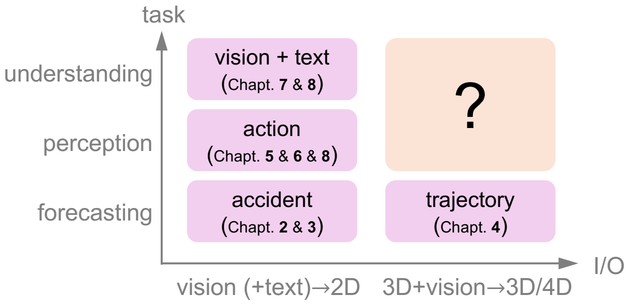


Figure 9.1 4D Visual Understanding.

image captioning models, a better approach could be to first caption the image or video data to get instance-level descriptive texts, which are grounded on the visual data. Then, an LLM [247, 324] or multi-modal LLM [201] can be used to summarize the generated texts into class-level descriptions. The caption-and-summarization scheme takes advantage of grounded language descriptions so that downstream vision-language adaptation will be more robust and generalizable for open-world visual understanding.

9.3 Future Work

To summarize, my existing research serves as a few early steps toward the long-term goal of visual intelligence in the 3D open world. There are many topics such as open-world 3D perception, prediction, and reasoning, which are indispensable for open visual intelligence but under-explored by far. In this section, I outline three major directions that are interesting, valuable, and promising:

4D open visual understanding. Existing research efforts in 3D vision and open-world learning have been studied individually for a long period. However, it would be more practical to apply AI systems in a 4D open environment where both the 3D space and 1D time matter for forecasting, perception, and understanding, as shown in Fig. 9.1. For example, without knowing the 4D human poses, e.g., the time series of 3D human poses, existing open-set video models can hardly tell whether a risky action of `falling down` is unknown or not if the model has been learned on a known action of `sitting down`. This is because the two types of actions show very similar visual appearance in videos from a single-view camera. Instead, with additional 3D depth or multi-view sensors introduced, the captured 4D human poses could distinguish between the known and unknown actions. As shown in Fig. 9.1, the research in the dissertation chapters has explored

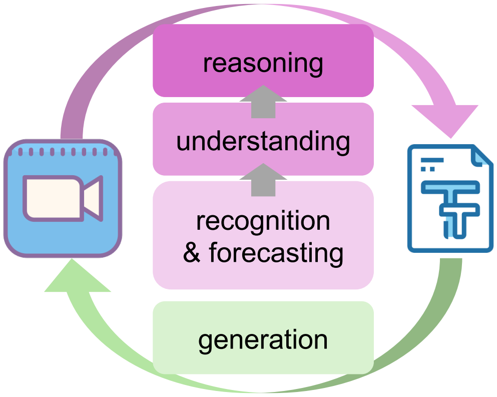


Figure 9.2 Generative Vision-Language Reasoning.

many of the 2D and 3D tasks in an open world, which sheds light on the 4D open visual world in the future, especially for perception and advanced visual understanding tasks.

Generative vision-language reasoning. Though existing foundation models [261, 336] have shown superior performance, they are still limited in open-world applications that require complex visual reasoning [323]. Revisiting our dissertation works [317, 11, 311, 312], the model capabilities are still far from human-level reasoning which is common in video applications. Therefore, it will be important to explore *how to learn the temporal causality of human activities in long-form video* in the future. My recent vision-language understanding work [39] which explores the procedure step localization in long-form videos, sets a starting point towards this direction. Moreover, existing advances in LLMs [247, 324], multi-modal LLM [104, 201, 172], and diffusion models [112, 294, 273] have shown that generative modeling by next-token-prediction or diffusion-denoising is potential to unify visual understanding and visual generation [129, 307]. Also, the latent features of a visual generative model are semantically editable and controllable to benefit the visual recognition [167, 94] and visual reasoning [100, 40, 175]. Motivated by these advances as summarized in Fig. 9.2, as well as my ongoing generative modeling [319] and the collaboration in MLLM [180], building an open-world vision-language reasoning system by generative models will be promising in the future.

9.4 List of Doctoral Works

In this section, the peer-reviewed publications and ongoing works or pre-prints during my Ph.D. period (2019.08 - 2024.07) are listed below for reference.

Peer-reviewed Publications:

1. **Wentao Bao**, Lichang Chen, Heng Huang, Yu Kong, "Prompting Language-Informed Distribution for Compositional Zero-Shot Learning," in *European Conference on Computer Vision (ECCV)*, 2024.
2. Yifan Li, Anh Dao, **Wenta Bao**, Zhen Tan, Tianlong Chen, Huan Liu, Yu Kong, "Facial Behavior Analysis with Instruction Tuning," in *European Conference on Computer Vision (ECCV)*, 2024.
3. Yuxiao Chen, Kai Li, **Wentao Bao**, Deep Patel, Yu Kong, Martin Renqiang Min, Dimitris N. Metaxas, "Learning to Localize Actions in Instructional Videos with LLM-Based Multi-Pathway Text-Video Alignment," in *European Conference on Computer Vision (ECCV)*, 2024.
4. **Wentao Bao**, Lele Chen, Libing Zeng, Zhong Li, Yi Xu, Junsong Yuan, Yu Kong, "Uncertainty-aware State Space Transformer for Egocentric 3D Hand Trajectory Forecasting," in *International Conference on Computer Vision (ICCV)*, 2023.
5. Libing Zeng, Lele Chen, **Wentao Bao**, Zhong Li, Yi Xu, Junsong Yuan, Nima Khademi Kalantari, "3D-aware Facial Landmark Detection via Multiview Consistent Training on Synthetic Data," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.
6. Yuansheng Zhu, **Wentao Bao**, Qi Yu, "Towards Open Set Video Anomaly Detection," in *European Conference on Computer Vision (ECCV)*, 2022.
7. **Wentao Bao**, Qi Yu, Yu Kong, "OpenTAL: Towards Open Set Temporal Action Localization," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022 (**Oral**).

8. Xinmiao Lin, **Wentao Bao**, Matthew Wright, Yu Kong, "Gradient Frequency Modulation for Visually Explaining Video Understanding Models," in *British Machine Vision Conference (BMVC)*, 2021.
9. **Wentao Bao**, Qi Yu, Yu Kong, "Evidential Deep Learning for Open Set Action Recognition," in *International Conference on Computer Vision (ICCV)*, 2021 (**Oral**).
10. **Wentao Bao**, Qi Yu, Yu Kong, "DRIVE: Deep Reinforced Accident Anticipation with Visual Explanation," in *International Conference on Computer Vision (ICCV)*, 2021.
11. Xiwen Dengxiong, **Wentao Bao**, Yu Kong, "Multiple Instance Relational Learning for Video Anomaly Detection," in *International Joint Conference on Neural Network (IJCNN)*, 2021.
12. **Wentao Bao**, Qi Yu, Yu Kong, "Uncertainty-based Traffic Accident Anticipation with Spatio-Temporal Relational Learning," in *28th ACM International Conference on Multimedia (MM)*, 2020.
13. Junwen Chen, **Wentao Bao**, Yu Kong, "Activity-driven Weakly-Supervised Spatio-Temporal Grounding from Untrimmed Videos," in *28th ACM International Conference on Multimedia (MM)*, 2020.
14. Hanbin Hong, **Wentao Bao**, Yuan Hong, Yu Kong, "Privacy Attributes-aware Message Passing Neural Network for Visual Privacy Attributes Classification," in *International Conference on Pattern Recognition (ICPR)*, 2020.
15. Junwen Chen, **Wentao Bao**, Yu Kong, "Group Activity Prediction with Sequential Relational Anticipation Model," in *European Conference on Computer Vision (ECCV)*, 2020.
16. **Wentao Bao**, Qi Yu, Yu Kong, "Object-Aware Centroid Voting for Monocular 3D Object Detection," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020.

Ongoing submissions and pre-prints:

1. **Wentao Bao**, Kai Li, Yuxiao Chen, Deep Patel, Martin Renqiang Min, Yu Kong, "Exploiting VLM Localizability and Semantics for Open Vocabulary Action Detection," (*in submission*), 2024.

2. **Wentao Bao**, Qi Yu, Yu Kong, "Latent Space Energy-based Model for Fine-grained Open Set Recognition," *arXiv preprint, arXiv:2309.10711* (*in submission*), 2024.

3. Suhan Park, **Wentao Bao**, Saniat Sohrawardi, Matthew Wright, Yu Kong, "Open-Set Deepfake Detection by Evidential Deep Learning," (*in submission*), 2024.

4. Yuansheng Zhu, Md Abdullah Al Forhad, **Wentao Bao**, Weishi Shi, Yu Kong, Qi Yu, "Taking No Shortcuts in Lifelong Learning by Following Mixture of Local Experts," (*in submission*), 2024.

5. Xinmiao Lin, **Wentao Bao**, Qi Yu, Yu Kong, "On Model Explanations with Transferable Neural Pathways," *arXiv preprint, arXiv:2309.09887*, 2023.

BIBLIOGRAPHY

[1] Sami Abu-El-Haija, Nisarg Kothari, Joonseok Lee, Paul Natsev, George Toderici, Balakrishnan Varadarajan, and Sudheendra Vijayanarasimhan. Youtube-8m: A large-scale video classification benchmark. *arXiv preprint arXiv:1609.08675*, 2016.

[2] Alexandre Alahi, Kratarth Goel, Vignesh Ramanathan, Alexandre Robicquet, Li Fei-Fei, and Silvio Savarese. Social lstm: Human trajectory prediction in crowded spaces. In *CVPR*, 2016.

[3] Stefano Alletto, Andrea Palazzi, Francesco Solera, Simone Calderara, and Rita Cucchiara. DR(eye)VE: a dataset for attention-based tasks with applications to autonomous and assisted driving. In *CVPR Workshop*, 2016.

[4] Alexander Amini, Wilko Schwarting, Ava Soleimany, and Daniela Rus. Deep evidential regression. In *NeurIPS*, 2020.

[5] Yuval Atzmon, Felix Kreuk, Uri Shalit, and Gal Chechik. A causal view of compositional zero-shot recognition. In *Adv. Neural Inform. Process. Syst.*, 2020.

[6] Hyojin Bahng, Sanghyuk Chun, Sangdoo Yun, Jaegul Choo, and Seong Joon Oh. Learning de-biased representations with biased representations. In *ICML*, 2020.

[7] Yueran Bai, Yingying Wang, Yunhai Tong, Yang Yang, Qiyue Liu, and Junhui Liu. Boundary content graph neural network for temporal action proposal generation. In *ECCV*, 2020.

[8] Duhyeon Bang, Kyungjune Baek, Jiwoo Kim, Yunho Jeon, Jin-Hwa Kim, Jiwon Kim, Jongwuk Lee, and Hyunjung Shim. Logit mixing training for more reliable and accurate prediction. In *IJCAI*, 2022.

[9] Hanoona Bangalath, Muhammad Maaz, Muhammad Uzair Khattak, Salman H Khan, and Fahad Shahbaz Khan. Bridging the gap between object and image-level representations for open-vocabulary detection. In *NIPS*, 2022.

[10] Wentao Bao, Qi Yu, and Yu Kong. Uncertainty-based traffic accident anticipation with spatio-temporal relational learning. In *ACM MM*, 2020.

[11] Wentao Bao, Qi Yu, and Yu Kong. Opental: Towards open set temporal action localization. In *CVPR*, pages 2979–2989, 2022.

[12] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In *Computer Vision–ECCV 2006: 9th European Conference on Computer Vision, Graz, Austria, May 7-13, 2006. Proceedings, Part I 9*, pages 404–417. Springer, 2006.

[13] Justin Bayer and Christian Osendorfer. Learning stochastic recurrent networks. In *NeurIPS*

Workshop, 2014.

[14] Jessa Bekker and Jesse Davis. Learning from positive and unlabeled data: A survey. *Machine Learning*, 109(4):719–760, 2020.

[15] Abhijit Bendale and Terrance Boult. Towards open world recognition. In *CVPR*, 2015.

[16] Abhijit Bendale and Terrance E. Boult. Towards open set deep networks. In *CVPR*, 2016.

[17] Huikun Bi, Ruisi Zhang, Tianlu Mao, Zhigang Deng, and Zhaoqi Wang. How can I see my future? FvTraj: Using first-person view for pedestrian trajectory prediction. In *ECCV*, 2020.

[18] Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural networks. In *International Conference on Machine Learning*, 2015.

[19] James Bradbury, Stephen Merity, Caiming Xiong, and Richard Socher. Quasi-recurrent neural networks. In *International Conference on Learning Representations*, 2017.

[20] Shyamal Buch, Victor Escorcia, Bernard Ghanem, Li Fei-Fei, and Juan Carlos Niebles. End-to-end, single-stream temporal action detection in untrimmed videos. In *BMVC*, 2017.

[21] Pau Panareda Busto, Ahsan Iqbal, and Juergen Gall. Open set domain adaptation for image and action recognition. *IEEE transactions on pattern analysis and machine intelligence*, 42(2):413–429, 2018.

[22] Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *CVPR*, 2015.

[23] Zhaowei Cai and Nuno Vasconcelos. Cascade R-CNN: Delving into high quality object detection. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[24] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, 2020.

[25] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *ICCV*, pages 9650–9660, 2021.

[26] Luigi Carratino, Moustapha Cissa, Rodolphe Jenatton, and Jean-Philippe Vert. On mixup regularization. *JMLR*, 23(325), 2022.

[27] J. Carreira and Andrew Zisserman. Quo vadis, action recognition? A new model and the kinetics dataset. In *CVPR*, 2017.

[28] Jun Cen, Peng Yun, Junhao Cai, Michael Yu Wang, and Ming Liu. Deep metric learning for open world semantic segmentation. In *ICCV*, 2021.

[29] Fu-Hsiang Chan, Yu-Ting Chen, Yu Xiang, and Min Sun. Anticipating accidents in dashcam videos. In *Asian Conference on Computer Vision*, 2016.

[30] Yu-Wei Chao, Sudheendra Vijayanarasimhan, Bryan Seybold, David A Ross, Jia Deng, and Rahul Sukthankar. Rethinking the faster r-cnn architecture for temporal action localization. In *CVPR*, 2018.

[31] Bertrand Charpentier, Daniel Zügner, and Stephan Günnemann. Posterior network: Uncertainty estimation without ood samples via density-based pseudo-counts. In *NeurIPS*, 2020.

[32] Hila Chefer, Shir Gur, and Lior Wolf. Generic attention-model explainability for interpreting bi-modal and encoder-decoder transformers. In *ICCV*, pages 397–406, 2021.

[33] Guangyao Chen, Peixi Peng, Xiangqian Wang, and Yonghong Tian. Adversarial reciprocal points learning for open set recognition. *IEEE TPAMI*, 2021.

[34] Guangyao Chen, Limeng Qiao, Yemin Shi, Peixi Peng, Jia Li, Tiejun Huang, Shiliang Pu, and Yonghong Tian. Learning open set network with discriminative reciprocal points. In *ECCV*, 2020.

[35] Junwen Chen, Wentao Bao, and Yu Kong. Group activity prediction with sequential relational anticipation model. In *ECCV*, 2020.

[36] Lei Chen, Zhan Tong, Yibing Song, Gangshan Wu, and Limin Wang. Cycleacr: Cycle modeling of actor-context relations for video action detection. *arXiv preprint arXiv:2303.16118*, 2023.

[37] Lei Chen, Zhan Tong, Yibing Song, Gangshan Wu, and Limin Wang. Efficient video action detection with token dropout and context refinement. In *ICCV*, 2023.

[38] Shoufa Chen, Peize Sun, Enze Xie, Chongjian Ge, Jiannan Wu, Lan Ma, Jiajun Shen, and Ping Luo. Watch only once: An end-to-end video action detection framework. In *ICCV*, pages 8178–8187, 2021.

[39] Yuxiao Chen, Kai Li, Wentao Bao, Deep Patel, Yu Kong, Martin Renqiang Min, and Dimitris N. Metaxas. Learning to localize actions in instructional videos with llm-based multi-pathway text-video alignment. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2024.

[40] Jaemin Cho, Abhay Zala, and Mohit Bansal. Dall-eval: Probing the reasoning skills and social biases of text-to-image generation models. In *ICCV*, 2023.

[41] Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2014.

[42] Jinwoo Choi, Chen Gao, Joseph CE Messou, and Jia-Bin Huang. Why can't i dance in the mall? learning to mitigate scene bias in action recognition. In *NeurIPS*, 2019.

[43] Junyoung Chung, Kyle Kastner, Laurent Dinh, Kratarth Goel, Aaron C Courville, and Yoshua Bengio. A recurrent latent variable model for sequential data. In *NeurIPS*, 2015.

[44] Charles E Connor, Howard E Egeth, and Steven Yantis. Visual attention: bottom-up versus top-down. *Current Biology*, 14(19):R850–R852, 2004.

[45] MMAction2 Contributors. Openmmlab's next generation video understanding toolbox and benchmark. https://github.com/open-mmlab/mmaction2, 2020.

[46] G. Corcoran and J. Clark. Traffic risk assessment: A two-stream approach using dynamic attention. In *Conference on Computer and Robot Vision*, 2019.

[47] M. Cornia, L. Baraldi, G. Serra, and R. Cucchiara. A deep multi-level network for saliency prediction. In *ICPR*, 2016.

[48] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, volume 1, pages 886–893. Ieee, 2005.

[49] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, et al. Scaling egocentric vision: The epic-kitchens dataset. In *ECCV*, 2018.

[50] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, et al. The epic-kitchens dataset: Collection, challenges and baselines. *IEEE TPAMI*, 43(11):4125–4141, 2020.

[51] Ishan Dave, Zacchaeus Scheffer, Akash Kumar, Sarah Shiraz, Yogesh Singh Rawat, and Mubarak Shah. Gabriellav2: Towards better generalization in surveillance videos for action detection. In *WACV*, pages 122–132, 2022.

[52] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *Proceedings of Neural Information Processing Systems*, 2016.

[53] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A

large-scale hierarchical image database. In *CVPR*, 2009.

[54] Li Deng. The mnist database of handwritten digit images for machine learning research [best of the web]. *IEEE signal processing magazine*, 29(6):141–142, 2012.

[55] T. Deng, H. Yan, L. Qin, T. Ngo, and B. S. Manjunath. How do drivers allocate their potential attention? driving fixation prediction via convolutional neural networks. *IEEE TITS*, 21(5):2146–2154, 2020.

[56] Tao Deng, Andong Chen, Min Gao, and Hongmei Yan. Top-down based saliency model in traffic driving environment. In *ITSC*, 2014.

[57] Tao Deng, Hongmei Yan, and Yong-Jie Li. Learning to boost bottom-up fixation prediction in driving environments via random forest. *IEEE TITS*, 19(9):3059–3067, 2017.

[58] John S. Denker and Yann LeCun. Transforming neural-net output levels to probability distributions. In *Proceedings of Neural Information Processing Systems*, 1990.

[59] Nachiket Deo and Mohan M Trivedi. Convolutional social pooling for vehicle trajectory prediction. In *CVPR*, 2018.

[60] Mohammad Mahdi Derakhshani, Enrique Sanchez, Adrian Bulat, Victor Guilherme Turrisi da Costa, Cees G. M. Snoek, Georgios Tzimiropoulos, and Brais Martinez. Bayesian prompt learning for image-language model generalization. In *ICCV*, 2023.

[61] Akshay Dhamija, Manuel Gunther, Jonathan Ventura, and Terrance Boult. The overlooked elephant of object detection: Open set. In *WACV*, 2020.

[62] Akshay Raj Dhamija, Manuel Günther, and Terrance E Boult. Reducing network agnostophobia. In *NeurIPS*, 2018.

[63] Christian Diller, Thomas Funkhouser, and Angela Dai. Forecasting characteristic 3d poses of human actions. In *CVPR*, 2022.

[64] Luke Ditria, Benjamin J Meyer, and Tom Drummond. OpenGAN: Open set generative adversarial networks. In *ACCV*, 2020.

[65] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021.

[66] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. CARLA: An open urban driving simulator. In *ACRL*, 2017.

[67] Yu Du, Fangyun Wei, Zihe Zhang, Miaojing Shi, Yue Gao, and Guoqi Li. Learning to prompt for open-vocabulary object detection with vision-language model. In *CVPR*, pages 14084–14093, 2022.

[68] J. Fang, D. Yan, J. Qiao, J. Xue, H. Wang, and S. Li. DADA-2000: Can driving accident be predicted by driver attention? analyzed by a benchmark. In *IEEE Intelligent Transportation Systems Conference*, 2019.

[69] Zhen Fang, Jie Lu, Anjin Liu, Feng Liu, and Guangquan Zhang. Learning bounds for open-set learning. In *ICML*, 2021.

[70] Alireza Fathi, Ali Farhadi, and James M Rehg. Understanding egocentric activities. In *ICCV*, 2011.

[71] Mishal Fatima, Muhammad Umar Karim Khan, and Chong Min Kyung. Global feature aggregation for accident anticipation. *arXiv preprint arXiv:2006.08942*, 2020.

[72] Gueter Josmy Faure, Min-Hung Chen, and Shang-Hong Lai. Holistic interaction transformer network for action detection. In *WACV*, pages 3340–3350, 2023.

[73] Christoph Feichtenhofer. X3d: Expanding architectures for efficient video recognition. In *CVPR*, pages 203–213, 2020.

[74] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *ICCV*, 2019.

[75] Marco Fraccaro, Søren Kaae Sønderby, Ulrich Paquet, and Ole Winther. Sequential neural models with stochastic layers. In *NeurIPS*, 2016.

[76] Hao Fu, Chunyuan Li, Xiaodong Liu, Jianfeng Gao, Asli Celikyilmaz, and Lawrence Carin. Cyclical annealing schedule: A simple approach to mitigating KL vanishing. In *NACCL*, 2019.

[77] Scott Fujimoto, Herke Van Hoof, and David Meger. Addressing function approximation error in actor-critic methods. In *ICML*, 2018.

[78] Yarin Gal. *Uncertainty in Deep Learning*. PhD thesis, Department of Engineering, University of Cambridge, 2016.

[79] Yarin Gal and Zoubin Ghahramani. Bayesian convolutional neural networks with bernoulli approximate variational inference. In *International Conference on Learning Representations (Workshop)*, 2016.

[80] Nisal Menuka Gamage, Deepana Ishtaweera, Martin Weigel, and Anusha Withana. So predictable! continuous 3d hand trajectory prediction in virtual reality. In *UIST*, 2021.

[81] Peng Gao, Shijie Geng, Renrui Zhang, Teli Ma, Rongyao Fang, Yongfeng Zhang, Hongsheng Li, and Yu Qiao. Clip-adapter: Better vision-language models with feature adapters. *International Journal of Computer Vision*, pages 1–15, 2023.

[82] Ziteng Gao, Limin Wang, Bing Han, and Sheng Guo. Adamixer: A fast-converging query-based object detector. In *CVPR*, pages 5364–5373, 2022.

[83] Zongyuan Ge, Sergey Demyanov, Zetao Chen, and Rahil Garnavi. Generative OpenMax for multi-class open set classification. In *BMVC*, 2017.

[84] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? The KITTI Vision Benchmark Suite. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

[85] Wilson S Geisler and Jeffrey S Perry. Real-time foveated multiresolution system for low-bandwidth video communication. In *Human Vision and Electronic Imaging III*, volume 3299, pages 294–305, 1998.

[86] Chuanxing Geng, Sheng-jun Huang, and Songcan Chen. Recent advances in open set recognition: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 2020.

[87] Soumya Suvra Ghosal and Yixuan Li. Are vision transformers robust to spurious correlations? *International Journal of Computer Vision*, pages 1–21, 2023.

[88] Partha Ghosh, Mehdi SM Sajjadi, Antonio Vergari, Michael Black, and Bernhard Schölkopf. From variational to deterministic autoencoders. In *ICLR*, 2020.

[89] Rohit Girdhar, Joao Carreira, Carl Doersch, and Andrew Zisserman. Video action transformer network. In *CVPR*, June 2019.

[90] Rohit Girdhar and Kristen Grauman. Anticipative video transformer. In *ICCV*, 2021.

[91] Laurent Girin, Simon Leglaive, Xiaoyu Bie, Julien Diard, Thomas Hueber, and Xavier Alameda-Pineda. Dynamical variational autoencoders: A comprehensive review. *Foundations and Trends in Machine Learning*, 15(1-2):1–175, 2021.

[92] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NeurIPS*, 2014.

[93] Yash Goyal, Ziyan Wu, Jan Ernst, Dhruv Batra, Devi Parikh, and Stefan Lee. Counterfactual visual explanations. In *ICML*, 2019.

[94] Alexandros Graikos, Nikolay Malkin, Nebojsa Jojic, and Dimitris Samaras. Diffusion models as plug-and-play priors. In *NeurIPS*, 2022.

[95] Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, et al. Ego4d: Around the world in 3,000 hours of egocentric video. In *CVPR*, 2022.

[96] Alex Graves. Practical variational inference for neural networks. In *Proceedings of Neural Information Processing Systems*, 2011.

[97] Arthur Gretton, Olivier Bousquet, Alex Smola, and Bernhard Schölkopf. Measuring statistical dependence with hilbert-schmidt norms. In *ICALT*, 2005.

[98] Chunhui Gu, Chen Sun, David A Ross, Carl Vondrick, Caroline Pantofaru, Yeqing Li, Sudheendra Vijayanarasimhan, George Toderici, Susanna Ricco, Rahul Sukthankar, et al. Ava: A video dataset of spatio-temporally localized atomic visual actions. In *CVPR*, 2018.

[99] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *ICML*, 2017.

[100] Tanmay Gupta and Aniruddha Kembhavi. Visual programming: Compositional visual reasoning without training. In *CVPR*, 2023.

[101] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft Actor-Critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *ICML*, 2018.

[102] Tuomas Haarnoja, Aurick Zhou, Kristian Hartikainen, George Tucker, Sehoon Ha, Jie Tan, Vikash Kumar, Henry Zhu, Abhishek Gupta, Pieter Abbeel, et al. Soft actor-critic algorithms and applications. *arXiv preprint arXiv:1812.05905*, 2018.

[103] Ehsan Hajiramezanali, Arman Hasanzadeh, Krishna Narayanan, Nick Duffield, Mingyuan Zhou, and Xiaoning Qian. Variational graph recurrent neural networks. In *Proceedings of Neural Information Processing Systems*, 2019.

[104] Yaru Hao, Haoyu Song, Li Dong, Shaohan Huang, Zewen Chi, Wenhui Wang, Shuming Ma, and Furu Wei. Language models are general-purpose interfaces. *arXiv preprint arXiv:2206.06336*, 2022.

[105] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, pages 2961–2969, 2017.

[106] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[107] Ruifei He, Shuyang Sun, Xin Yu, Chuhui Xue, Wenqing Zhang, Philip Torr, Song Bai, and Xiaojuan Qi. Is synthetic data from generative models ready for image recognition? In *ICLR*, 2023.

[108] Yihui He, Chenchen Zhu, Jianren Wang, Marios Savvides, and Xiangyu Zhang. Bounding box regression with uncertainty for accurate object detection. In *CVPR*, 2019.

[109] Fabian Caba Heilbron, Juan Carlos Niebles, and Bernard Ghanem. Fast temporal activity proposals for efficient detection of human actions in untrimmed videos. In *CVPR*, 2016.

[110] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-VAE: Learning basic visual concepts with a constrained variational framework. In *ICLR*, 2017.

[111] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[112] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[113] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[114] Neil Houlsby, Ferenc Huszár, Zoubin Ghahramani, and Máté Lengyel. Bayesian active learning for classification and preference learning. *arXiv preprint arXiv:1112.5745*, 2011.

[115] H. Hu, Y. Lin, M. Liu, H. Cheng, Y. Chang, and M. Sun. Deep 360 Pilot: Learning a deep agent for piloting through 360° sports videos. In *CVPR*, 2017.

[116] Siteng Huang, Biao Gong, Yutong Feng, Yiliang Lv, and Donglin Wang. Troika: Multi-path cross-modal traction for compositional zero-shot learning. In *CVPR*, 2024.

[117] Wei-Jhe Huang, Jheng-Hsien Yeh, Min-Hung Chen, Gueter Josmy Faure, and Shang-Hong Lai. Interaction-aware prompting for zero-shot spatio-temporal action detection. In *ICCV Workshop*, pages 284–293, 2023.

[118] Xiaohui Huang, Pan He, Anand Rangarajan, and Sanjay Ranka. Intelligent intersection: Two-stream convolutional networks for real-time near-accident detection in traffic video. *ACM TSAS*, 6(2), 2020.

[119] Dat Huynh and Ehsan Elhamifar. Compositional zero-shot learning via fine-grained dense feature composition. In *Adv. Neural Inform. Process. Syst.*, 2020.

[120] Jaedong Hwang, Seoung Wug Oh, Joon-Young Lee, and Bohyung Han. Exemplar-based open-set panoptic segmentation network. In *CVPR*, 2021.

[121] Phillip Isola, Joseph J Lim, and Edward H Adelson. Discovering states and transformations in image collections. In *CVPR*, 2015.

[122] Pavel Izmailov, Polina Kirichenko, Nate Gruver, and Andrew G Wilson. On feature learning in the presence of spurious correlations. In *NeurIPS*, 2022.

[123] Lalit P Jain, Walter J Scheirer, and Terrance E Boult. Multi-class open set recognition using probability of inclusion. In *ECCV*, 2014.

[124] Hueihan Jhuang, Juergen Gall, Silvia Zuffi, Cordelia Schmid, and Michael J Black. Towards understanding action recognition. In *ICCV*, pages 3192–3199, 2013.

[125] Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual prompt tuning. In *ECCV*, 2022.

[126] Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. Mistral 7b. *arXiv preprint arXiv:2310.06825*, 2023.

[127] M. Jiang, X. Boix, G. Roig, J. Xu, L. Van Gool, and Q. Zhao. Learning to predict sequences of human visual fixations. *IEEE TNNLS*, 27(6):1241–1252, 2016.

[128] Y.-G. Jiang, J. Liu, A. Roshan Zamir, G. Toderici, I. Laptev, M. Shah, and R. Sukthankar. THUMOS challenge: Action recognition with a large number of classes. http://crcv.ucf.edu/THUMOS14/, 2014.

[129] Yang Jin, Kun Xu, Liwei Chen, Chao Liao, Jianchao Tan, Bin Chen, Chenyi Lei, An Liu, Chengru Song, Xiaoqiang Lei, et al. Unified language-vision pretraining with dynamic discrete visual tokenization. *arXiv preprint arXiv:2309.04669*, 2023.

[130] Audun Jøsang. *Subjective logic*. Springer, 2016.

[131] KJ Joseph, Salman Khan, Fahad Shahbaz Khan, and Vineeth N Balasubramanian. Towards open world object detection. In *CVPR*, 2021.

[132] Chen Ju, Tengda Han, Kunhao Zheng, Ya Zhang, and Weidi Xie. Prompting visual-language models for efficient video understanding. In *ECCV*, 2022.

[133] Pedro Ribeiro Mendes Júnior, Terrance E Boult, Jacques Wainer, and Anderson Rocha. Specialized support vector machines for open-set recognition. *arXiv preprint arXiv:1606.03802*, 2016.

[134] Vicky Kalogeiton, Philippe Weinzaepfel, Vittorio Ferrari, and Cordelia Schmid. Action tubelet detector for spatio-temporal action localization. In *ICCV*, 2017.

[135] Shyamgopal Karthik, Massimiliano Mancini, and Zeynep Akata. Kg-sp: Knowledge guided simple primitives for open world compositional zero-shot learning. In *CVPR*, 2022.

[136] Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? In *Proceedings of Neural Information Processing Systems*, 2017.

[137] Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *CVPR*, 2018.

[138] Muhammad Uzair Khattak, Hanoona Rasheed, Muhammad Maaz, Salman Khan, and Fahad Shahbaz Khan. Maple: Multi-modal prompt learning. In *CVPR*, 2023.

[139] Byungju Kim, Hyunwoo Kim, Kyungsu Kim, Sungjin Kim, and Junmo Kim. Learning not to learn: Training deep neural networks with biased data. In *CVPR*, 2019.

[140] Dahun Kim, Anelia Angelova, and Weicheng Kuo. Region-aware pretraining for open-vocabulary object detection with vision transformers. In *CVPR*, 2023.

[141] Jinkyu Kim and John Canny. Interpretable learning for self-driving cars by visualizing causal attention. In *ICCV*, 2017.

[142] Jinkyu Kim, Anna Rohrbach, Trevor Darrell, John Canny, and Zeynep Akata. Textual explanations for self-driving vehicles. In *ECCV*, 2018.

[143] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. In *International Conference on Learning Representations*, 2013.

[144] Thomas N Kipf and Max Welling. Variational graph auto-encoders. In *Proceedings of Neural Information Processing Systems (Workshop)*, 2016.

[145] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, 2017.

[146] Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *ICML*, 2017.

[147] Shu Kong and Deva Ramanan. OpenGAN: Open-set recognition via open data generation. In *ICCV*, 2021.

[148] Yu Kong and Yun Fu. Human action recognition and prediction: A survey. *arXiv preprint arXiv:1806.11230*, 2018.

[149] Yu Kong, Zhiqiang Tao, and Yun Fu. Adversarial action prediction networks. *IEEE TPAMI*, 2018.

[150] Okan Köpüklü, Xiangyu Wei, and Gerhard Rigoll. You only watch once: A unified cnn architecture for real-time spatiotemporal action localization. *arXiv preprint arXiv:1911.06644*, 2019.

[151] Florian Kraus and Klaus Dietmayer. Uncertainty estimation in one-stage object detection. In *ITSC*, 2019.

[152] Rahul G Krishnan, Uri Shalit, and David Sontag. Deep kalman filters. *arXiv preprint arXiv:1511.05121*, 2015.

[153] Ranganath Krishnan, Mahesh Subedar, and Omesh Tickoo. BAR: Bayesian activity recognition using variational inference. In *NeurIPS*, 2018.

[154] Ranganath Krishnan, Mahesh Subedar, and Omesh Tickoo. Specifying weight priors in bayesian deep neural networks with empirical bayes. In *AAAI*, 2020.

[155] Ranganath Krishnan and Omesh Tickoo. Improving model calibration with accuracy versus uncertainty optimization. In *NeurIPS*, 2020.

[156] Alex Krizhevsky and Geoff Hinton. Convolutional deep belief networks on cifar-10. *Unpublished manuscript*, 40(7):1–9, 2010.

[157] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

[158] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

[159] Hildegard Kuehne, Hueihan Jhuang, Estíbaliz Garrote, Tomaso Poggio, and Thomas Serre. HMDB: a large video database for human motion recognition. In *ICCV*, 2011.

[160] Weicheng Kuo, Yin Cui, Xiuye Gu, AJ Piergiovanni, and Anelia Angelova. F-vlm: Open-vocabulary object detection upon frozen vision and language models. In *ICLR*, 2022.

[161] Hyeongjun Kwon, Taeyong Song, Somi Jeong, Jin Kim, Jinhyun Jang, and Kwanghoon Sohn. Probabilistic prompt learning for dense prediction. In *CVPR*, 2023.

[162] Taein Kwon, Bugra Tekin, Jan Stühmer, Federica Bogo, and Marc Pollefeys. H2O: Two hands manipulating objects for first person interaction recognition. In *ICCV*, 2021.

[163] Yongchan Kwon, Joong-Ho Won, Beom Joon Kim, and Myunghee Cho Paik. Uncertainty quantification using bayesian neural networks in classification: Application to ischemic stroke lesion segmentation. In *Medical Imaging with Deep Learning*, 2018.

[164] Brenden M Lake, Tomer D Ullman, Joshua B Tenenbaum, and Samuel J Gershman. Building machines that learn and think like people. *Behavioral and brain sciences*, 40, 2017.

[165] Olivier Le Meur and Zhi Liu. Saccadic model of eye movements for free-viewing condition.

Vision Research, 116:152 – 164, 2015.

[166] Yann LeCun, Corinna Cortes, and Christopher J.C. Burges. The mnist database of handwritten digits. http://yann.lecun.com/exdb/mnist.

[167] Sharon Lee, Yunzhi Zhang, Shangzhe Wu, and Jiajun Wu. Language-informed visual concept learning. *arXiv preprint arXiv:2312.03587*, 2023.

[168] Simon Leglaive, Xavier Alameda-Pineda, Laurent Girin, and Radu Horaud. A recurrent variational autoencoder for speech enhancement. In *ICASSP*, 2020.

[169] Martha Lewis, Qinan Yu, Jack Merullo, and Ellie Pavlick. Does clip bind concepts? probing compositionality in large image models. *arXiv preprint arXiv:2212.10537*, 2022.

[170] Buyu Li, Yu Liu, and Xiaogang Wang. Gradient harmonized single-stage detector. In *AAAI*, 2019.

[171] Fayin Li and Harry Wechsler. Open set face recognition using transduction. *IEEE transactions on pattern analysis and machine intelligence*, 27(11):1686–1697, 2005.

[172] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *ICML*, 2023.

[173] Quanyi Li, Zhenghao Peng, Lan Feng, Qihang Zhang, Zhenghai Xue, and Bolei Zhou. Metadrive: Composing diverse driving scenarios for generalizable reinforcement learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[174] Xiangyu Li, Xu Yang, Kun Wei, Cheng Deng, and Muli Yang. Siamese contrastive embedding network for compositional zero-shot learning. In *CVPR*, 2022.

[175] Xiaochuan Li, Baoyu Fan, Runze Zhang, Liang Jin, Di Wang, Zhenhua Guo, Yaqian Zhao, and Rengang Li. Image content generation with causal reasoning. In *AAAI*, 2024.

[176] Yanghao Li, Hanzi Mao, Ross Girshick, and Kaiming He. Exploring plain vision transformer backbones for object detection. In *ECCV*, pages 280–296, 2022.

[177] Yanghao Li, Tushar Nagarajan, Bo Xiong, and Kristen Grauman. Ego-exo: Transferring visual representations from third-person to first-person videos. In *CVPR*, 2021.

[178] Yi Li, Hualiang Wang, Yiqun Duan, and Xiaomeng Li. Clip surgery for better explainability with enhancement in open-vocabulary tasks. *arXiv preprint arXiv:2304.05653*, 2023.

[179] Yi Li, Hualiang Wang, Yiqun Duan, Hang Xu, and Xiaomeng Li. Exploring visual interpretability for contrastive language-image pre-training. *arXiv preprint arXiv:2209.07046*, 2022.

[180] Yifan Li, Anh Dao, Wentao Bao, Zhen Tan, Tianlong Chen, Huan Liu, and Yu Kong. Facial behavior analysis with instruction tuning. In *ECCV*, 2024.

[181] Yiming Li, Ziang Cao, Andrew Liang, Benjamin Liang, Luoyao Chen, Hang Zhao, and Chen Feng. Egocentric prediction of action target in 3d. In *CVPR*, 2022.

[182] Yin Li, Miao Liu, and James M. Rehg. In the eye of beholder: Joint learning of gaze and actions in first person video. In *ECCV*, 2018.

[183] Yingwei Li, Yi Li, and Nuno Vasconcelos. RESOUND: Towards action recognition without representation bias. In *ECCV*, 2018.

[184] Yixuan Li, Lei Chen, Runyu He, Zhenzhi Wang, Gangshan Wu, and Limin Wang. Multisports: A multi-person video dataset of spatio-temporally localized sports actions. In *ICCV*, pages 13536–13545, 2021.

[185] Yong-Lu Li, Yue Xu, Xiaohan Mao, and Cewu Lu. Symmetry and group in attribute-object compositions. In *CVPR*, 2020.

[186] Yun Li, Zhe Liu, Hang Chen, and Lina Yao. Context-based and diversity-driven specificity in compositional zero-shot learning. In *CVPR*, 2024.

[187] Zhengqi Li, Tali Dekel, Forrester Cole, Richard Tucker, Noah Snavely, Ce Liu, and William T Freeman. Learning the depths of moving people by watching frozen people. In *CVPR*, 2019.

[188] Feng Liang, Bichen Wu, Xiaoliang Dai, Kunpeng Li, Yinan Zhao, Hang Zhang, Peizhao Zhang, Peter Vajda, and Diana Marculescu. Open-vocabulary semantic segmentation with mask-adapted clip. In *CVPR*, 2023.

[189] John Liechty, Rik Pieters, and Michel Wedel. Global and local covert visual attention: Evidence from a bayesian hidden markov model. *Psychometrika*, 68(4):519–541, 2003.

[190] Bill Yuchen Lin, Wangchunshu Zhou, Ming Shen, Pei Zhou, Chandra Bhagavatula, Yejin Choi, and Xiang Ren. CommonGen: A constrained text generation challenge for generative commonsense reasoning. In *EMNLP*, 2020.

[191] Chuang Lin, Peize Sun, Yi Jiang, Ping Luo, Lizhen Qu, Gholamreza Haffari, Zehuan Yuan, and Jianfei Cai. Learning object-language alignments for open-vocabulary object detection. In *ICLR*, 2022.

[192] Chuming Lin, Chengming Xu, Donghao Luo, Yabiao Wang, Ying Tai, Chengjie Wang, Jilin Li, Feiyue Huang, and Yanwei Fu. Learning salient boundary feature for anchor-free temporal action localization. In *CVPR*, 2021.

[193] Ji Lin, Chuang Gan, and Song Han. TSM: Temporal shift module for efficient video

understanding. In *ICCV*, 2019.

[194] Tianwei Lin, Xiao Liu, Xin Li, Errui Ding, and Shilei Wen. Bmn: Boundary-matching network for temporal action proposal generation. In *ICCV*, 2019.

[195] Tianwei Lin, Xu Zhao, Haisheng Su, Chongjing Wang, and Ming Yang. Bsn: Boundary sensitive network for temporal action proposal generation. In *ECCV*, 2018.

[196] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[197] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, 2017.

[198] Ziyi Lin, Shijie Geng, Renrui Zhang, Peng Gao, Gerard de Melo, Xiaogang Wang, Jifeng Dai, Yu Qiao, and Hongsheng Li. Frozen clip models are efficient video learners. In *ECCV*, pages 388–404, 2022.

[199] Zachary C Lipton, John Berkowitz, and Charles Elkan. A critical review of recurrent neural networks for sequence learning. *arXiv:1506.00019*, 2015.

[200] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. *arXiv preprint arXiv:2310.03744*, 2023.

[201] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *NeurIPS*, 2023.

[202] Miao Liu, Lingni Ma, Kiran Somasundaram, Yin Li, Kristen Grauman, James M Rehg, and Chao Li. Egocentric activity recognition and localization on a 3d map. In *ECCV*, 2022.

[203] Miao Liu, Siyu Tang, Yin Li, and James M Rehg. Forecasting human-object interaction: joint prediction of motor attention and actions in first person video. In *ECCV*, 2020.

[204] Ruyang Liu, Jingjia Huang, Ge Li, Jiashi Feng, Xinglong Wu, and Thomas H Li. Revisiting temporal modeling for clip-based image-to-video knowledge transferring. In *CVPR*, 2023.

[205] Shaowei Liu, Subarna Tripathi, Somdeb Majumdar, and Xiaolong Wang. Joint hand motion and interaction hotspots prediction from egocentric videos. In *CVPR*, 2022.

[206] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. *arXiv preprint arXiv:2303.05499*, 2023.

[207] Weiyang Liu, Yandong Wen, Zhiding Yu, Ming Li, Bhiksha Raj, and Le Song. Sphereface:

Deep hypersphere embedding for face recognition. In *CVPR*, 2017.

[208] Xinyang Liu, Dongsheng Wang, Miaoge Li, Zhibin Duan, Yishi Xu, Bo Chen, and Mingyuan Zhou. Patch-token aligned bayesian prompt learning for vision-language models. *arXiv preprint arXiv:2303.09100*, 2023.

[209] Zhe Liu, Yun Li, Lina Yao, Xiaojun Chang, Wei Fang, Xiaojun Wu, and Yi Yang. Simple primitives with feasibility-and contextuality-dependence for open-world compositional zero-shot learning. *arXiv preprint arXiv:2211.02895*, 2022.

[210] Fuchen Long, Ting Yao, Zhaofan Qiu, Xinmei Tian, Jiebo Luo, and Tao Mei. Gaussian temporal awareness networks for action localization. In *CVPR*, 2019.

[211] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2018.

[212] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60:91–110, 2004.

[213] Cewu Lu, Ranjay Krishna, Michael Bernstein, and Li Fei-Fei. Visual relationship detection with language priors. In *ECCV*, 2016.

[214] Xiaocheng Lu, Ziming Liu, Song Guo, and Jingcai Guo. Decomposed soft prompt guided fusion enhancing for compositional zero-shot learning. In *CVPR*, 2023.

[215] Yuning Lu, Jianzhuang Liu, Yonggang Zhang, Yajing Liu, and Xinmei Tian. Prompt distribution learning. In *CVPR*, 2022.

[216] Shugao Ma, Leonid Sigal, and Stan Sclaroff. Learning activity progression in LSTMs for activity detection and early detection. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[217] Zixian Ma, Jerry Hong, Mustafa Omer Gul, Mona Gandhi, Irena Gao, and Ranjay Krishna. Crepe: Can vision-language foundation models reason compositionally? In *CVPR*, 2023.

[218] Andrey Malinin and Mark Gales. Predictive uncertainty estimation via prior networks. In *NeurIPS*, 2018.

[219] Srikanth Malla, Isht Dwivedi, Behzad Dariush, and Chiho Choi. Nemo: Future object localization using noisy ego priors. In *ITSC*, 2022.

[220] Massimiliano Mancini, Muhammad Ferjad Naeem, Yongqin Xian, and Zeynep Akata. Open world compositional zero-shot learning. In *CVPR*, 2021.

[221] Devraj Mandal, Sanath Narayan, Sai Kumar Dwivedi, Vikram Gupta, Shuaib Ahmed, Fahad Shahbaz Khan, and Ling Shao. Out-of-distribution detection for generalized zero-shot

action recognition. In *CVPR*, 2019.

[222] Mayug Maniparambil, Chris Vorster, Derek Molloy, Noel Murphy, Kevin McGuinness, and Noel E O'Connor. Enhancing clip with gpt-4: Harnessing visual descriptions as prompts. *arXiv preprint arXiv:2307.11661*, 2023.

[223] Sachit Menon and Carl Vondrick. Visual classification via description from large language models. In *ICLR*, 2023.

[224] Dimity Miller, Lachlan Nicholson, Feras Dayoub, and Niko Sünderhauf. Dropout sampling for robust object detection in open-set conditions. In *ICRA*, 2018.

[225] Kyle Min and Jason J. Corso. TASED-Net: Temporally-aggregating spatial encoder-decoder network for video saliency detection. In *ICCV*, 2019.

[226] Ishan Misra, Abhinav Gupta, and Martial Hebert. From red wine to red tomato: Composition with context. In *CVPR*, 2017.

[227] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *ICML*, 2016.

[228] Volodymyr Mnih, Nicolas Heess, Alex Graves, et al. Recurrent models of visual attention. In *NeurIPS*, 2014.

[229] Liliane Momeni, Mathilde Caron, Arsha Nagrani, Andrew Zisserman, and Cordelia Schmid. Verbs in action: Improving verb understanding in video-language models. In *ICCV*, pages 15579–15591, 2023.

[230] Mathew Monfort, Kandan Ramakrishnan, Alex Andonian, Barry A. McNamara, Alex Lascelles, Bowen Pan, Quanfu Fan, Dan Gutfreund, Rogério Schmidt Feris, and Aude Oliva. Multi-Moments in Time: Learning and interpreting models for multi-action video understanding. *CoRR*, abs/1911.00232, 2019.

[231] Jishnu Mukhoti and Yarin Gal. Evaluating bayesian deep learning methods for semantic segmentation. *arXiv preprint arXiv:1811.12709*, 2018.

[232] Jishnu Mukhoti, Viveka Kulharia, Amartya Sanyal, Stuart Golodetz, Philip HS Torr, and Puneet K Dokania. Calibrating deep neural networks using focal loss. In *NeurIPS*, 2020.

[233] Martin Mundt, Iuliia Pliushch, Sagnik Majumder, and Visvanathan Ramesh. Open set recognition through deep neural network uncertainty: Does out-of-distribution detection require generative classifiers? In *ICCVW*, 2019.

[234] Muhammad Ferjad Naeem, Yongqin Xian, Federico Tombari, and Zeynep Akata. Learning

graph embeddings for compositional zero-shot learning. In *CVPR*, 2021.

[235] Mahdi Pakdaman Naeini, Gregory Cooper, and Milos Hauskrecht. Obtaining well calibrated probabilities using bayesian binning. In *AAAI*, 2015.

[236] Sauradip Nag, Xiatian Zhu, Yi-Zhe Song, and Tao Xiang. Zero-shot temporal action detection via vision-language prompting. In *ECCV*, 2022.

[237] Tushar Nagarajan and Kristen Grauman. Attributes as operators: factorizing unseen attribute-object compositions. In *ECCV*, 2018.

[238] Tushar Nagarajan, Yanghao Li, Christoph Feichtenhofer, and Kristen Grauman. Ego-topo: Environment affordances from egocentric video. In *CVPR*, 2020.

[239] Nihal V Nayak, Peilin Yu, and Stephen H Bach. Learning to compose soft prompts for compositional zero-shot learning. In *ICLR*, 2023.

[240] Lawrence Neal, Matthew Olson, Xiaoli Fern, Weng-Keen Wong, and Fuxin Li. Open set learning with counterfactual images. In *ECCV*, 2018.

[241] Radford M Neal. *Bayesian learning for neural networks*, volume 118. Springer Science & Business Media, 2012.

[242] Lukas Neumann, Andrew Zisserman, and Andrea Vedaldi. Future event prediction: If and when. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (Workshop)*, 2019.

[243] Bolin Ni, Houwen Peng, Minghao Chen, Songyang Zhang, Gaofeng Meng, Jianlong Fu, Shiming Xiang, and Haibin Ling. Expanding language-image pretrained models for general video recognition. In *ECCV*, pages 1–18, 2022.

[244] Adrián Núñez-Marcos, Gorka Azkune, and Ignacio Arganda-Carreras. Egocentric vision-based action recognition: a survey. *Neurocomputing*, 472:175–197, 2022.

[245] Dimitri Ognibene, Christian Balkenius, and Gianluca Baldassarre. A reinforcement-learning model of top-down attention based on a potential-action map. In *The Challenge of Anticipation: A Unifying Framework for the Analysis and Design of Artificial Cognitive Systems*, pages 161–184. Springer, 2008.

[246] Hugo Oliveira, Caio Silva, Gabriel LS Machado, Keiller Nogueira, and Jefersson A dos Santos. Fully convolutional open set segmentation. *Machine Learning*, pages 1–52, 2021.

[247] OpenAI. OpenAI GPT-3.5 API [gpt-3.5-turbo-0125]. https://openai.com/blog/chatgpt. Accessed: 2023.

[248] OpenAI. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[249] Poojan Oza and Vishal M Patel. C2AE: Class conditioned auto-encoder for open-set recognition. In *CVPR*, 2019.

[250] Junting Pan, Siyu Chen, Mike Zheng Shou, Yu Liu, Jing Shao, and Hongsheng Li. Actor-context-actor relation network for spatio-temporal action localization. In *CVPR*, pages 464–474, 2021.

[251] Junting Pan, Ziyi Lin, Xiatian Zhu, Jing Shao, and Hongsheng Li. St-adapter: Parameter-efficient image-to-video transfer learning. In *NeurIPS*, pages 26462–26477, 2022.

[252] Guansong Pang, Chunhua Shen, Longbing Cao, and Anton van den Hengel. Deep learning for anomaly detection: A review. *arXiv preprint arXiv:2007.02500*, 2020.

[253] Paschalis Panteleris, Iason Oikonomidis, and Antonis Argyros. Using a single rgb frame for real time 3d hand pose estimation in the wild. In *WACV*, 2018.

[254] Seulki Park, Jongin Lim, Younghan Jeon, and Jin Young Choi. Influence-balanced loss for imbalanced visual classification. In *ICCV*, 2021.

[255] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An imperative style, high-performance deep learning library. In *Proceedings of Neural Information Processing Systems*, 2019.

[256] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *EMNLP*, 2014.

[257] Pramuditha Perera, Vlad I Morariu, Rajiv Jain, Varun Manjunatha, Curtis Wigington, Vicente Ordonez, and Vishal M Patel. Generative-discriminative feature representations for open-set recognition. In *CVPR*, 2020.

[258] Trung Pham, Thanh-Toan Do, Gustavo Carneiro, Ian Reid, et al. Bayesian semantic instance segmentation in open set world. In *ECCV*, 2018.

[259] Senthil Purushwalkam, Maximilian Nickel, Abhinav Gupta, and Marc'Aurelio Ranzato. Task-driven modular networks for zero-shot compositional learning. In *ICCV*, 2019.

[260] Jianing Qiu, Lipeng Chen, Xiao Gu, Frank P-W Lo, Ya-Yen Tsai, Jiankai Sun, Jiaqi Liu, and Benny Lo. Egocentric human trajectory forecasting with a wearable camera and multi-modal fusion. *IEEE Robotics and Automation Letters*, 7(4):8799–8806, 2022.

[261] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021.

[262] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *JMLR*, 21(1):5485–5551, 2020.

[263] Syama Sundar Rangapuram, Matthias W Seeger, Jan Gasthaus, Lorenzo Stella, Yuyang Wang, and Tim Januschowski. Deep state space models for time series forecasting. In *NeurIPS*, 2018.

[264] Hanoona Rasheed, Muhammad Uzair Khattak, Muhammad Maaz, Salman Khan, and Fahad Shahbaz Khan. Fine-tuned clip models are efficient video learners. In *CVPR*, 2023.

[265] Vivek Rathod, Bryan Seybold, Sudheendra Vijayanarasimhan, Austin Myers, Xiuye Gu, Vighnesh Birodkar, and David A Ross. Open-vocabulary temporal action detection with off-the-shelf image-text features. In *BMVC*, 2022.

[266] Anastasia Razdaibiedina, Yuning Mao, Rui Hou, Madian Khabsa, Mike Lewis, Jimmy Ba, and Amjad Almahairi. Residual prompt tuning: Improving prompt tuning with residual reparameterization. In *ACL*, 2023.

[267] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Proceedings of Neural Information Processing Systems*, 2015.

[268] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE TPAMI*, 39(6):1137–1149, 2017.

[269] Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. Generalized intersection over union: A metric and a loss for bounding box regression. In *CVPR*, pages 658–666, 2019.

[270] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *International Conference on Machine Learning*, 2014.

[271] Byungseok Roh, JaeWoong Shin, Wuhyun Shin, and Saehoon Kim. Sparse DETR: Efficient end-to-end object detection with learnable sparsity. In *ICLR*, 2022.

[272] Alina Roitberg, Chaoxiang Ma, Monica Haurilet, and Rainer Stiefelhagen. Open set driver activity recognition. In *IVS*, 2020.

[273] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Om-

mer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

[274] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115:211–252, 2015.

[275] Chaitanya Ryali, Yuan-Ting Hu, Daniel Bolya, Chen Wei, Haoqi Fan, Po-Yao Huang, Vaibhav Aggarwal, Arkabandhu Chowdhury, Omid Poursaeed, Judy Hoffman, et al. Hiera: A hierarchical vision transformer without the bells-and-whistles. In *ICML*, 2023.

[276] Kuniaki Saito, Donghyun Kim, and Kate Saenko. Openmatch: Open-set semi-supervised learning with open-set consistency regularization. In *NeurIPS*, 2021.

[277] D. M. Saxena, S. Bae, A. Nakhaei, K. Fujimura, and M. Likhachev. Driving in dense traffic with model-free reinforcement learning. In *ICRA*, 2020.

[278] Walter J Scheirer, Anderson de Rezende Rocha, Archana Sapkota, and Terrance E Boult. Toward open set recognition. *IEEE transactions on pattern analysis and machine intelligence*, 35(7):1757–1772, 2012.

[279] Walter J Scheirer, Lalit P Jain, and Terrance E Boult. Probability models for open set recognition. *IEEE transactions on pattern analysis and machine intelligence*, 36(11):2317–2324, 2014.

[280] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[281] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *ICCV*, pages 618–626, 2017.

[282] Murat Sensoy, Lance Kaplan, Federico Cerutti, and Maryam Saleki. Uncertainty-aware deep classifiers using generative models. In *AAAI*, 2020.

[283] Murat Sensoy, Lance Kaplan, and Melih Kandemir. Evidential deep learning to quantify classification uncertainty. In *NeurIPS*, 2018.

[284] Kari Sentz, Scott Ferson, et al. *Combination of evidence in Dempster-Shafer theory*, volume 4015. Sandia National Laboratories Albuquerque, 2002.

[285] Youngjoo Seo, Michaël Defferrard, Pierre Vandergheynst, and Xavier Bresson. Structured sequence modeling with graph convolutional recurrent networks. In *Proceedings of Neural Information Processing Systems*, 2018.

[286] Ankit Shah, Jean Baptiste Lamare, Tuan Nguyen Anh, and Alexander Hauptmann. CADP: A novel dataset for CCTV traffic camera based accident analysis. In *International Workshop on Traffic and Street Surveillance for Safety and Security*, 2018.

[287] Weishi Shi, Xujiang Zhao, Feng Chen, and Qi Yu. Multifaceted uncertainty estimation for label-efficient deep learning. In *NeurIPS*, 2020.

[288] Zheng Shou, Jonathan Chan, Alireza Zareian, Kazuyuki Miyazawa, and Shih-Fu Chang. Cdc: Convolutional-de-convolutional networks for precise temporal action localization in untrimmed videos. In *CVPR*, 2017.

[289] Kumar Shridhar, Felix Laumann, and Marcus Liwicki. Uncertainty estimations by softplus normalization in bayesian convolutional neural networks with variational inference. *arXiv:1806.05978*, 2018.

[290] Yu Shu, Yemin Shi, Yaowei Wang, Tiejun Huang, and Yonghong Tian. p-odn: prototype-based open deep network for open set recognition. *Scientific reports*, 10(1):1–13, 2020.

[291] Yu Shu, Yemin Shi, Yaowei Wang, Yixiong Zou, Qingsheng Yuan, and Yonghong Tian. ODN: Opening the deep network for open-set action recognition. In *ICME*, 2018.

[292] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[293] Gurkirt Singh, Stephen Akrigg, Manuele Di Maio, Valentina Fontana, Reza Javanmard Alitappeh, Salman Khan, Suman Saha, Kossar Jeddisaravi, Farzad Yousefi, Jacob Culley, Tom Nicholson, Jordan Omokeowa, Stanislao Grazioso, Andrew Bradley, Giuseppe Di Gironimo, and Fabio Cuzzolin. Road: The road event awareness dataset for autonomous driving. *IEEE TPAMI*, 45(1):1036–1054, 2023.

[294] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

[295] Le Song, Alex Smola, Arthur Gretton, Justin Bedo, and Karsten Borgwardt. Feature selection via dependence maximization. *Journal of Machine Learning Research*, 13(5), 2012.

[296] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. UCF101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012.

[297] Deepak Sridhar, Niamul Quader, Srikanth Muralidharan, Yaoxin Li, Peng Dai, and Juwei Lu. Class semantics-based attention for action detection. In *ICCV*, 2021.

[298] Mahesh Subedar, Ranganath Krishnan, Paulo Lopez Meyer, Omesh Tickoo, and Jonathan Huang. Uncertainty-aware audiovisual activity recognition using deep bayesian variational inference. In *ICCV*, 2019.

[299] Chen Sun, Abhinav Shrivastava, Carl Vondrick, Kevin Murphy, Rahul Sukthankar, and Cordelia Schmid. Actor-centric relation network. In *ECCV*, pages 318–334, 2018.

[300] Chen Sun, Abhinav Shrivastava, Carl Vondrick, Rahul Sukthankar, Kevin Murphy, and Cordelia Schmid. Relational action forecasting. In *CVPR*, 2019.

[301] Peize Sun, Rufeng Zhang, Yi Jiang, Tao Kong, Chenfeng Xu, Wei Zhan, Masayoshi Tomizuka, Lei Li, Zehuan Yuan, Changhu Wang, et al. Sparse r-cnn: End-to-end object detection with learnable proposals. In *CVPR*, pages 14454–14463, 2021.

[302] Xin Sun, Zhenning Yang, Chi Zhang, Keck-Voon Ling, and Guohao Peng. Conditional Gaussian distribution learning for open set recognition. In *CVPR*, 2020.

[303] Tomoyuki Suzuki, Hirokatsu Kataoka, Yoshimitsu Aoki, and Yutaka Satoh. Anticipating traffic accidents with adaptive loss and large-scale incident DB. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[304] Yoshiaki Takimoto, Yusuke Tanaka, Takeshi Kurashima, Shuhei Yamamoto, Maya Okawa, and Hiroyuki Toda. Predicting traffic accidents with event recorder data. In *Proceedings of ACM SIGSPATIAL International Workshop on Prediction of Human Mobility*, 2019.

[305] Binh Tang and David S Matteson. Probabilistic transformer for time series analysis. In *NeurIPS*, 2021.

[306] Jiajun Tang, Jin Xia, Xinzhi Mu, Bo Pang, and Cewu Lu. Asynchronous interaction aggregation for action detection. In *ECCV*, pages 71–87, 2020.

[307] Chameleon Team. Chameleon: Mixed-modal early-fusion foundation models. *arXiv preprint arXiv:2405.09818*, 2024.

[308] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *ECCV*, 2020.

[309] Bugra Tekin, Federica Bogo, and Marc Pollefeys. H+o: Unified egocentric recognition of 3d hand-object poses and interactions. In *CVPR*, 2019.

[310] **Wentao Bao**, Lele Chen, Libing Zeng, Zhong Li, Yi Xu, Junsong Yuan, and Yu Kong. Uncertainty-aware state space transformer for egocentric 3d hand trajectory forecasting. In *International Conference on Computer Vision (ICCV)*, 2023.

[311] **Wentao Bao**, Lichang Chen, Heng Huang, and Yu Kong. Prompting language-informed distribution for compositional zero-shot learning. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2024.

[312] **Wentao Bao**, Kai Li, Yuxiao Chen, Deep Patel, Martin Renqiang Min, and Yu Kong.

Exploiting vlm localizability and semantics for open vocabulary action detection. *arXiv preprint*, 2024.

[313] **Wentao Bao**, Bin Xu, and Zhenzhong Chen. MonoFENet: Monocular 3d object detection with feature enhancement networks. *IEEE Transactions on Image Processing (TIP)*, 29:2753–2765, November 2019.

[314] **Wentao Bao**, Qi Yu, and Yu Kong. Object-aware centroid voting for monocular 3d object detection. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020.

[315] **Wentao Bao**, Qi Yu, and Yu Kong. Uncertainty-based traffic accident anticipation with spatio-temporal relational learning. In *Proceedings of the 28th ACM International Conference on Multimedia (ACM MM)*, 2020.

[316] **Wentao Bao**, Qi Yu, and Yu Kong. DRIVE: Deep reinforced accident anticipation with visual explanation. In *International Conference on Computer Vision (ICCV)*, 2021.

[317] **Wentao Bao**, Qi Yu, and Yu Kong. Evidential deep learning for open set action recognition. In *International Conference on Computer Vision (ICCV), **Oral***, 2021.

[318] **Wentao Bao**, Qi Yu, and Yu Kong. OpenTAL: Towards open set temporal action localization. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), **Oral***, 2022.

[319] **Wentao Bao**, Qi Yu, and Yu Kong. Latent space energy-based model for fine-grained open set recognition. *arXiv preprint arXiv:2309.10711*, 2023.

[320] Luca Anthony Thiede and Pratik Prabhanjan Brahma. Analyzing the variety loss in the context of probabilistic trajectory prediction. In *CVPR*, 2019.

[321] Pavel Tokmakov, Yu-Xiong Wang, and Martial Hebert. Learning compositional representations for few-shot recognition. In *ICCV*, 2019.

[322] Ilya O Tolstikhin, Neil Houlsby, Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Thomas Unterthiner, Jessica Yung, Andreas Steiner, Daniel Keysers, Jakob Uszkoreit, et al. Mlp-mixer: An all-mlp architecture for vision. In *NeurIPS*, pages 24261–24272, 2021.

[323] Shengbang Tong, Zhuang Liu, Yuexiang Zhai, Yi Ma, Yann LeCun, and Saining Xie. Eyes wide shut? exploring the visual shortcomings of multimodal llms. *arXiv preprint arXiv:2401.06209*, 2024.

[324] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[325] Matthew Trager, Pramuditha Perera, Luca Zancato, Alessandro Achille, Parminder Bhatia, Bing Xiang, and Stefano Soatto. Linear spaces of meanings: the compositional language of vlms. *arXiv preprint arXiv:2302.14383*, 2023.

[326] Dustin Tran, Jasper Snoek, and Balaji Lakshminarayanan. Practical uncertainty estimation and out-of-distribution robustness in deep learning. Technical report, Google Brain, 2020. NeurIPS Tutorial.

[327] Vladimir Naumovich Vapnik, Vlamimir Vapnik, et al. *Statistical learning theory*. wiley New York, 1998.

[328] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of Neural Information Processing Systems*, 2017.

[329] Trent Victor. *Keeping eye and mind on the road*. PhD thesis, Acta Universitatis Upsaliensis, 2005.

[330] Henan Wang, Muli Yang, Kun Wei, and Cheng Deng. Hierarchical prompt learning for compositional zero-shot recognition. In *IJCAI*, 2023.

[331] Jiashun Wang, Huazhe Xu, Medhini Narasimhan, and Xiaolong Wang. Multi-person 3d motion prediction with multi-range transformers. In *NeurIPS*, 2021.

[332] Mengmeng Wang, Jiazheng Xing, and Yong Liu. Actionclip: A new paradigm for video action recognition. *arXiv preprint arXiv:2109.08472*, 2021.

[333] Weiyao Wang, Matt Feiszli, Heng Wang, and Du Tran. Unidentified video objects: A benchmark for dense, open-world segmentation. In *ICCV*, 2021.

[334] Xiaohan Wang, Yu Wu, Linchao Zhu, and Yi Yang. Symbiotic attention with privileged information for egocentric action recognition. In *AAAI*, 2020.

[335] Yezhen Wang, Bo Li, Tong Che, Kaiyang Zhou, Ziwei Liu, and Dongsheng Li. Energy-based open-world uncertainty modeling for confidence calibration. In *ICCV*, 2021.

[336] Yi Wang, Kunchang Li, Yizhuo Li, Yinan He, Bingkun Huang, Zhiyu Zhao, Hongjie Zhang, Jilan Xu, Yi Liu, Zun Wang, et al. Internvideo: General video foundation models via generative and discriminative learning. *arXiv preprint arXiv:2212.03191*, 2022.

[337] Yufei Wang and Tianwei Ni. Meta-SAC: Auto-tune the entropy temperature of soft actor-critic via metagradient. In *ICML Workshop*, 2020.

[338] Syed Talal Wasim, Muzammal Naseer, Salman Khan, Fahad Shahbaz Khan, and Mubarak Shah. Vita-clip: Video and text adaptive clip via multimodal prompting. In *CVPR*, pages

23034–23044, 2023.

[339] Philippe Weinzaepfel and Grégory Rogez. Mimetics: Towards understanding human actions out of context. *International Journal of Computer Vision*, pages 1–16, 2021.

[340] Zejia Weng, Xitong Yang, Ang Li, Zuxuan Wu, and Yu-Gang Jiang. Open-vclip: Transforming clip to an open-vocabulary video model via interpolated weight optimization. In *ICML*, 2023.

[341] Jürgen Wiest, Matthias Höffken, Ulrich Kreßel, and Klaus Dietmayer. Probabilistic trajectory prediction with gaussian mixture models. In *IVS*, 2012.

[342] Max Wolff, Wieland Brendel, and Stuart Wolff. The independent compositional subspace hypothesis for the structure of clip's last layer. In *ICLR Workshop*, 2023.

[343] Chao-Yuan Wu, Christoph Feichtenhofer, Haoqi Fan, Kaiming He, Philipp Krahenbuhl, and Ross Girshick. Long-term feature banks for detailed video understanding. In *CVPR*, 2019.

[344] Di Wu, Nabin Sharma, and Michael Blumenstein. Recent advances in video-based human action recognition using deep learning: A review. In *IJCNN*, 2017.

[345] Jianchao Wu, Zhanghui Kuang, Limin Wang, Wayne Zhang, and Gangshan Wu. Context-aware rcnn: A baseline for action detection in videos. In *ECCV*, pages 440–456, 2020.

[346] Jianzong Wu, Xiangtai Li, Shilin Xu Haobo Yuan, Henghui Ding, Yibo Yang, Xia Li, Jiangning Zhang, Yunhai Tong, Xudong Jiang, Bernard Ghanem, et al. Towards open vocabulary learning: A survey. *arXiv preprint arXiv:2306.15880*, 2023.

[347] Tao Wu, Mengqi Cao, Ziteng Gao, Gangshan Wu, and Limin Wang. Stmixer: A one-stage sparse action detector. In *CVPR*, pages 14720–14729, 2023.

[348] Xiaoshi Wu, Feng Zhu, Rui Zhao, and Hongsheng Li. Cora: Adapting clip for open-vocabulary detection with region prompting and anchor pre-matching. In *CVPR*, 2023.

[349] Ye Xia, Jinkyu Kim, John Canny, Karl Zipser, Teresa Canas-Bajo, and David Whitney. Periphery-fovea multi-resolution driving model guided by human attention. In *WACV*, 2020.

[350] Ye Xia, Danqing Zhang, Jinkyu Kim, Ken Nakayama, Karl Zipser, and David Whitney. Predicting driver attention in critical situations. In *ACCV*, 2018.

[351] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[352] Guangyue Xu, Parisa Kordjamshidi, and Joyce Chai. Prompting large pre-trained vision-

language models for compositional concept learning. *arXiv preprint arXiv:2211.05077*, 2022.

[353] Hu Xu, Bing Liu, Lei Shu, and P Yu. Open-world learning and application to product classification. In *WWW*, 2019.

[354] Huijuan Xu, Abir Das, and Kate Saenko. R-c3d: Region convolutional 3d network for temporal activity detection. In *ICCV*, 2017.

[355] Mai Xu, Yuhang Song, Jianyi Wang, MingLang Qiao, Liangyu Huo, and Zulin Wang. Predicting head movement in panoramic video: A deep reinforcement learning approach. *IEEE TPAMI*, 41(11):2693–2708, 2018.

[356] Mengmeng Xu, Chen Zhao, David S Rojas, Ali Thabet, and Bernard Ghanem. G-tad: Sub-graph localization for temporal action detection. In *CVPR*, 2020.

[357] Hongwei Xue, Yuchong Sun, Bei Liu, Jianlong Fu, Ruihua Song, Houqiang Li, and Jiebo Luo. Clip-vip: Adapting pre-trained image-text model to video-language representation alignment. In *ICLR*, 2022.

[358] An Yan, Yu Wang, Yiwu Zhong, Chengyu Dong, Zexue He, Yujie Lu, William Wang, Jingbo Shang, and Julian McAuley. Learning concise and descriptive attributes for visual recognition. *arXiv preprint arXiv:2308.03685*, 2023.

[359] Shen Yan, Xuehan Xiong, Arsha Nagrani, Anurag Arnab, Zhonghao Wang, Weina Ge, David Ross, and Cordelia Schmid. Unloc: A unified framework for video localization tasks. In *ICCV*, 2023.

[360] Ceyuan Yang, Yinghao Xu, Jianping Shi, Bo Dai, and Bolei Zhou. Temporal pyramid network for action recognition. In *CVPR*, 2020.

[361] Hong-Ming Yang, Xu-Yao Zhang, Fei Yin, and Cheng-Lin Liu. Robust classification with convolutional prototype learning. In *CVPR*, 2018.

[362] Hong-Ming Yang, Xu-Yao Zhang, Fei Yin, Qing Yang, and Cheng-Lin Liu. Convolutional prototype network for open set recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[363] Jingkang Yang, Pengyun Wang, Dejian Zou, Zitang Zhou, Kunyuan Ding, Wenxuan Peng, Haoqi Wang, Guangyao Chen, Bo Li, Yiyou Sun, et al. Openood: Benchmarking generalized out-of-distribution detection. In *NeurIPS*, 2022.

[364] Mengyue Yang, Furui Liu, Zhitang Chen, Xinwei Shen, Jianye Hao, and Jun Wang. Causalvae: Disentangled representation learning via neural structural causal models. In *CVPR*, 2021.

[365] Ming Yang, Shuiwang Ji, Wei Xu, Jinjun Wang, Fengjun Lv, Kai Yu, Yihong Gong, Mert Dikmen, Dennis J Lin, and Thomas S Huang. Detecting human actions in surveillance videos. In *TRECVID*, 2009.

[366] Yang Yang, Chunping Hou, Yue Lang, Dai Guan, Danyang Huang, and Jinchen Xu. Open-set human activity recognition based on micro-doppler signatures. *Pattern Recognition*, 85:60–69, 2019.

[367] Zhibo Yang, Lihan Huang, Yupei Chen, Zijun Wei, Seoyoung Ahn, Gregory Zelinsky, Dimitris Samaras, and Minh Hoai. Predicting goal-directed human attention using inverse reinforcement learning. In *CVPR*, 2020.

[368] Yu Yao, Mingze Xu, Yuchen Wang, David J Crandall, and Ella M Atkins. Unsupervised traffic accident detection in first-person videos. In *International Conference on Intelligent Robots and Systems*, 2019.

[369] Denis Yarats, Amy Zhang, Ilya Kostrikov, Brandon Amos, Joelle Pineau, and Rob Fergus. Improving sample efficiency in model-free reinforcement learning from images. *arXiv preprint arXiv:1910.01741*, 2019.

[370] Serena Yeung, Olga Russakovsky, Greg Mori, and Li Fei-Fei. End-to-end learning of action detection from frame glimpses in videos. In *CVPR*, 2016.

[371] Li Yingzhen and Stephan Mandt. Disentangled sequential autoencoder. In *ICML*, 2018.

[372] Ryota Yoshihashi, Wen Shao, Rei Kawakami, Shaodi You, Makoto Iida, and Takeshi Naemura. Classification-reconstruction learning for open-set recognition. In *CVPR*, 2019.

[373] Tackgeun You and Bohyung Han. Traffic accident benchmark for causality recognition. In *ECCV*, 2020.

[374] Aron Yu and Kristen Grauman. Fine-grained visual comparisons with local learning. In *CVPR*, 2014.

[375] Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan, and Trevor Darrell. BDD100K: A diverse driving dataset for heterogeneous multitask learning. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

[376] Qing Yu, Daiki Ikami, Go Irie, and Kiyoharu Aizawa. Multi-task curriculum framework for open-set semi-supervised learning. In *ECCV*, 2020.

[377] Liangzhe Yuan, Rui Qian, Yin Cui, Boqing Gong, Florian Schroff, Ming-Hsuan Yang, Hartwig Adam, and Ting Liu. Contextualized spatio-temporal contrastive learning with self-supervision. In *CVPR*, pages 13977–13986, 2022.

[378] Weihao Yuan, Xiaodong Gu, Zuozhuo Dai, Siyu Zhu, and Ping Tan. NeW CRFs: Neural window fully-connected crfs for monocular depth estimation. In *CVPR*, 2022.

[379] Ye Yuan and Kris Kitani. Ego-pose estimation and forecasting as real-time PD control. In *ICCV*, 2019.

[380] Ye Yuan, Xinshuo Weng, Yanglan Ou, and Kris M Kitani. Agentformer: Agent-aware transformers for socio-temporal multi-agent forecasting. In *ICCV*, 2021.

[381] Zhongqi Yue, Tan Wang, Qianru Sun, Xian-Sheng Hua, and Hanwang Zhang. Counterfactual zero-shot and open-set visual recognition. In *CVPR*, 2021.

[382] Mert Yuksekgonul, Federico Bianchi, Pratyusha Kalluri, Dan Jurafsky, and James Zou. When and why vision-language models behave like bags-of-words, and what to do about it? In *ICLR*, 2023.

[383] Kimin Yun, Yongjin Kwon, Sungchan Oh, Jinyoung Moon, and Jongyoul Park. Vision-based garbage dumping action detection for real-world surveillance platform. *ETRI Journal*, 41(4):494–505, 2019.

[384] Yuhang Zang, Wei Li, Kaiyang Zhou, Chen Huang, and Chen Change Loy. Open-vocabulary detr with conditional matching. In *ECCV*, pages 106–122, 2022.

[385] Alireza Zareian, Kevin Dela Rosa, Derek Hao Hu, and Shih-Fu Chang. Open-vocabulary object detection using captions. In *CVPR*, pages 14393–14402, 2021.

[386] Rowan Zellers, Yonatan Bisk, Ali Farhadi, and Yejin Choi. From recognition to cognition: Visual commonsense reasoning. In *CVPR*, 2019.

[387] Kuo-Hao Zeng, Shih-Han Chou, Fu-Hsiang Chan, Juan Carlos Niebles, and Min Sun. Agent-centric risk assessment: Accident anticipation and risky region localization. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[388] Runhao Zeng, Wenbing Huang, Mingkui Tan, Yu Rong, Peilin Zhao, Junzhou Huang, and Chuang Gan. Graph convolutional networks for temporal action localization. In *ICCV*, 2019.

[389] Yuanhao Zhai, Ziyi Liu, Zhenyu Wu, Yi Wu, Chunluan Zhou, David Doermann, Junsong Yuan, and Gang Hua. Soar: Scene-debiasing open-set action recognition. In *ICCV*, 2023.

[390] Hong-Bo Zhang, Yi-Xiang Zhang, Bineng Zhong, Qing Lei, Lijie Yang, Ji-Xiang Du, and Duan-Sheng Chen. A comprehensive survey of vision-based human action recognition methods. *Sensors*, 19(5):1005, 2019.

[391] Hui Zhang and Henghui Ding. Prototypical matching and open set rejection for zero-shot semantic segmentation. In *ICCV*, 2021.

[392] Pengchuan Zhang, Xiujun Li, Xiaowei Hu, Jianwei Yang, Lei Zhang, Lijuan Wang, Yejin Choi, and Jianfeng Gao. Vinvl: Revisiting visual representations in vision-language models. In *CVPR*, 2021.

[393] Pu Zhang, Wanli Ouyang, Pengfei Zhang, Jianru Xue, and Nanning Zheng. Sr-lstm: State refinement for lstm towards pedestrian trajectory prediction. In *CVPR*, 2019.

[394] Ruohan Zhang, Zhuode Liu, Luxin Zhang, Jake A. Whritner, Karl S. Muller, Mary M. Hayhoe, and Dana H. Ballard. AGIL: Learning attention from human for visuomotor tasks. In *ECCV*, 2018.

[395] Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*, 2022.

[396] Tian Zhang, Kongming Liang, Ruoyi Du, Xian Sun, Zhanyu Ma, and Jun Guo. Learning invariant visual representations for compositional zero-shot learning. In *ECCV*, 2022.

[397] Zhoutong Zhang, Forrester Cole, Richard Tucker, William T Freeman, and Tali Dekel. Consistent depth of moving objects in video. *ACM TOG*, 40(4):1–12, 2021.

[398] Jiaojiao Zhao, Yanyi Zhang, Xinyu Li, Hao Chen, Bing Shuai, Mingze Xu, Chunhui Liu, Kaustav Kundu, Yuanjun Xiong, Davide Modolo, et al. Tuber: Tubelet transformer for video action detection. In *CVPR*, pages 13598–13607, 2022.

[399] Peisen Zhao, Lingxi Xie, Chen Ju, Ya Zhang, Yanfeng Wang, and Qi Tian. Bottom-up temporal action localization with mutual regularization. In *ECCV*, 2020.

[400] Rui Zhao, Kang Wang, Hui Su, and Qiang Ji. Bayesian graph convolution lstm for skeleton based action recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, 2019.

[401] Xujiang Zhao, Yuzhe Ou, Lance Kaplan, Feng Chen, and Jin-Hee Cho. Quantifying classification uncertainty using regularized evidential neural networks. In *AAAI*, 2019.

[402] Yue Zhao, Yuanjun Xiong, Limin Wang, Zhirong Wu, Xiaoou Tang, and Dahua Lin. Temporal action detection with structured segment networks. In *ICCV*, 2017.

[403] Yin-Dong Zheng, Guo Chen, Minglei Yuan, and Tong Lu. Mrsn: Multi-relation support network for video action detection. *arXiv preprint arXiv:2304.11975*, 2023.

[404] Zhaoheng Zheng, Haidong Zhu, and Ram Nevatia. Caila: Concept-aware intra-layer adapters for compositional zero-shot learning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1721–1731, 2024.

[405] Yiwu Zhong, Jianwei Yang, Pengchuan Zhang, Chunyuan Li, Noel Codella, Liunian Harold Li, Luowei Zhou, Xiyang Dai, Lu Yuan, Yin Li, et al. Regionclip: Region-based language-image pretraining. In *CVPR*, pages 16793–16803, 2022.

[406] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *CVPR*, pages 2921–2929, 2016.

[407] Da-Wei Zhou, Han-Jia Ye, and De-Chuan Zhan. Learning placeholders for open-set recognition. In *CVPR*, 2021.

[408] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Conditional prompt learning for vision-language models. In *CVPR*, 2022.

[409] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *IJCV*, 2022.

[410] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. Open3D: A modern library for 3D data processing. *arXiv:1801.09847*, 2018.

[411] Ziqin Zhou, Yinjie Lei, Bowen Zhang, Lingqiao Liu, and Yifan Liu. Zegclip: Towards adapting clip for zero-shot semantic segmentation. In *CVPR*, 2023.

[412] Beier Zhu, Yulei Niu, Yucheng Han, Yue Wu, and Hanwang Zhang. Prompt-aligned gradient for prompt tuning. In *ICCV*, pages 15659–15669, 2023.

[413] Zixin Zhu, Wei Tang, Le Wang, Nanning Zheng, and Gang Hua. Enriching local and global contexts for temporal action localization. In *ICCV*, 2021.

[414] Yixiong Zou, Shanghang Zhang, Guangyao Chen, Yonghong Tian, Kurt Keutzer, and José M. F. Moura. Annotation-efficient untrimmed video action recognition. In *ACM MM*, 2021.

[415] Yixiong Zou, Shanghang Zhang, Ke Chen, Yonghong Tian, Yaowei Wang, and José MF Moura. Compositional few-shot recognition with primitive discovery and enhancing. In *ACM MM*, 2020.