

FROM PIXELS TO IDENTITY:  
VISUAL RECOGNITION AND BIOMETRIC APPLICATIONS

By

Minchul Kim

A DISSERTATION

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

Computer Science—Doctor of Philosophy

2025

## ABSTRACT

Automated visual recognition has undergone a transformative evolution, advancing from handcrafted feature extraction to deep learning-driven systems that now permeate modern security, social, and personal computing platforms. Within this rapidly evolving landscape, face and body recognition have emerged as critical tasks—driven by their non-contact nature, scalability, and growing presence in real-world applications. However, achieving robust and generalizable performance in unconstrained settings continues to pose significant challenges, including image degradation, pose misalignment, limited training data, and the complexities of multimodal recognition.

This thesis investigates these challenges through the lens of biometric recognition, leveraging the transformative potential of deep learning and generative artificial intelligence to address both algorithmic and data-centric limitations. It introduces six major contributions. *AdaFace* proposes an adaptive margin loss that prioritizes learning from high-quality samples, improving performance in low-quality image conditions. *CAFace* targets video-based recognition with an attention-based feature aggregation framework optimized for temporal redundancy and long-duration sequences. *DCFace* pioneers synthetic dataset generation using a dual-condition diffusion model, enabling ethical, diverse, and scalable data creation for face recognition. *KPRPE* introduces a keypoint-aware positional encoding scheme that enhances robustness to misalignment and geometric variation. *SapiensID* unifies face and full-body recognition via a multi-resolution transformer trained on the large-scale, multimodal WebBody4M dataset.

Building upon these advances, the thesis concludes with a contribution aimed at real-world deployment: an efficient unified backbone for human recognition. This architecture introduces Keypoint-based Token Fusion (KP-ToFu) and Keypoint Absolute Position Encoding (KP-APE) to reduce computational cost while preserving spatial fidelity and identity-relevant detail. The result is a model that achieves a good performance with significantly lower FLOPs, making unified recognition systems viable for resource-constrained applications.

Together, these contributions form a comprehensive exploration of visual recognition in the deep learning era, highlighting how adaptive loss design, synthetic data generation, positional encoding, and architectural innovations can collectively address longstanding challenges. This thesis lays the foundation for the next generation of intelligent biometric systems—systems that are robust and explainable for deployment in complex, real-world environments.

Copyright by  
MINCHUL KIM  
2025

## ACKNOWLEDGEMENTS

Throughout the course of my PhD journey, I have received immense support, guidance, and encouragement from many incredible individuals, to whom I am deeply grateful.

First and foremost, I would like to express my sincere gratitude to my advisor, Dr. Xiaoming Liu, for his invaluable guidance, unwavering support, and insightful mentorship throughout my PhD. Your expertise and encouragement have shaped both my research and professional growth.

I am also grateful to Dr. Anil Jain for his wisdom, inspiring feedback, and the many thought-provoking discussions that helped refine my work. It has been an honor to learn from you.

To my thesis committee members — Dr. Arun Ross, Dr. Anil Jain, Dr. Daniel Morris, and Dr. Kong Yu — thank you for your time, thoughtful questions, and constructive suggestions. Your perspectives and insights pushed me to broaden the scope and impact of my research.

A special shout-out to Christopher Perry — working with you on the BRIAR project was a true pleasure. Your technical skill, dedication, and collaboration made a tremendous difference, and the project wouldn't have been possible without you.

To my fellow labmates in the Computer Vision Lab — Feng Liu, Shengjie Zhu, Masa Hu, Andrew Hou, Abhinav Kumar, Vishal Asnani, Xiao Guo, Yiyang Su, Jie Zhu, Girish Ganesan, Zeyuan Yin, Zhihao Zhang, Zhiyuan Ren, Zhizhong Huang, Gu Ziang, and Dingqiang Ye — thank you for the stimulating discussions, support, and camaraderie. Our time together, both inside and outside the lab, has been invaluable.

Most importantly, I would like to thank my wife for standing by my side throughout this journey — your patience, love, and belief in me mean everything. And to my son, Gio, who was born during my PhD and continues to grow into a happy, curious little human — your presence gave new meaning and motivation to my work. Finally, thank you to my parents for your unconditional love, sacrifices, and unwavering support — this achievement is as much yours as it is mine.

## TABLE OF CONTENTS

CHAPTER 1	INTRODUCTION . . . . .	1
1.1	Thesis Contributions . . . . .	4
1.2	Thesis Organization . . . . .	4
CHAPTER 2	ADAFACE: QUALITY ADAPTIVE MARGIN LOSS FOR FACE RECOGNITION . . . . .	6
2.1	Introduction . . . . .	6
2.2	Related Work . . . . .	9
2.3	Proposed Approach . . . . .	12
2.4	Experiments . . . . .	18
2.5	Gradient Scaling Term . . . . .	24
2.6	Feature Norm Analysis . . . . .	25
2.7	Visualization of Success and Failed Test Images . . . . .	30
2.8	Comparison with General Image-Quality Aware Learning Method . . . . .	30
2.9	Effect of Batch Size . . . . .	31
2.10	Implementation Details and Code . . . . .	31
2.11	Conclusion . . . . .	32
CHAPTER 3	CLUSTER AND AGGREGATE: FACE RECOGNITION WITH LARGE PROBE SET . . . . .	33
3.1	Introduction . . . . .	33
3.2	Related Work . . . . .	37
3.3	Proposed Approach . . . . .	38
3.4	Experiments . . . . .	45
3.5	Implementation Details . . . . .	51
3.6	Norm Embedding . . . . .	51
3.7	Additional Performance Results . . . . .	52
3.8	Resource and Efficiency Comparison . . . . .	53
3.9	Training Progress and Learned Assignment . . . . .	55
3.10	Weight Visualization . . . . .	56
3.11	Comparison of Assignment Maps in Various Scenarios . . . . .	56
3.12	Effect of Sequence Length . . . . .	57
3.13	Conclusions . . . . .	58
CHAPTER 4	DCFACE: SYNTHETIC FACE GENERATION WITH DUAL CONDITION DIFFUSION MODEL . . . . .	60
4.1	Introduction . . . . .	60
4.2	Related Works . . . . .	64
4.3	Proposed Approach . . . . .	65
4.4	Dataset Evaluation . . . . .	71
4.5	Experiments . . . . .	73
4.6	Training Details . . . . .	78
4.7	More Experiment Results . . . . .	80

4.8	Analysis . . . . .	81
4.9	Visualizations . . . . .	85
4.10	Miscellaneous . . . . .	86
4.11	Societal Concerns . . . . .	87
4.12	Implementation Details and Code . . . . .	88
4.13	Conclusion . . . . .	88
CHAPTER 5	KEYPOINT RELATIVE POSITION ENCODING FOR FACE RECOGNITION . . . . .	89
5.1	Introduction . . . . .	89
5.2	Related Works . . . . .	92
5.3	Proposed Method . . . . .	94
5.4	Face Recognition Experiments . . . . .	98
5.5	Gait Recognition Experiments . . . . .	103
5.6	Training Details . . . . .	104
5.7	Supplementary Performance Analysis . . . . .	105
5.8	Training Landmark Detector (MobileNet-RetinaFace) . . . . .	108
5.9	IJB-S Evaluation Method . . . . .	111
5.10	Alignment Visualizations . . . . .	113
5.11	Comparison with SoTA Off-the-Shelf Landmark Detector . . . . .	113
5.12	Pipeline Detail . . . . .	114
5.13	KPRPE Visualization . . . . .	115
5.14	Conclusion . . . . .	115
CHAPTER 6	SAPIENSID: FOUNDATION MODEL FOR UNIFIED HUMAN RECOGNITION . . . . .	118
6.1	Introduction . . . . .	118
6.2	Related Works . . . . .	121
6.3	Proposed Method . . . . .	123
6.4	Experiments . . . . .	133
6.5	Method Details . . . . .	137
6.6	Performance . . . . .	143
6.7	Visualization . . . . .	148
6.8	Potential Application of Retina Patch . . . . .	153
6.9	Limitations . . . . .	153
6.10	Ethical Concerns . . . . .	154
6.11	Conclusion . . . . .	154
CHAPTER 7	EFFICIENT HUMAN RECOGNITION FRAMEWORK . . . . .	155
7.1	Introduction . . . . .	155
7.2	Related Work . . . . .	158
7.3	Proposed Work . . . . .	159
7.4	Experiments . . . . .	164
7.5	Conclusion . . . . .	166

CHAPTER 8	DISCUSSION AND CONCLUSION . . . . .	167
8.1	Historical Context and Research Trajectory . . . . .	167
8.2	Limitations and Open Challenges . . . . .	168
8.3	Looking Ahead: Potential Future Directions . . . . .	172
8.4	Closing Remarks . . . . .	173
BIBLIOGRAPHY	. . . . .	174

# CHAPTER 1

## INTRODUCTION

For five decades, the pursuit of automated facial recognition has evolved from a futuristic concept into a tangible and widely deployed technology. Initially perceived as a computationally formidable challenge demanding laborious, hand-crafted feature extraction techniques, exemplified by early efforts [113], face recognition (FR) has matured into a foundational element of contemporary security systems, user convenience features, and online social platforms. This remarkable trajectory, increasingly driven by sophisticated deep learning models, is intertwined with complex considerations surrounding ethics, data governance, and the very definition of 'identity' in automated systems. This paper examines this half-century progression, highlighting key technological milestones, persistent challenges, and the future directions anticipated for automated FR.

Today, FR stands as perhaps the most widely utilized biometric identification method, partly because it closely resembles the way humans naturally identify one another [4, 108]. Several practical advantages underpin its ubiquity. Facial identification can occur without direct contact and from a distance, offering a less invasive experience compared to biometrics requiring physical touch, such as fingerprint or iris scanning [6]. The technology readily integrates with affordable camera sensors, fostering accessibility and large-scale implementation across various applications [6, 108]. The non-contact nature also presents hygienic advantages, a factor gaining prominence recently [5, 135]. Moreover, FR systems can operate discreetly using prevalent surveillance infrastructure and benefit significantly from the vast repositories of facial images already existing in government and institutional databases, including passport, visa, and driver's license photos [108].

The field of face recognition has undergone a paradigm shift with the advent of generative artificial intelligence (GenAI) [43, 81]. No longer confined to passive analysis, visual recognition now operates within an ecosystem where synthetic data generation, augmentation, and domain adaptation powered by GenAI have become integral to advancing state-of-the-art

systems [15, 228]. This transition is reshaping how machines perceive, interpret, and interact with the visual world, unlocking possibilities that were previously unattainable. From identifying individuals in images to interpreting complex scenes and bridging multimodal domains, visual recognition is now a cornerstone of modern AI applications.

However, achieving robust visual recognition in unconstrained, real-world environments presents multifaceted challenges. Variability in image quality, diverse lighting conditions, complex poses, occlusions, and the scarcity of high-quality labeled datasets remain persistent barriers. GenAI [64, 98, 232, 302] offers transformative solutions to these issues, yet it also introduces new concerns, such as domain gap, data privacy and the potential misuse of synthetic content [82, 133, 234, 300]. To fully harness the capabilities of generative technologies while addressing these concerns, the field must innovate across algorithm design, data generation strategies, and ethical frameworks.

This thesis explores visual recognition through the lens of biometric recognition (face and body) and the challenges posed by image degradation, the limitations of existing datasets, and the need for unified recognition systems that can operate seamlessly across modalities. While generative AI plays a pivotal role in some aspects—particularly in data synthesis and domain adaptation—the broader narrative reflects a holistic effort to build recognition systems that are robust, efficient, and generalizable to real-world complexities.

AdaFace [122] introduces an adaptive loss function that prioritizes learning from high-quality, informative samples, mitigating the adverse effects of low-quality images often encountered in real-world data. CAFace [123] further extends the challenge to the video domain and proposes an attention-based feature fusion framework that performs frame selection on arbitrary length of video. Both works lay the foundation for conducting robust visual recognition in the presence of low quality imageries.

DCFace [124] pioneers face dataset generation task, synthesizing diverse and realistic face datasets. This work marks a significant step toward replacing real datasets with synthetic ones, addressing ethical concerns associated with biometric data collection. Additionally, it

investigates the potential advantages of incorporating synthetic datasets alongside real data to enhance visual recognition performance.

KPRPE [125] incorporates semantic keypoints in visual recognition, enhancing their resilience to misalignment and geometric transformations often prevalent in low quality images. This innovation highlights how generative-inspired positional encoding can empower models to better handle real-world variability.

SapiensID [126] further takes this idea of keypoint enhanced recognition and introduces a unified recognition system that spans facial and full-body identification, demonstrating the potential for overcoming modality-specific boundaries. Complemented by the creation of the WebBody4M dataset, SapiensID exemplifies the importance of good quality data in creating versatile and generalizable recognition systems.

Together, these contributions provide a comprehensive framework for advancing visual recognition systems in the context of real-world challenges and the transformative influence of generative artificial intelligence. By addressing critical aspects such as image degradation, the scarcity of high-quality datasets, and the complexities of multimodal recognition, this thesis demonstrates how a combination of innovative algorithms, adaptive methodologies, and ethical considerations can push the boundaries of the field. Generative AI serves as both a tool and a catalyst in this journey, enabling solutions that are robust, efficient, and ethically sound.

This exploration of visual recognition under the GenAI paradigm underscores its pivotal role in shaping the future of biometric and multimodal recognition. By seamlessly integrating generative methodologies with resilient recognition frameworks, this work lays the foundation for systems capable of thriving in diverse and unpredictable environments. The advancements presented here not only address the specific challenges of face and body recognition but also pave the way for a new generation of intelligent systems equipped to meet the demands of increasingly complex visual domains.

## 1.1 Thesis Contributions

This thesis addresses critical challenges in visual recognition, focusing on robustness, data efficiency, and ethical considerations. The primary contributions are as follows:

- **Robust Recognition under Image Degradation:** *AdaFace* introduces a novel adaptive loss function that learns better representations from low-quality images, enhancing performance in degraded conditions.
- **Scalable Recognition in Video Data:** *CAFace* proposes a novel feature fusion framework with clustering and aggregation mechanisms based on attention, enabling efficient recognition that scales to long videos.
- **Synthetic Dataset Generation:** *DCFace* pioneers the generation of diverse synthetic datasets with dual condition diffusion model and demonstrates the benefits of combining synthetic and real data for enhanced visual recognition.
- **Resilience to Misalignment:** *KPRPE* introduces KeyPoint Relative Position Encoding, an enhancement to traditional relative position encoding, making recognition robust to misalignment and geometric transformations.
- **Unified Recognition Across Modalities:** *SapiensID* presents a system for recognizing both faces and bodies, supported by the WebBody4M dataset, emphasizing versatility and generalizability.
- **Efficient Unified Recognition:** The final contribution proposes an efficient ViT-based architecture that unifies face and body recognition while reducing computational cost. It introduces Keypoint-based Token Fusion (KP-ToFu) and Keypoint Absolute Position Encoding (KP-APE), achieving state-of-the-art results with significantly lower FLOPs.

## 1.2 Thesis Organization

The thesis is organized as follows:

- **Chapter 2:** Introduces *AdaFace* for handling low-quality images.
- **Chapter 3:** Discusses *CAFace* for robust video-based recognition.
- **Chapter 4:** Presents *DCFace* for synthetic dataset generation.

- **Chapter 5:** Explores *KPRPE* for improved robustness to misalignment.
- **Chapter 6:** Details *SapiensID* for unified face and body recognition.
- **Chapter 7:** Describes the proposed efficient ViT-based backbone with KP-ToFu and KP-APE for real-time unified human recognition.
- **Chapter 8:** Discussion of current limitations and future research directions.



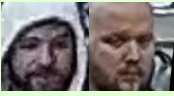
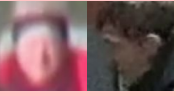
## CHAPTER 2

### ADAFACE: QUALITY ADAPTIVE MARGIN LOSS FOR FACE RECOGNITION

Recognition in low quality face datasets is challenging because facial attributes are obscured and degraded. Advances in margin-based loss functions have resulted in enhanced discriminability of faces in the embedding space. Further, previous studies have studied the effect of adaptive losses to assign more importance to misclassified (hard) examples. In this work, we introduce another aspect of adaptiveness in the loss function, namely the image quality. We argue that the strategy to emphasize misclassified samples should be adjusted according to their image quality. Specifically, the relative importance of easy or hard samples should be based on the sample’s image quality. We propose a new loss function that emphasizes samples of different difficulties based on their image quality. Our method achieves this in the form of an adaptive margin function by approximating the image quality with feature norms. Extensive experiments show that our method, AdaFace, improves the face recognition performance over the state-of-the-art (SoTA) on four datasets (IJB-B, IJB-C, IJB-S and TinyFace). Code and models are released in [Link](#).

#### 2.1 Introduction

Image quality is a combination of attributes that indicates how faithfully an image captures the original scene [206]. Factors that affect the image quality include brightness, contrast, sharpness, noise, color constancy, resolution, tone reproduction, *etc.* Face images, the focus of this paper, can be captured under a variety of settings for lighting, pose and facial expression, and sometimes under extreme visual changes such as a subject’s age or make-up. These parameter settings make the recognition task difficult for learned face recognition (FR) models. Still, the task is achievable in the sense that humans or models can often recognize faces under these difficult settings [231]. However, when a face image is of low quality, depending on the degree, the recognition task becomes infeasible. Fig. 2.1 shows examples of both high quality and low quality face images. It is not possible to recognize the

Recognizability Image Quality	Easy to Recognize	Hard to Recognize	Impossible to Recognize
High Quality			
Low Quality			



 : Images contain enough clues to identify the subject  
 : Images **do not** have enough clues to identify the subject

Figure 2.1 Examples of face images with different qualities and recognizabilities. Both high and low quality images contain variations in pose, occlusion and resolution that sometimes make the recognition task difficult, yet achievable. Depending on the degree of degradation, some images may become impossible to recognize. By studying the different impacts these images have in training, this work aims to design a novel loss function that is adaptive to a sample’s recognizability, driven by its image quality.

subjects in the last column of Fig. 2.1.

Low quality images like the bottom row of Fig. 2.1 are increasingly becoming an important part of face recognition datasets because they are encountered in surveillance videos and drone footage. Given that SoTA FR methods [55, 56, 102, 139] are able to obtain over 98% verification accuracy in relatively high quality datasets such as LFW or CFP-FP [100, 202], recent FR challenges have moved to lower quality datasets such as IJB-B, IJB-C and IJB-S [112, 169, 253]. Although the challenge is to attain high accuracy on low quality datasets, most popular training datasets still remain comprised of high quality images [55, 82]. Since only a small portion of training data is low quality, it is important to properly leverage it during training.

One problem with low quality face images is that they tend to be unrecognizable. When the image degradation is too large, the relevant identity information vanishes from the image, resulting in *unidentifiable images*. These unidentifiable images are detrimental to the training procedure since a model will try to exploit other visual characteristics, such as clothing color or image resolution, to lower the training loss. If these images are dominant in the distribution

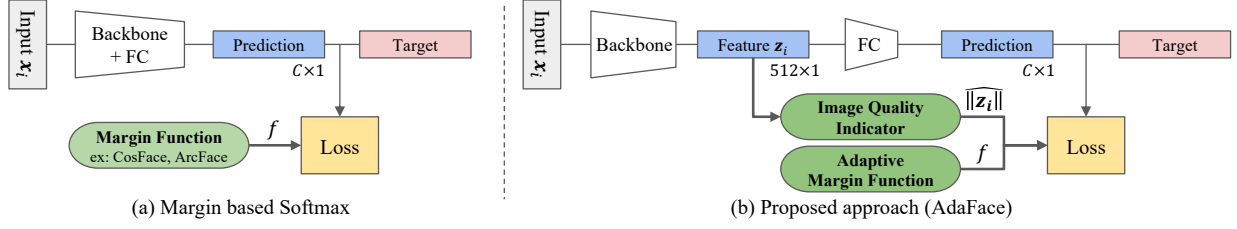


Figure 2.2 Conventional margin based softmax loss vs our AdaFace. (a) A FR training pipeline with a margin based softmax loss. The loss function takes the margin function to induce smaller intra-class variations. Some examples are SphereFace, CosFace and ArcFace [55, 154, 240]. (b) Proposed adaptive margin function (AdaFace) that is adjusted based on the image quality indicator. If the image quality is indicated to be low, the loss function emphasizes easy samples (thereby avoiding unidentifiable images). Otherwise, the loss emphasizes hard samples.

of low quality images, the model is likely to perform poorly on low quality datasets during testing.

Motivated by the presence of unidentifiable facial images, we would like to design a loss function which assigns different importance to samples of different difficulties according to the image quality. We aim to emphasize hard samples for the high quality images and easy samples for low quality images. Typically, assigning different importance to different difficulties of samples is done by looking at the training progression (curriculum learning) [19, 102]. Yet, we show that the sample importance should be adjusted by looking at both the difficulty and the image quality.

The reason why importance should be set differently according to the image quality is that naively emphasizing hard samples always puts a strong emphasis on unidentifiable images. This is because one can only make a random guess about unidentifiable images and thus, they are always in the hard sample group. There are challenges in introducing image quality into the objective. This is because image quality is a term that is hard to quantify due to its broad definition and scaling samples based on the difficulty often introduces ad-hoc procedures that are heuristic in nature.

In this work, we present a loss function to achieve the above goal in a seamless way. We find that 1) feature norm can be a good proxy for the image quality, and 2) various margin functions amount to assigning different importance to different difficulties of samples. These two findings are combined in a unified loss function, AdaFace, that adaptively changes the margin function to assign different importance to different difficulties of samples, based on the image quality (see Fig. 2.2).

In summary, the contributions of this paper include:

- We propose a loss function, AdaFace, that assigns different importance to different difficulties of samples according to their image quality. By incorporating image quality, we avoid emphasizing unidentifiable images while focusing on hard yet recognizable samples.
- We show that the angular margin scales the learning signal (gradient) based on the training sample’s difficulty. This observation motivates us to change margin function adaptively to emphasize hard samples if the image quality is high, and ignore very hard samples (unidentifiable images) if the image quality is low.
- We demonstrate that feature norms can serve as the proxy of image quality. It bypasses the need for an additional module to estimate image quality. Thus, adaptive margin function is achieved without additional complexity.
- We verify the efficacy of the proposed method by extensive evaluations on 9 datasets (LFW, CFP-FP, CPLFW, AgeDB, CALFW, IJB-B, IJB-C, IJB-S and TinyFace) of various qualities. We show that the recognition performance on low quality datasets can be hugely increased while maintaining performance on high quality datasets.

## 2.2 Related Work

**Margin Based Loss Function** The margin based softmax loss function is widely used for training face recognition (FR) models [55, 102, 154, 240]. Margin is added to the softmax loss because without the margin, learned features are not sufficiently discriminative. SphereFace [154], CosFace [240] and ArcFace [55] introduce different forms of margin functions.

Specifically, it can be written as,

$$\mathcal{L} = -\log \frac{\exp(f(\theta_{y_i}, m))}{\exp(f(\theta_{y_i}, m)) + \sum_{j \neq y_i}^n \exp(s \cos \theta_j)}, \quad (2.1)$$

where  $\theta_j$  is the angle between the feature vector and the  $j^{th}$  classifier weight vector,  $y_i$  is the index of the ground truth (GT) label, and  $m$  is the margin, which is a scalar hyper-parameter.  $f$  is a margin function, where

$$f(\theta_j, m)_{\text{SphereFace}} = \begin{cases} s \cos(m\theta_j) & j = y_i \\ s \cos \theta_j & j \neq y_i \end{cases}, \quad (2.2)$$

$$f(\theta_j, m)_{\text{CosFace}} = \begin{cases} s(\cos \theta_j - m) & j = y_i \\ s \cos \theta_j & j \neq y_i \end{cases}, \quad (2.3)$$

$$f(\theta_j, m)_{\text{ArcFace}} = \begin{cases} s \cos(\theta_j + m) & j = y_i \\ s \cos \theta_j & j \neq y_i \end{cases}. \quad (2.4)$$

Sometimes, ArcFace is referred to as an *angular* margin and CosFace is referred to as an *additive* margin. Here,  $s$  is a hyper-parameter for scaling. P2SGrad [287] notes that  $m$  and  $s$  are sensitive hyper-parameters and proposes to directly modify the gradient to be free of  $m$  and  $s$ .

Our approach aims to model the margin  $m$  as a function of the image quality because  $f(\theta_{y_i}, m)$  has an impact on which samples contribute more gradient (*i.e.* learning signal) during training.

**Adaptive Loss Functions** Many studies have introduced an element of adaptiveness in the training objective for either hard sample mining [145, 248], scheduling difficulty during training [102, 211], or finding optimal hyperparameters [286]. For example, CurricularFace [102] brings the idea of curriculum learning into the loss function. During the initial stages of training, the margin for  $\cos \theta_j$  (negative cosine similarity) is set to be small so that easy samples can be learned and in the later stages, the margin is increased so that hard samples

are learned. Specifically, it is written as

$$f(\theta_j, m)_{\text{Curricular}} = \begin{cases} s \cos(\theta_j + m) & j = y_i \\ N(t, \cos \theta_j) & j \neq y_i \end{cases}, \quad (2.5)$$

where

$$N(t, \cos \theta_j) = \begin{cases} \cos(\theta_j) & s \cos(\theta_{y_i} + m) \geq \cos \theta_j \\ \cos(\theta_j)(t + \cos \theta_j) & s \cos(\theta_{y_i} + m) < \cos \theta_j \end{cases}, \quad (2.6)$$

and  $t$  is a parameter that increases as the training progresses. Therefore, in CurricularFace, the adaptiveness in the margin is based on the training progression (curriculum).

On the contrary, we argue that the adaptiveness in the margin should be based on the image quality. We believe that among high quality images, if a sample is hard (with respect to a model), the network should learn to exploit the information in the image, but in low quality images, if a sample is hard, it is more likely to be devoid of proper identity clues and the network should not try hard to fit on it.

MagFace [172] explores the idea of applying different margins based on recognizability. It applies large angular margins to high norm features on the premise that high norm features are easily recognizable. Large margin pushes features of high norm closer to class centers. Yet, it fails to emphasize hard training samples, which is important for learning discriminative features. It is also worth mentioning that DDL [101] uses the distillation loss to minimize the gap between easy and hard sample features.

**Face Recognition with Low Quality Images** Recent FR models have achieved high performance on datasets where facial attributes are discernable, *e.g.*, LFW [100], CFP-FP [202], CPLFW [296], AgeDB [174] and CALFW [297]. Good performance on these datasets can be achieved when the FR model learns discriminative features invariant to lighting, age or pose variations. However, FR in unconstrained scenarios such as in surveillance or low quality videos [276] brings more problems to the table. Examples of datasets in this setting are IJB-B [253], IJB-C [169] and IJB-S [112], where most of the images are of low quality, and some do not contain sufficient identity information, even for human examiners. The key to

good performance involves both 1) learning discriminative features for low quality images and 2) learning to discard images that contain few identity cues. The latter is sometimes referred to as *quality aware fusion*.

To perform quality aware fusion, probabilistic approaches have been proposed to predict uncertainty in FR representation [38, 139, 181, 208, 298]. They assume the features are distributions and the variance can be used to calculate the certainty in prediction. However, probabilistic approaches often resort to learning mean and variance separately, which is not simple during training and suboptimal as the variance is optimized with a fixed mean. Our work, however, is a modification to the conventional softmax loss, making the framework easy to use. And we use the feature norm as a proxy for quality during quality-aware fusion.

QSub-PM [293] and UGG [294] also show good performances in LQ video recognition by using rich subspace (matrix) representation for comparison and using auxiliary context (such as a body) to aid feature fusion respectively.

Synthetic data or augmentations can be used to mimic low quality data. [69, 210] adopts 3D reconstruction to generate faces. Extra steps complicate the training procedure, making it hard to generalize to other domains. We adopt easily applicable crop, blur and photometric augmentations.

### 2.3 Proposed Approach

The cross entropy softmax loss of a sample  $\mathbf{x}_i$  can be formulated as follows,

$$\mathcal{L}_{CE}(\mathbf{x}_i) = -\log \frac{\exp(\mathbf{W}_{y_i} \mathbf{z}_i + b_{y_i})}{\sum_{j=1}^C \exp(\mathbf{W}_j \mathbf{z}_j + b_j)}, \quad (2.7)$$

where  $\mathbf{z}_i \in \mathbb{R}^d$  is the  $\mathbf{x}_i$ 's feature embedding, and  $\mathbf{x}_i$  belongs to the  $y_i$ th class.  $\mathbf{W}_j$  refers to the  $j$ th column of the last FC layer weight matrix,  $\mathbf{W} \in \mathbb{R}^{d \times C}$ , and  $b_j$  refers to the corresponding bias term.  $C$  refers to the number of classes.

During test time, for an arbitrary pair of images,  $\mathbf{x}_p$  and  $\mathbf{x}_q$ , the cosine similarity metric,  $\frac{\mathbf{z}_p \cdot \mathbf{z}_q}{\|\mathbf{z}_p\| \|\mathbf{z}_q\|}$  is used to find the closest matching identities. To make the training objective directly optimize the cosine distance, [154, 239] use normalized softmax where the bias term is set to

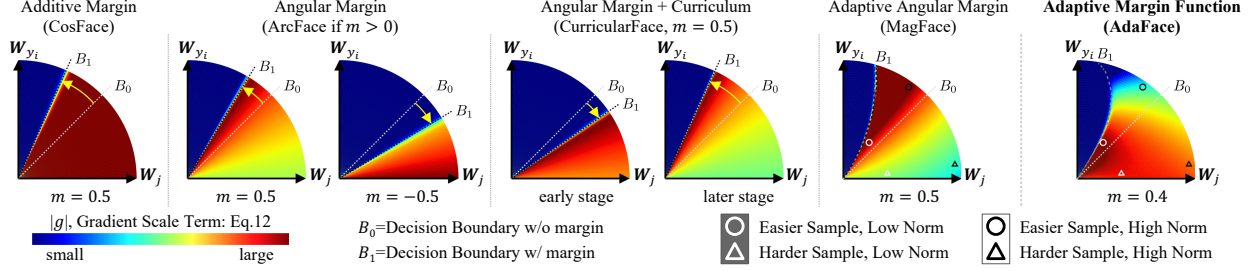


Figure 2.3 Illustration of different margin functions and their gradient scaling terms on the feature space.  $B_0$  and  $B_1$  show the decision boundary with and without margin  $m$ , respectively. The yellow arrow indicates the shift in the boundary due to margin  $m$ . Our work adaptively changes the margin functions based on the norm. With high norm, we emphasize samples away from the boundary and with low norm we emphasize samples near the boundary. Circles and triangles in the arc show example scenarios in the right most plot (AdaFace).

zero and the feature  $\mathbf{z}_i$  is normalized and rescaled with  $s$  during training. This modification results in

$$\mathcal{L}_{CE}(\mathbf{x}_i) = -\log \frac{\exp(s \cdot \cos \theta_{y_i})}{\sum_{j=1}^C \exp(s \cos \theta_j)}, \quad (2.8)$$

where  $\theta_j$  corresponds to the angle between  $\mathbf{z}_i$  and  $\mathbf{W}_j$ . Follow-up works [55, 240] take this formulation and introduces a margin to reduce the intra-class variations. Generally, it can be written as Eq. 2.1 where margin functions are defined in Eqs. 2.2, 2.3 and 2.4 correspondingly.

### 2.3.1 Margin Form and the Gradient

Previous works on margin based softmax focused on how the margin shifts the decision boundaries and what their geometric interpretations are [55, 240]. In this section, we show that during backpropagation, the gradient change due to the margin has the effect of scaling the importance of a sample relative to the others. In other words, angular margin can introduce an additional term in the gradient equation that scales the signal according to the sample’s difficulty. To show this, we will look at how the gradient equation changes with the margin function  $f(\theta_{y_i}, m)$ .

Let  $P_j^{(i)}$  be the probability output at class  $j$  after softmax operation on an input  $\mathbf{x}_i$ . By

deriving the gradient equations for  $\mathcal{L}_{CE}$  w.r.t.  $\mathbf{W}_j$  and  $\mathbf{x}_i$ , we obtain the following,

$$P_j^{(i)} = \frac{\exp(f(\cos \theta_{y_i}))}{\exp(f(\cos \theta_{y_i})) + \sum_{j \neq y_i}^n \exp(s \cos \theta_j)}, \quad (2.9)$$

$$\frac{\partial \mathcal{L}_{CE}}{\partial \mathbf{W}_j} = \left( P_j^{(i)} - \mathbb{1}(y_i = j) \right) \frac{\partial f(\cos \theta_j)}{\partial \cos \theta_j} \frac{\partial \cos \theta_j}{\partial \mathbf{W}_j}, \quad (2.10)$$

$$\frac{\partial \mathcal{L}_{CE}}{\partial \mathbf{x}_i} = \sum_{k=1}^C \left( P_k^{(i)} - \mathbb{1}(y_i = k) \right) \frac{\partial f(\cos \theta_k)}{\partial \cos \theta_k} \frac{\partial \cos \theta_k}{\partial \mathbf{x}_i}. \quad (2.11)$$

In Eqs. 2.10 and 2.11, the first two terms,  $\left( P_j^{(i)} - \mathbb{1}(y_i = j) \right)$  and  $\frac{\partial f(\cos \theta_j)}{\partial \cos \theta_j}$  are scalars. Also, these two are the only terms affected by parameter  $m$  through  $f(\cos \theta_{y_i})$ . As the direction term,  $\frac{\partial \cos \theta_j}{\partial \mathbf{W}_j}$  is free of  $m$ , we can think of the first two scalar terms as a gradient scaling term (GST) and denote,

$$g := \left( P_j^{(i)} - \mathbb{1}(y_i = j) \right) \frac{\partial f(\cos \theta_j)}{\partial \cos \theta_j}. \quad (2.12)$$

For the purpose of the GST analysis, we will consider the class index  $j = y_i$ , since all negative class indices  $j \neq y_i$  do not have a margin in Eqs. 2.2, 2.3, and 2.4. The GST for the normalized softmax loss is

$$g_{\text{softmax}} = (P_{y_i}^{(i)} - 1)s, \quad (2.13)$$

since  $f(\cos \theta_{y_i}) = s \cdot \cos \theta_{y_i}$  and  $\frac{\partial f(\cos \theta_{y_i})}{\partial \cos \theta_{y_i}} = s$ . The GST for the CosFace [240] is also

$$g_{\text{CosFace}} = (P_{y_i}^{(i)} - 1)s, \quad (2.14)$$

as  $f(\cos \theta_{y_i}) = s(\cos \theta_{y_i} - m)$  and  $\frac{\partial f(\cos \theta_{y_i})}{\partial \cos \theta_{y_i}} = s$ . Yet, the GST for ArcFace [55] turns out to be

$$g_{\text{ArcFace}} = (P_j^{(i)} - 1)s \left( \cos(m) + \frac{\cos \theta_{y_i} \sin(m)}{\sqrt{1 - \cos^2 \theta_{y_i}}} \right). \quad (2.15)$$

Since the GST is a function of  $\theta_{y_i}$  and  $m$  as in Eq. 2.15, it is possible to use it to control the emphasis on samples based on the difficulty, *i.e.*,  $\theta_{y_i}$  during training.

To understand the effect of GST, we visualize GST *w.r.t.* the features. Fig. 2.3 shows the GST as the color in the feature space. Note that for the angular margin, the GST peaks at

the decision boundary but slowly decreases as it moves away towards  $\mathbf{W}_j$  and harder samples receive less emphasis. If we change the sign of the angular margin, we see an opposite effect. Note that, in the 6th column, MagFace [172] is an extension of ArcFace (positive angular margin) with larger margin assigned to high norm feature. Both ArcFace and MagFace fail to put high emphasis on hard samples (green area near  $\mathbf{W}_j$ ). We combine all margin functions (positive and negative angular margins and additive margins) to emphasize hard samples when necessary.

Note that this adaptiveness is also different from approaches that use the training stage to change the relative importance of different difficulties of samples [102]. Fig. 2.3 shows CurricularFace where the decision boundary and the GST  $g$  change depending on the training stage.

### 2.3.2 Norm and Image quality

Image quality is a comprehensive term that covers characteristics such as brightness, contrast and sharpness. Image quality assessment (IQA) is widely studied in computer vision [283]. SER-FIQ [227] is an unsupervised DL method for face IQA. BRISQUE [173] is a popular algorithm for blind/no-reference IQA. However, such methods are computationally expensive to use during training. In this work, we refrain from introducing an additional module that calculates the image quality. Instead, we use the feature norm as a proxy for the image quality. We observe that, in models trained with a margin-based softmax loss, the feature norm exhibits a trend that is correlated with the image quality.

In Fig. 2.4 (a) we show a correlation plot between the feature norm and the image quality (IQ) score calculated with (1-BRISQUE) as a green curve. We randomly sampled 1,534 images from the training dataset (MS1MV2 [55] with augmentations described in Sec. 2.4.1) and calculate the feature norm using a pretrained model. At the final epoch, the correlation score between the feature norm and IQ score reaches 0.5235 (out of  $-1$  and  $1$ ). The corresponding scatter plot is shown in Fig. 2.4 (b). This high correlation between the feature norm and the IQ score supports our use of feature norm as the proxy of image quality.

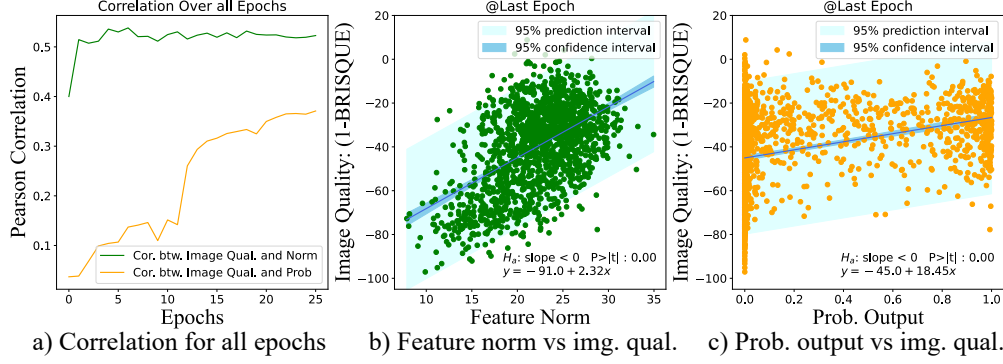


Figure 2.4 (a) A plot of Pearson correlation with image quality score (1-BRISQUE) over training epochs. The green and orange curves correspond to the correlation plot using the feature norm  $\|\mathbf{z}_i\|$  and the probability output for the ground truth index  $P_{y_i}$ , respectively. (b) and (c) Corresponding scatter plots for the last epoch. The blue line on the scatter plot and the corresponding equation shows the least square line fitted to the data points.

In Fig. 2.4 (a) we also show a correlation plot between the probability output  $P_{y_i}$  and the IQ score as an orange curve. Note that the correlation is always higher for the feature norm than for  $P_{y_i}$ . Furthermore, the correlation between the feature norm and IQ score is visible from an early stage of training. This is a useful property for using the feature norm as the proxy of image quality because we can rely on the proxy from the early stage of training. Also, in Fig. 2.4 (c), we show a scatter plot between  $P_{y_i}$  and IQ score. Notice that there is a non-linear relationship between  $P_{y_i}$  and the image quality. One way to describe a sample’s difficulty is with  $1 - P_{y_i}$ , and the plot shows that the distribution of the difficulty of samples is different based on image quality. Therefore, it makes sense to consider the image quality when adjusting the sample importance according to the difficulty.

### 2.3.3 AdaFace: Adaptive Margin based on Norm

To address the problem caused by the unidentifiable images, we propose to adapt the margin function based on the feature norm. In Sec. 2.3.1, we have shown that using different margin functions can emphasize different difficulties of samples. Also, in Sec. 2.3.2, we have observed that the feature norm can be a good way to find low quality images. We will merge the two findings and propose a new loss for FR.

**Image Quality Indicator** As the feature norm,  $\|\mathbf{z}_i\|$  is a model dependent quantity, we normalize it using batch statistics  $\mu_z$  and  $\sigma_z$ . Specifically, we let

$$\widehat{\|\mathbf{z}_i\|} = \left[ \frac{\|\mathbf{z}_i\| - \mu_z}{\sigma_z/h} \right]_{-1}^1, \quad (2.16)$$

where  $\mu_z$  and  $\sigma_z$  are the mean and standard deviation of all  $\|\mathbf{z}_i\|$  within a batch. And  $[\cdot]$  refers to clipping the value between  $-1$  and  $1$  and stopping the gradient from flowing. Since  $\frac{\|\mathbf{z}_i\| - \mu_z}{\sigma_z/h}$  makes the batch distribution of  $\widehat{\|\mathbf{z}_i\|}$  as approximately unit Gaussian, we clip the value to be within  $-1$  and  $1$  for better handling. It is known that approximately 68% of the unit Gaussian distribution falls between  $-1$  and  $1$ , so we introduce the term  $h$  to control the concentration. We set  $h$  such that most of the values  $\frac{\|\mathbf{z}_i\| - \mu_z}{\sigma_z/h}$  fall between  $-1$  and  $1$ . A good value to achieve this would be  $h = 0.33$ . Later in Sec. 2.4.2, we ablate and validate this claim. We stop the gradient from flowing during backpropagation because we do not want features to be optimized to have low norms.

If the batch size is small, the batch statistics  $\mu_z$  and  $\sigma_z$  can be unstable. Thus we use the exponential moving average (EMA) of  $\mu_z$  and  $\sigma_z$  across multiple steps to stabilize the batch statistics. Specifically, let  $\mu^{(k)}$  and  $\sigma^{(k)}$  be the  $k$ -th step batch statistics of  $\|\mathbf{z}_i\|$ . Then

$$\mu_z = \alpha \mu_z^{(k)} + (1 - \alpha) \mu_z^{(k-1)}, \quad (2.17)$$

and  $\alpha$  is a momentum set to 0.99. The same is true for  $\sigma_z$ .

**Adaptive Margin Function** We design a margin function such that 1) if image quality is high, we emphasize hard samples, and 2) if image quality is low, we de-emphasize hard samples. We achieve this with two adaptive terms  $g_{\text{angle}}$  and  $g_{\text{add}}$ , referring to angular and additive margins, respectively. Specifically, we let

$$f(\theta_j, m)_{\text{AdaFace}} = \begin{cases} s(\cos(\theta_j + g_{\text{angle}}) - g_{\text{add}}) & j = y_i \\ s \cos \theta_j & j \neq y_i \end{cases} \quad (2.18)$$

where  $g_{\text{angle}}$  and  $g_{\text{add}}$  are the functions of  $\widehat{\|\mathbf{z}_i\|}$ . We define

$$g_{\text{angle}} = -m \cdot \widehat{\|\mathbf{z}_i\|}, \quad g_{\text{add}} = m \cdot \widehat{\|\mathbf{z}_i\|} + m. \quad (2.19)$$

Note that when  $\|\widehat{\mathbf{z}}_i\| = -1$ , the proposed function becomes ArcFace. When  $\|\widehat{\mathbf{z}}_i\| = 0$ , it becomes CosFace. When  $\|\widehat{\mathbf{z}}_i\| = 1$ , it becomes a negative angular margin with a shift. Fig. 2.3 shows the effect of the adaptive function on the gradient. The high norm features will receive a higher gradient scale, far away from the decision boundary, whereas the low norm features will receive higher gradient scale near the decision boundary. For low norm features, the harder samples away from the boundary are de-emphasized.

## 2.4 Experiments

### 2.4.1 Datasets and Implementation Details

**Datasets** We use MS1MV2 [55], MS1MV3 [57] and WebFace4M [300] as our training datasets. Each dataset contains 5.8M, 5.1M and 4.2M facial images, respectively. We test on 9 datasets of varying qualities. Following the protocol of [210], we categorize the test datasets into 3 types according to the visual quality (examples shown in Fig. 2.5).

- **High Quality:** LFW [100], CFP-FP [202], CPLFW [296] AgeDB [174] and CALFW [297] are popular benchmarks for FR in the well controlled setting. While the images show variations in lighting, pose, or age, they are of sufficiently good quality for face recognition.
- **Mixed Quality:** IJB-B and IJB-C [169, 253] are datasets collected for the purpose of introducing low quality images in the validation protocol. They contain both high quality images and low quality videos of celebrities.
- **Low Quality:** IJB-S [112] and TinyFace [46] are datasets with low quality images and/or videos. IJB-S is a surveillance video dataset, with test protocols such as *Surveillance-to-Single*, *Surveillance-to-Booking* and *Surveillance-to-Surveillance*. The first/second word in the protocol refers to the probe/gallery image source. *Surveillance* refers to the surveillance video, *Single* refers to a high quality enrollment image and *Booking* refers to multiple enrollment images taken from different viewpoints. TinyFace consists only of low quality images.

**Training Settings** We preprocess the dataset by cropping and aligning faces with five landmarks, as in [55, 285], resulting in  $112 \times 112$  images. For the backbone, we adopt



Figure 2.5 Examples of three categories of test datasets in our study.

ResNet [86] as modified in [55]. We use the same optimizer and a learning rate schedule as in [102], and train for 24 epochs. The model is trained with SGD with the initial learning rate of 0.1 and step scheduling at 10, 18 and 22 epochs. If the dataset contains augmentations, we add 2 more epochs for convergence. For the scale parameter  $s$ , we set it to 64, following the suggestion of [55, 240].

**Augmentations** Since our proposed method is designed to train better in the presence of unidentifiable images in the training data, we introduce three on-the-fly augmentations that are widely used in image classification tasks [88], *i.e.*, cropping, rescaling and photometric jittering. These augmentations will create more data but also introduce more unidentifiable images. It is a trade-off that has to be balanced. In FR, these augmentations are not used because they generally do not bring benefit to the performance (as shown in Sec. 2.4.2). We show that our loss function is capable of reaping the benefit of augmentations because it can adapt to ignore unidentifiable images.

Cropping defines a random rectangular area (patch) and makes the region outside the area to be 0. We do not cut and resize the image as the alignment of the face is important. Photometric augmentation randomly scales hue, saturation and brightness. Rescaling involves resizing an image to a smaller scale and back, resulting in blurriness. These operations are applied randomly with a probability of 0.2.

#### 2.4.2 Ablation and Analysis

For hyperparameter  $m$  and  $h$  ablation, we adopt a ResNet18 backbone and use 1/6th of the randomly sampled MS1MV2. We use two performance metrics. For High Quality Datasets

(HQ), we use an average of 1:1 verification accuracy in LFW, CFP-FP, CPLFW, AgeDB and CALFW. For Low Quality Datasets (LQ), we use an average of the closed-set rank-1 retrieval and the open-set TPIR@FIPR=1% for all 3 protocols of IJB-S. Unless otherwise stated, we augment the data as described in Sec. 2.4.1.

**Effect of Image Quality Indicator Concentration  $h$**  In Sec. 2.3.3, we claim that  $h = 0.33$  is a good value. To validate this claim, we show in Tab. 2.1 the performance when varying  $h$ . When  $h = 0.33$ , the model performs the best. For  $h = 0.22$  or  $h = 0.66$ , the performance is still higher than CurricularFace. As long as  $h$  is set such that  $\widehat{\|\mathbf{z}_i\|}$  has some variation,  $h$  is not very sensitive. We set  $h = 0.33$ .

**Effect of Hyperparameter  $m$**  The margin  $m$  corresponds to both the maximum range of the angular margin and the magnitude of the additive margin. Tab. 2.1 shows that the performance is best for HQ datasets when  $m = 0.4$  and for LQ datasets when  $m = 0.75$ . Large  $m$  results in large angular margin variation based on the image quality, resulting in more adaptivity. In subsequent experiments, we choose  $m = 0.4$  since it achieves good performance for LQ datasets without sacrificing performance on HQ datasets.

**Effect of Proxy Choice** In Tab. 2.1, to show the effectiveness of using the feature norm as a proxy for image quality, we switch the feature norm with other quantities such as (1-BRISQUE) or  $P_{y_i}$ . The performance using the feature norm is superior to using others. The BRISQUE score is precomputed for the training dataset, so it is not as effective in capturing the image quality when training with augmentation. We include  $P_{y_i}$  to show that the adaptiveness in feature norm is different from adaptiveness in difficulty.

**Effect of Augmentation** We introduce on-the-fly augmentations in our training data. Our proposed loss can effectively handle the unidentifiable images, which are generated occasionally during augmentations. We experiment with a larger model ResNet50 on the full MS1MV2 dataset.

Tab. 2.2 shows that indeed the augmentation brings performance gains for AdaFace. The performance on HQ datasets stays the same, whereas LQ datasets enjoy a significant

Method	$h$	$m$	Proxy	HQ Datasets	LQ Datasets
CurricularFace [102]	-	0.50		93.43	32.92
AdaFace	0.22	0.40	Norm	93.67	34.92
AdaFace	<b>0.33</b>			<b>93.74</b>	<b>35.40</b>
AdaFace	0.66			93.70	35.29
AdaFace	0.33	<b>0.40</b>	Norm	<b>93.74</b>	35.40
AdaFace		0.50		93.56	35.23
AdaFace		0.75		93.37	<b>35.69</b>
AdaFace	0.33	0.40	<b>Norm</b>	<b>93.74</b>	<b>35.40</b>
-			1-BRISQUE	93.43	34.55
-			$P_{y_i}$	93.46	35.17

Table 2.1 Ablation of our margin function parameters  $h$  and  $m$ , and the image quality proxy choice on the ResNet18 backbone. The performance metrics are as described in Sec. 2.4.2.

Method	$p$	HQ Datasets	LQ Datasets
CurricularFace [102]	<b>0.0</b>	<b>96.85</b>	<b>41.00</b>
CurricularFace [102]	0.2	96.75	40.84
CurricularFace [102]	0.3	96.59	40.58
AdaFace	0.0	96.72	40.95
AdaFace	<b>0.2</b>	<b>96.88</b>	41.82
AdaFace	0.3	96.78	<b>41.93</b>

Table 2.2 Ablation of augmentation probability  $p$ , on the ResNet50 backbone. The metrics are the same as Tab. 2.1.

performance gain. Note that the augmentation hurts the performance of CurricularFace, which is in line with our assumption that augmentation is a tradeoff between a positive effect from getting more data and a negative effect from unidentifiable images. Prior works on margin-based softmax do not include on-the-fly augmentations as the performance could be worse. AdaFace avoids overfitting on unidentifiable images, therefore it can exploit the augmentation better.

**Analysis** To show how the feature norm  $\|\mathbf{z}_i\|$  and the difficulty of training samples change during training, we plot the sample trajectory in Fig. 2.6. A total of 1,536 samples are randomly sampled from the training data. Each column in the heatmap represents a sample, and the x-axis is sorted according to the norm of the last epoch. Sample #600 is approximately a middle point of the transition from low to high norm samples. The bottom plot shows that many of the probability trajectories of low norm samples never get high probability till the end. It is in line with our claim that low norm features are more likely to be unidentifiable images. It justifies our motivation to put less emphasis on these cases, although they are

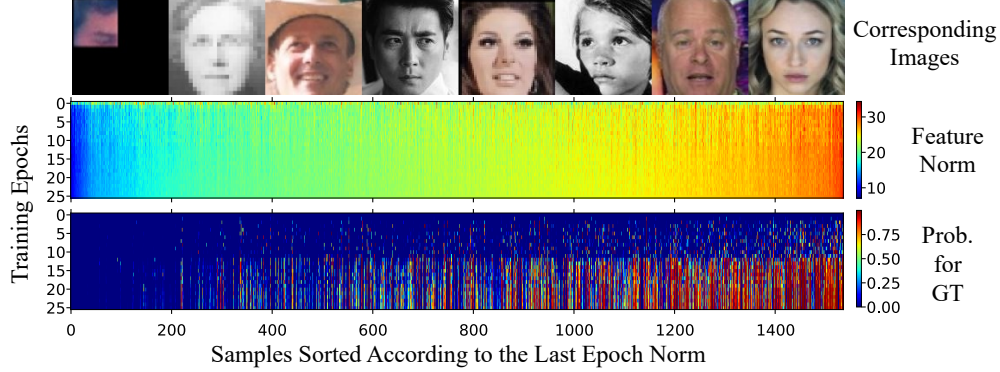


Figure 2.6 A plot of training samples’ trajectories of feature norm  $\|z_i\|$  and the probability output for the ground truth index  $P_{y_i}$ . We randomly select 1,536 samples from the training data with augmentations, and show 8 images evenly sampled from them. The features with low norm have a different probability trajectory than others and the corresponding images are hard to identify.

“hard” cases. The percentage of samples with augmentations is higher for the low norm features than for the high norm features. For samples number #0 to #600, about 62.0% are with at least one type of augmentation. For the samples #600 or higher, the percentage is about 38.5%.

**Time Complexity** Compared to classic margin-based loss functions, our method adds a negligible amount of computation in training. With the same setting, ArcFace [55] takes 0.3193s per iteration while AdaFace takes 0.3229s (+1%).

### 2.4.3 Comparison with SoTA methods

To compare with SoTA methods, we evaluate ResNet100 trained with AdaFace loss on 9 datasets as listed in Sec. 2.4.1. For the high quality datasets, Tab. 2.3 (a) shows that AdaFace performs on par with competitive methods such as BroadFace [128], SCF-ArcFace [139] and VPL-ArcFace [56]. This strong performance in high quality datasets is due to the hard sample emphasis on high quality cases during training. Note that some performances in high quality datasets are saturated, making the gain less pronounced. Thus, choosing one model over the others is somewhat difficult based solely on the numbers. Unlike SCF-ArcFace, our method does not use additional learnable layers, nor requires 2-stage training. It is a revamp of the

Method	Venue	Train Data	High Quality					Mixed Quality		
			LFW [100]	CFP-FP [202]	CPLFW [296]	AgeDB [174]	CALFW [297]	AVG	IJB-B [253]	IJB-C [169]
CosFace ( $m = 0.35$ ) [240]	CVPR18	MS1MV2	99.81	98.12	92.28	98.11	95.76	96.82	94.80	96.37
ArcFace ( $m = 0.50$ ) [55]	CVPR19	MS1MV2	<b>99.83</b>	98.27	92.08	98.28	95.45	96.78	94.25	96.03
AFRN [114]	ICCV19	MS1MV2	<b>99.85</b>	95.56	<b>93.48</b>	95.35	<b>96.30</b>	96.11	88.50	93.00
MV-Softmax [248]	AAAI20	MS1MV2	99.80	98.28	92.83	97.95	96.10	96.99	93.60	95.20
CurricularFace [102]	CVPR20	MS1MV2	99.80	98.37	93.13	<b>98.32</b>	<b>96.20</b>	97.16	94.80	96.10
URL [210]	CVPR20	MS1MV2	99.78	<b>98.64</b>	-	-	-	-	-	<b>96.60</b>
BroadFace [128]	ECCV20	MS1MV2	<b>99.85</b>	<b>98.63</b>	93.17	<b>98.38</b>	<b>96.20</b>	<b>97.25</b>	94.97	96.38
MagFace [172]	CVPR21	MS1MV2	<b>99.83</b>	98.46	92.87	98.17	96.15	97.10	94.51	95.97
SCF-ArcFace [139]	CVPR21	MS1MV2	99.82	98.40	93.16	98.30	96.12	97.16	94.74	96.09
DAM-CurricularFace [152]	ICCV21	MS1MV2	-	-	-	-	-	-	<b>95.12</b>	96.20
<b>AdaFace</b> ( $m = 0.4$ )	CVPR22	MS1MV2	99.82	98.49	<b>93.53</b>	98.05	96.08	<b>97.19</b>	<b>95.67</b>	<b>96.89</b>
VPL-ArcFace [56]	CVPR21	MS1MV3	<b>99.83</b>	<b>99.11</b>	93.45	<b>98.60</b>	<b>96.12</b>	<b>97.42</b>	95.56	96.76
<b>AdaFace</b> ( $m = 0.4$ )	CVPR22	MS1MV3	<b>99.83</b>	99.03	<b>93.93</b>	98.17	96.02	97.40	<b>95.84</b>	<b>97.09</b>
ArcFace* [55]	CVPR19	WebFace4M	<b>99.83</b>	<b>99.19</b>	94.35	<b>97.95</b>	96.00	97.46	95.75	97.16
<b>AdaFace</b> ( $m = 0.4$ )	CVPR22	WebFace4M	99.80	99.17	<b>94.63</b>	97.90	<b>96.05</b>	<b>97.51</b>	<b>96.03</b>	<b>97.39</b>

(a) A performance comparison of recent methods on high and mixed quality datasets.

Method	Train Data	Low Quality (IJB-S [112] and TinyFace [46])										
		Surveillance-to-Single [112]			Surveillance-to-Booking [112]			Surveillance-to-Surveillance [112]			TinyFace [46]	
		Rank-1	Rank-5	1%	Rank-1	Rank-5	1%	Rank-1	Rank-5	1%	Rank-1	Rank-5
PFE [208]	MS1MV2 [55]	50.16	58.33	31.88	53.60	61.75	35.99	9.20	20.82	0.84	-	-
ArcFace [55]	MS1MV2 [55]	57.35	64.42	41.85	57.36	64.95	41.23	-	-	-	-	-
URL [210]	MS1MV2 [55]	59.79	65.78	41.06	61.98	67.12	42.73	-	-	-	63.89	68.67
CurricularFace* [102]	MS1MV2 [55]	62.43	68.68	47.68	63.81	69.74	47.57	19.54	32.80	2.53	63.68	67.65
AdaFace ( $m = 0.4$ )	MS1MV2 [55]	65.26	70.53	51.66	66.27	71.61	50.87	23.74	37.47	2.50	68.21	71.54
AdaFace ( $m = 0.4$ )	MS1MV3 [57]	67.12	72.67	53.67	67.83	72.88	52.03	26.23	40.60	3.28	67.81	70.98
ArcFace* [55]	WebFace4M [300]	69.26	74.31	57.06	70.31	75.15	56.89	32.13	46.67	5.32	71.11	74.38
AdaFace ( $m = 0.4$ )	WebFace4M [300]	70.42	75.29	58.27	70.93	76.11	58.02	35.05	48.22	4.96	72.02	74.52

(b) A performance comparison of recent methods on low quality datasets.

Table 2.3 Comparison on benchmark datasets, with the ResNet100 backbone.

loss function, which makes it easier to apply our method to new tasks or backbones.

For mixed quality datasets, Tab. 2.3 (a) clearly shows the improvement of AdaFace. On IJB-B and IJB-C, AdaFace reduces the errors of the second best relatively by 11% and 9% respectively. This shows the efficacy of using feature norms as an image quality proxy to treat samples differently.

For low quality datasets, Tab. 2.3 (b) shows that AdaFace substantially outperforms all baselines. Compared to the second best, our averaged performance gain over 4 Rank-1 metrics is 3.5%, and over 3 TPIR@=FPIR=1% metrics is 2.4%. These results show that AdaFace is effective in learning a good representation for the low quality settings as it prevents the model from fitting on unidentifiable images.

We further train on a refined dataset, MS1MV3 [57] for a fair comparison with a recent work VPL-ArcFace [56]. The performance using MS1MV3 is higher than MS1MV2 due to less noise in MS1MV3. We also train on newly released WebFace4M [300] dataset. While one method might shine on one type of data, it is remarkable to see that collectively Adaface achieves SOTA performance on test data with a wide range of image quality, and on various training sets.

## 2.5 Gradient Scaling Term

In Sec. 3.1, the gradient scaling term (GST),  $g$  is introduced. Specifically, it is derived from the gradient equation for the margin-based softmax loss and defined as

$$g := \left( P_j^{(i)} - \mathbb{1}(y_i = j) \right) \frac{\partial f(\cos \theta_j)}{\partial \cos \theta_j}, \quad (2.20)$$

where

$$P_j^{(i)} = \frac{\exp(f(\cos \theta_{y_i}))}{\exp(f(\cos \theta_{y_i})) + \sum_{j \neq y_i}^n \exp(s \cos \theta_j)}. \quad (2.21)$$

This scalar term,  $g$  affects the magnitude of the gradient during backpropagation from the margin-based softmax loss. The form of  $g$  depends on the form of the margin function  $f(\cos \theta_j)$ . In Tab.1 of AdaFace Supplementary, we summarize the margin function  $f(\cos \theta_j)$  and the corresponding GST when  $j = y_i$ , the ground truth index.

Note that  $P_{y_i}$  is also affected by the choice of the margin function  $f(\cos \theta_{y_i})$  as in Eqn. 2.21. So,  $g$  is a function of  $m$ , except for Softmax, and  $g$  is affected by  $m$  through  $f(\cos \theta_{y_i})$  in  $P_{y_i}$ . For Angular Margin,  $m$  appears in the equation for  $g$  directly. We derive  $g$  for Angular Margin below. The term  $g$  for the Adaptive Angular Margin and CurricularFace [102] can be obtained using the  $g$  from the Angular Margin. The GST term for AdaFace can be obtained by using  $g$  for the Angular Margin and the Additive Margin, and replacing  $m$  with adaptive terms  $g_{\text{angle}}$  and  $g_{\text{add}}$ . This is possible because  $\|z_i\|$  is treated as a constant.

### 2.5.1 Derivation of Angular Margin

We can rewrite  $f(\cos \theta_{y_i})$  as

$$\begin{aligned} f(\cos \theta_{y_i}) &= s \cdot (\cos(\theta_{y_i} + m)) \\ &= s \cdot (\cos \theta_{y_i} \cos m - \sin \theta_{y_i} \sin m) \\ &= s \cdot \left( \cos \theta_{y_i} \cos m - \sqrt{1 - \cos^2 \theta_{y_i}} \sin m \right), \end{aligned} \quad (2.22)$$

by the laws of trigonometry. Therefore,

$$\frac{\partial f(\cos \theta_{y_i})}{\partial \cos \theta_{y_i}} = s \left( \cos(m) + \frac{\cos \theta_{y_i} \sin(m)}{\sqrt{1 - \cos^2 \theta_{y_i}}} \right). \quad (2.23)$$

### 2.5.2 Interpretation of $g$

For Softmax and Additive Margin, we see that  $g = (P_{y_i}^{(i)} - 1)s$ . Since the softmax operation in  $P_{y_i}^{(i)}$  has a tendency to scale the result to be close to either 0 or 1, the first term in  $g$ ,  $(P_{y_i}^{(i)} - 1)$  tends to be close to 1 or 0 far away from the decision boundary. In the equation for  $P_{y_i}$ , there is also  $s$  which is a scaling hyper-parameter, and is often set to  $s = 64$  [55, 102, 154, 240]. This high  $s$  makes the softmax operation even steeper near the decision boundary. This results in almost equal GST for samples away from the decision boundary, regardless of how far they are from the decision boundary. This is evident in Fig. 2.7, where the blue curve is flat except near the decision boundary when  $s$  is high.

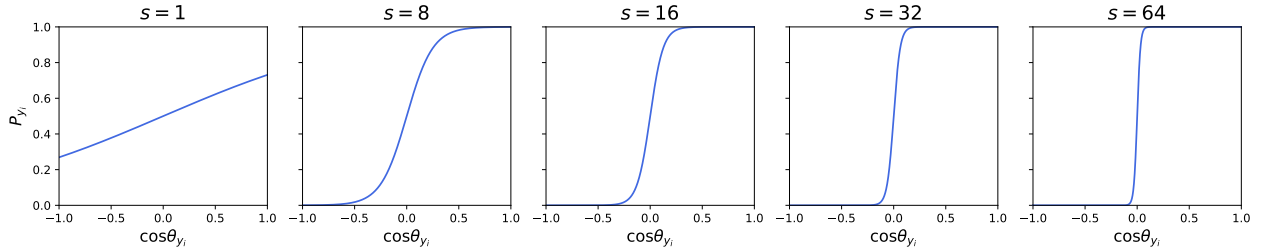


Figure 2.7 Plot of  $P_{y_i}$  for different values of  $s$ . In this figure,  $P_{y_i}$  is calculated with  $f(\cos \theta_j)$  from Softmax (*i.e.*  $m = 0$ ).

For Softmax and Additive Margin,  $\frac{\partial f(\cos \theta_{y_i})}{\partial \cos \theta_{y_i}} = s$ . This term is different for Angular Margin due to  $\frac{\partial f(\cos \theta_{y_i})}{\partial \cos \theta_{y_i}}$  being a function of  $\cos \theta_{y_i}$ . The exact form of  $\frac{\partial f(\cos \theta_{y_i})}{\partial \cos \theta_{y_i}}$  for Angular Margin is found in Eqn. 2.23. As shown in Fig. 2.8, Eqn. 2.23 is monotonically increasing with respect to  $\cos \theta_{y_i}$  when  $m > 0$  and vice versa. Note that  $\cos \theta_{y_i}$  is how close the sample is to the ground truth weight vector, and it is closely related to the difficulty of the sample during training. Therefore, this partial derivative term from the angular margin,  $\frac{\partial f(\cos \theta_{y_i})}{\partial \cos \theta_{y_i}}$ , can be viewed as scaling the importance of sample based on the difficulty.

## 2.6 Feature Norm Analysis

### 2.6.1 Correlation between Norm and BRISQUE during Training

In the Sec. 3.2 of the main paper, we introduce the idea of using the feature norm as a proxy of the image quality. We observe that in models trained with a margin-based softmax

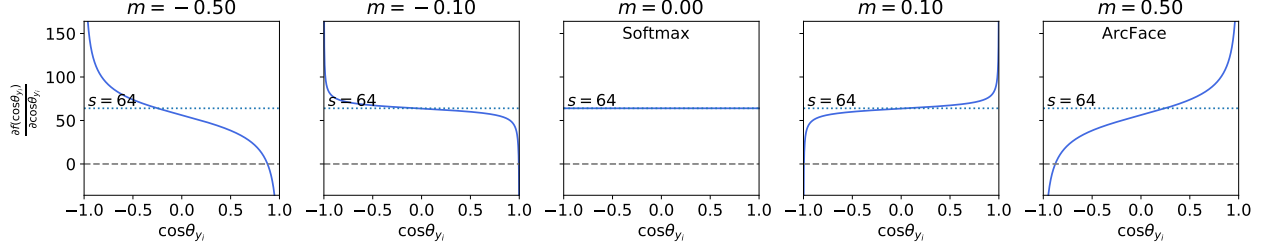


Figure 2.8 Plot of  $\frac{\partial f(\cos \theta_{y_i})}{\partial \cos \theta_{y_i}}$  for different value of  $m$  when the margin function is Angular Margin.

loss, the feature norm exhibits a trend that is correlated with the image quality. Here, we show for ArcFace and AdaFace, both loss functions exhibit this trend, in Fig. 2.9. Regardless of the form of the margin function, the correlation between the feature norm and the image quality is quite similar (green plot in 1st and 2nd columns). We leverage this behavior to design the proxy for the image quality.

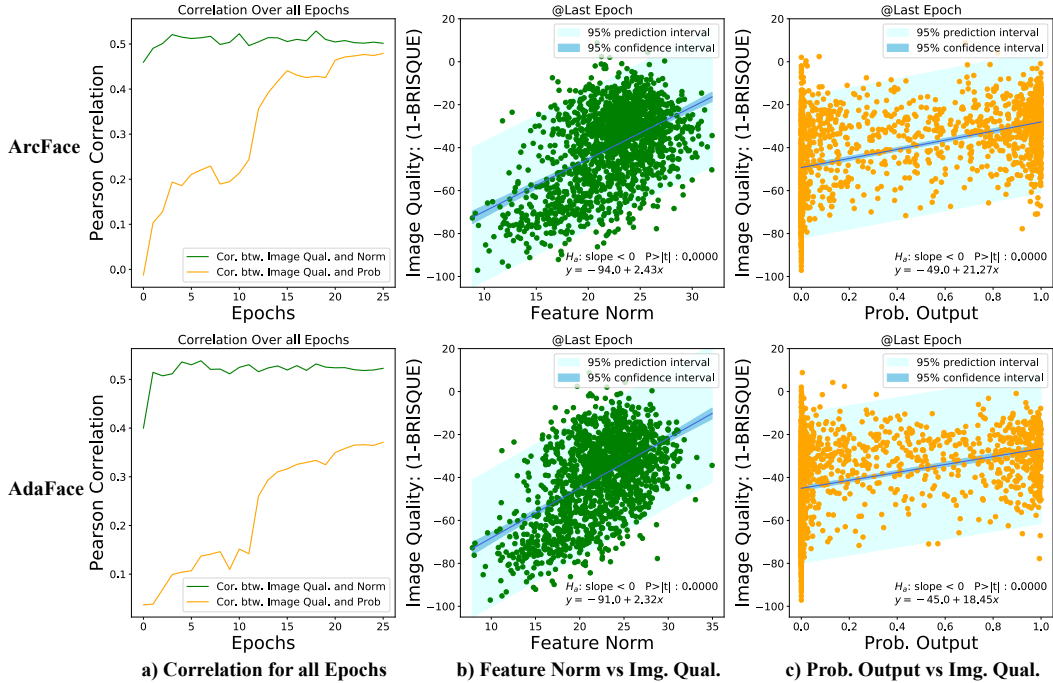
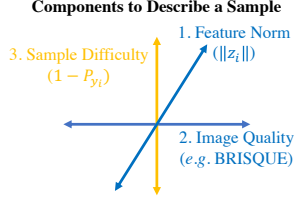


Figure 2.9 Comparison between ArcFace and AdaFace on the correlation between the feature norm and the image quality. We randomly sampled 1,534 images from the training dataset (MS1MV2 [55]) to show this plot.

We use three concepts (image quality, feature norm and sample difficulty) to describe a



Method	Relationship	Gradient Flow to $\ z_i\ $
MagFace [172]	Sample Difficulty vs. $\ z_i\ $	Yes
AdaFace	Image Qual. vs. $\ z_i\ $	No

Figure 2.10 An illustration of different components to describe a sample and their usage in previous works.

sample, as illustrated in Fig. 2.10. We leverage the correlation between the feature norm and the image quality to apply different emphasis to different difficulty of samples. In contrast, MagFace learns a representation that aligns the feature norm with recognizability. The term, *image quality* in MagFace paper [172] refers to the face recognizability, which is closer in meaning to the sample difficulty than the term, image quality, we use in our paper. Please refer to the Fig. 1 (a) and the first contribution claim of the MagFace paper [172]. Also note the difference in gradient flow through the feature norm,  $\|z_i\|$ . MagFace relies on learning the feature that has  $\|z_i\|$  aligned with the recognizability of the sample, requiring the gradient to flow through  $\|z_i\|$  during backpropagation. The loss function has the incentive to reduce the margin by reducing  $\|z_i\|$ . However, our objective is to adaptively change the loss function, itself, so we treat  $\|z_i\|$  as a constant. Finally, from Tab. 3 of our main paper, AdaFace substantially outperforms MagFace, e.g. reducing the errors of MagFace on IJB-B and IJB-C relatively by 21% and 23% respectively.

### 2.6.2 Training Sample Visualization



Figure 2.11 Actual training data examples corresponding to 6 zones. A pretrained AdaFace model is used as a feature extractor.

We show some visualization of the actual training images. From the randomly sampled 1,534 images from the training dataset (MS1MV2 [55]), we divide the samples into 6 different zones. We plot the samples by  $\cos \theta_{y_i}$  (decreasing) as the x-axis and the feature norm  $\|z_i\|$  as y-axis in Fig. 2.11. We divide the plot into 6 zones and sample a few images from each group. Clearly, there are not many samples in the zones highlighted by the gray area (top right and bottom left). This indicates that the sample difficulty distribution is different for each level of feature norm. Furthermore, the samples in the dark green area are mostly unrecognizable images. AdaFace de-emphasizes these samples. Also, the samples in the bright pink area are more difficult samples than the dark pink area. AdaFace puts more emphasis on the harder samples when the feature norm is high. We would like to remind the readers that this figure may serve as an empirical validation of the two-dimensional face image categorization we made in Fig. 1 of the main paper.

### 2.6.3 Training Samples' Gradient Scaling Term for AdaFace

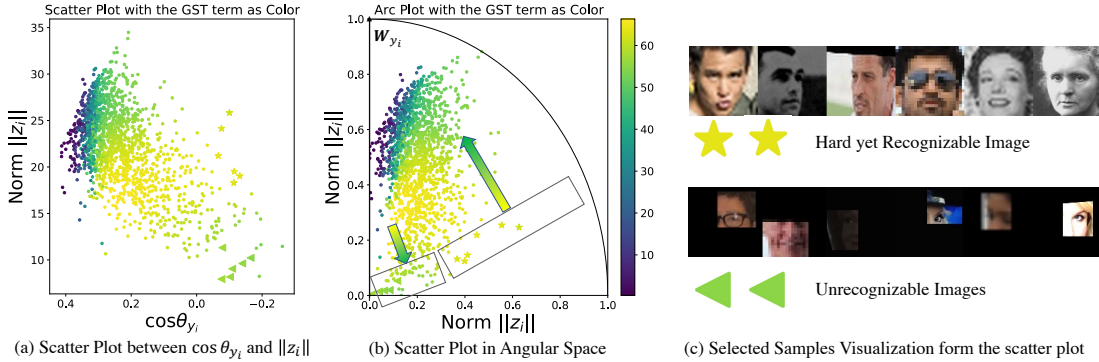


Figure 2.12 (a) Scatter plot of samples from Fig. 2.11 with the color as the GST term. (b): Scatter plot of the same 1,534 points in angular space. For each feature, the angle from  $W_{y_i}$  is calculated from  $\cos \theta_{y_i}$  and the distance from the origin is calculated from  $\|z_i\|$ . Both terms are normalized for visualization. (c): Sample image visualization from the low norm and high norm regions of similar  $\cos \theta_{y_i}$ .

In Fig. 2.12 (a), we plot the actual GST term for AdaFace. We use the same 1,534 images from the training dataset (MS1MV2 [55]) as in Fig. 2.11. The color of points indicates the

magnitude of the GST term. The purple points on the left side of the scatter plot are samples past the decision boundary. Therefore the magnitude of GST term is low. The effective difference in GST term for samples outside the decision boundary can be seen by the color change from green to yellow. Note that AdaFace de-emphasizes samples of low feature norm and high difficulty. This is shown in the lower right region of the plot. In Fig. 2.12 (b), we warp the plot into the angular space to make a correspondence with the Fig. 3 of the main paper, where we illustrate the GST term for AdaFace. We illustrate how actual training samples are distributed in this angular space. In Fig. 2.12 (b) and (c), we visualize two groups of images where one is from the low feature norm area (triangle) and the other is from the high feature norm area (star). AdaFace exploits images that are hard yet recognizable, as indicated by the yellow star regions, and lowers the learning signal from the unrecognizable images, as indicated by the green triangle regions.

#### 2.6.4 Train Samples' Gradient Scaling Term Comparison with ArcFace

In Fig. 2.13, we compare the GST term placed on training samples. We have two groups of images. One group is comprised of unrecognizable images, shown under the red bar. Another group is comprised of hard yet recognizable images, shown under the green bar. Each bar corresponds to one training sample, and the height of the bar indicates the magnitude of the gradient scaling term (GST). For ArcFace shown on the left, the same level of GST is placed on all samples. However, in AdaFace, unrecognizable samples are less emphasized relative to the recognizable samples.

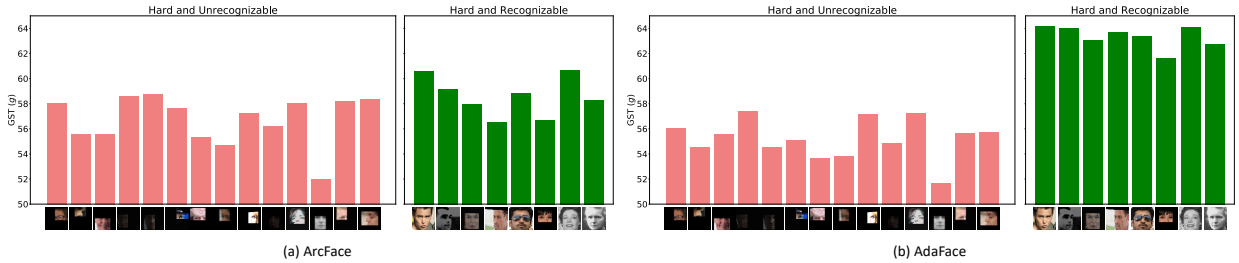


Figure 2.13 Comparison of the magnitude of GST term between ArcFace and AdaFace.



Figure 2.14 Examples from IJB-C [169] dataset, where ArcFace fails to identify the subject whereas AdaFace successfully finds the correct match between the probe and the gallery. On the left is the set of probe images and on the right is the set of gallery images.

## 2.7 Visualization of Success and Failed Test Images

We show samples from IJB-C [169] dataset to show which samples are correctly classified in AdaFace, compared to ArcFace [55]. In each pair of probe and gallery images, we write the rank and the similarity score for both ArcFace and AdaFace. Rank= 1 is the correct match and a high similarity score is desired. Note that the majority of the cases where AdaFace successfully matches the hard samples for ArcFace are comprised of low quality samples. This shows that indeed AdaFace works well on low quality images.

## 2.8 Comparison with General Image-Quality Aware Learning Method

We compare our method with QualNet [120] (CVPR21) as a comparison with general image-quality aware learning method. The scope of general image-quality aware learning methods is not limited to face recognition, but the idea is applicable. In Tab. 2.4, we show the comparison with QualNet with models trained on CASIA-WebFace. AdaFace outperforms QualNet on the TinyFace test set. QualNet aligns the low quality (LQ) image feature distribution to the high quality (HQ) features' distribution via a fixed pretrained decoder. In contrast, AdaFace prevents LQ images from degrading the overall recognition performance by de-emphasizing heavily degraded LQ images. Since LQ facial images can often be devoid of identity, it helps to avoid overfitting on unidentifiable LQ images and learn to exploit the identifiable LQ images. This improves generalization across HQ and LQ.

Method	Training Set	Test set	Rank1	Rank5
QualNet [120]	CASIA-Webface	TinyFace	35.54	44.45
AdaFace			<b>44.39</b>	<b>47.23</b>

Table 2.4 Closed set identification performance (ranked match rate) on TinyFace. For a fair comparison, we adopt the train/test setting of QualNet. QualNet results are directly taken from the CVPR21 paper.

## 2.9 Effect of Batch Size

Our image quality proxy  $\widehat{\|\mathbf{z}_i\|}$  does not depend on the batch size due to the exponential moving average in Eq.17 of the main paper (rewritten below).

$$\widehat{\|\mathbf{z}_i\|} = \left[ \frac{\|\mathbf{z}_i\| - \mu_z}{\sigma_z/h} \right]_{-1}^1, \quad (2.24)$$

$$\mu_z = \alpha \mu_z^{(k)} + (1 - \alpha) \mu_z^{(k-1)}. \quad (2.25)$$

To empirically show this, we train R50 model on MS1MV2 with the batch size of 128, 256 and 512 and report their performance on IJB-B TAR@FAR=0.01%. As shown in Tab. 2.5, the difference due to the batch size is minimal.

Method	Batch size 128	Batch size 256	Batch size 512
AdaFace	94.32	94.42	94.35

Table 2.5 Performance comparison by varying the batch size. This shows that AdaFace performance not subject to different batch sizes.

## 2.10 Implementation Details and Code

The code is released at <https://github.com/mk-minchul/AdaFace>. For preprocessing the training data MS1MV2 [55], we reference InsightFace [1] and InsightFacePytorch [2], for the backbone model definition, TFace [3] and for evaluation of LFW [100], CFP-FP [202], CPLFW [296], AgeDB [174], CALFW [297], IJB-B [253], and IJB-C [169], we use InsightFace [1]. For preprocessing IJB-S [112] and TinyFace [46], we use MTCNN [285] to align faces.

### 2.11 Conclusion

In this work, we address the problem arising from unidentifiable face images in the training dataset. Data collection processes or data augmentations introduce these images in the training data. Motivated by the difference in recognizability based on image quality, we tackle the problem by 1) using a feature norm as a proxy for the image quality and 2) changing the margin function adaptively based on the feature norm to control the gradient scale assigned to different quality of images. We evaluate the efficacy of the proposed adaptive loss on various qualities of datasets and achieve SoTA for mixed and low quality face datasets.

**Limitations** This work addresses the existence of unidentifiable images in the training data. However, a noisy label is also one of the prominent characteristics of large-scale facial training datasets. Our loss function does not give special treatment to mislabeled samples. Since our adaptive loss assigns large importance to difficult samples of high quality, high quality mislabeled images can be wrongly emphasized. We believe future works may adaptively handle both unidentifiability and label noise at the same time.

**Potential Societal Impacts** We believe that the Computer Vision community as a whole should strive to minimize the negative societal impact. Our experiments use the training dataset MS1MV\*, which is a by-product of MS-Celeb [161], a dataset withdrawn by its creator. Our usage of MS1MV\* is necessary to compare our result with SoTA methods on a fair basis. However, we believe the community should move to new datasets, so we include results on newly released WebFace4M [300], to facilitate future research. In the scientific community, collecting human data requires IRB approval to ensure informed consent. While IRB status is typically not provided by dataset creators, we assume that most FR datasets (with the exceptions of IJB-S) do not have IRB, due to the nature of collection procedures. One direction of the FR community is to collect large datasets with informed consent, fostering R&D without societal concerns.

## CHAPTER 3

### CLUSTER AND AGGREGATE: FACE RECOGNITION WITH LARGE PROBE SET

Feature fusion plays a crucial role in unconstrained face recognition where inputs (probes or galleries) comprise of a set of  $N$  low quality images whose individual qualities vary. Advances in attention and recurrent modules have led to feature fusion that can model the relationship among the images in the input set. However, attention mechanisms cannot scale to large  $N$  due to their quadratic complexity and recurrent modules suffer from input order sensitivity. We propose a two-stage feature fusion paradigm, *Cluster and Aggregate*, that can both scale to large  $N$  and maintain the ability to perform sequential inference with order invariance. Specifically, Cluster stage is a linear assignment of  $N$  inputs to  $M$  global cluster centers, and Aggregation stage is a fusion over  $M$  clustered features. The clustered features play an integral role when the inputs are sequential as they can serve as a summarization of past features. By leveraging the order-invariance of incremental averaging operation, we design an update rule that achieves batch-order invariance, which guarantees that the contributions of early image in the sequence do not diminish as time steps increase. Experiments on IJB-B and IJB-S benchmark datasets show the superiority of the proposed two-stage paradigm in unconstrained face recognition. Code and pretrained models are available in [Link](#).

#### 3.1 Introduction

Face Recognition (FR) matches a set of input query imagery, known as *probe*, to enrolled identity database, known as *gallery*. Verification is to confirm the claimed probe’s identity and identification is to identify the unknown probe’s identity by searching a known database [195]. In either case, a probe can go beyond an image and include a set of images, videos, or their combinations [21]. Thus FR involves fusing features of multiple images or videos to create a discriminative feature for a probe.

Due to the interest in unconstrained surveillance scenarios, *e.g.* IJB-S [112], the role of fusion is becoming more important. Unconstrained FR is often based on probes from low

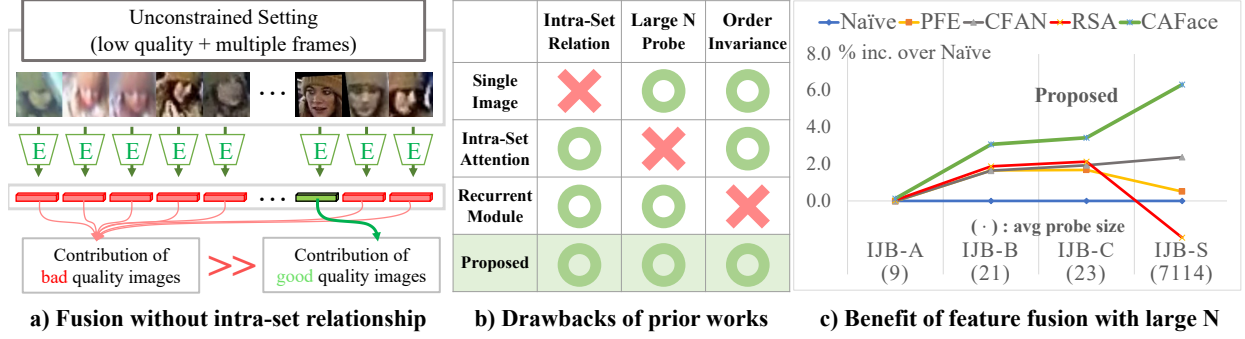


Figure 3.1 a) An illustration of the importance of the intra-set relationship in feature fusion.

Without the intra-set relationship, a large weight on a good quality image can still be outweighed by many bad quality images in a probe set. b) We need a framework that can both account for the intra-set relationship of large  $N$  probes and handle sequential inputs with order invariance. c) The role of fusion model increases with larger probe size. For our proposed method, CAFE, the relative performance gain over Naïve (simple averaging) method, *i.e.*,  $\frac{\text{CAFace-Naïve}}{\text{Naïve}} * 100\%$ , increases with the probe size across four datasets. PFE [209] and CFAN [78] are single-image based and lack intra-set relationship. RSA [156] computes intra-set relationship but unusable for large  $N$ .

quality images and videos. It is challenging due to two issues: 1) individual video frames can be of poor quality, causing erroneous FR model prediction and 2) the number of images in a probe can be very large, *e.g.*, a probe video in IJB-S may have 500,000 frames. Feature fusion across all frames in the probe is especially crucial if frame-based predictions are unreliable. While prior works [122, 172] address the first issue of prediction in low quality images, the large size of probe set was not addressed. Fig. 3.1 a) illustrates the problem caused by the absence of proper feature fusion. The contribution of good quality image can be made insignificant in the presence of many other poor quality images in the set.

This paper aims to learn a fusion function that maps an unordered set of  $N$  probe features  $\{\mathbf{f}_i\}^N$  of the same person to a single fused output  $\mathbf{f}$ . Note that  $\mathbf{f}_i = E(\mathbf{x}_i)$  is the feature extracted from the  $i$ -th sample in the set, using a fixed feature extractor  $E$ . The task of fusing multiple features involves 1) estimating the quality of individual features and

2) modeling the intra-set relationship of the features. Prior feature fusion works utilize either simple average pooling [42, 184], reinforcement learning [157], recurrent models [77] or self-attention [78, 156, 159, 247, 266].

Typically, to compute the intra-set relationship among inputs of an arbitrary size  $N$ , one would adopt set-to-set functions such as Multihead Self Attention (MSA) [156, 236, 247], enabling inputs to propagate information among themselves. The downside of this approach is its computational cost of  $O(N^2)$  which becomes infeasible when  $N$  exceeds a few thousand. Also, when the inputs are sequential as in a live video feed, it is nontrivial to model the intra-set relationship except to compute attention over all past frames at each time step. Recurrent methods [77, 93] are useful in the sequential inference but their drawback is set order inconsistency, *i.e.*, as the number of sequential steps  $T$  increases, the contribution of early frames in a set decreases. Fig. 3.1 b) contrasts various fusion methods.

A feature fusion framework that can consider both 1) intra-set relationship for a large  $N$  and 2) efficient sequential inference is necessary in the real-world unconstrained FR scenarios. Fig. 3.1 c) shows the average probe sizes of four datasets. IJB-S [112]’s probe size is too large for intra-set attention such as RSA [156] to perform inference with all frames concurrently.

We present a feature fusion framework, Cluster and Aggregate (CAFace), that achieves two abovementioned criteria. It consists of two modules: Cluster Network (CN) and Aggregation Network (AGN). CN makes soft assignments of  $N$  features into  $M$  fixed number of clusters, *i.e.*,  $\{\mathbf{f}_i\}^N \rightarrow \{\mathbf{f}'_j\}^M$  where  $M \ll N$ . While  $N$  varies from one set to other,  $M$  is fixed. AGN combines  $M$  clustered features into a single feature  $\mathbf{f}$ , *i.e.*,  $\{\mathbf{f}'_j\}^M \rightarrow \mathbf{f}$ . Conceptually,  $M$  intermediate cluster features serve as a summarization of  $N$  inputs and AGN models the intra-set relationship among  $\{\mathbf{f}'_j\}^M$ .

The proposed framework depends on learning global cluster assignments  $\{\mathbf{f}_i\}^N \rightarrow \{\mathbf{f}'_j\}^M$  that are consistent across different probes. Thus, we propose learning shared cluster centers that are input independent. These centers govern the clustering assignments. But, it is not obvious which clustering criterion is the best for feature fusion. Thus, we design CN to

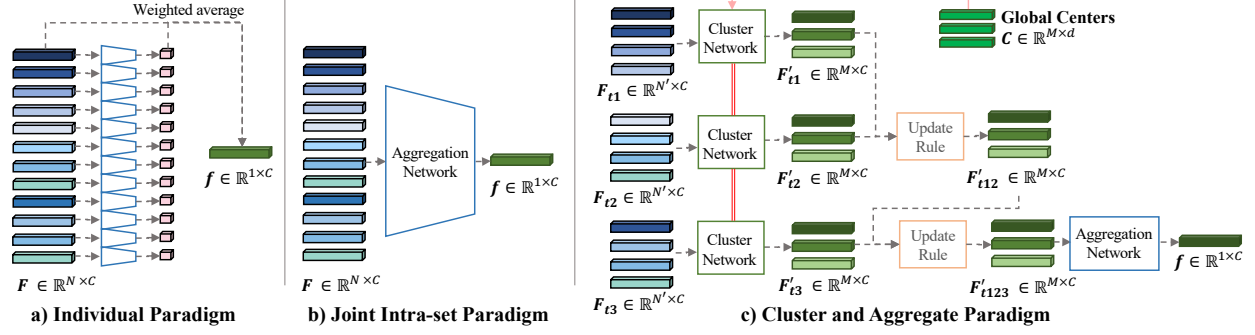


Figure 3.2 Comparison of feature fusion paradigms. a) In the individual paradigm, each probe sample’s weight is determined independently. b) In the intra-set paradigm, the sample weight is determined based on all inputs. However, when  $N$  is large or sequential, intra-set calculations become infeasible. c) In the Cluster and Aggregate paradigm, the intermediate representation  $\mathbf{F}'$  (green) can be updated across batches, allowing for large  $N$  intra-set modeling and sequential inference. Sharing universal cluster centers  $\mathbf{C}$  ensures consistency of  $\mathbf{F}'$  across batches. Unlike RNN, the update rule is batch-order invariant.

discover learned clusters with an end-to-end differentiable framework that allows AGN to back-propagate the gradients to CN. The cluster assignments are learned to maximize the FR performance. We also design an input pipeline, Style Input Maker (SIM) that can helps CN perform class (identity) agnostic clustering efficiently.

The purpose of introducing an intermediate stage  $\{\mathbf{f}'_j\}^M$  is to facilitate the sequential inference. The key design of CN is to formulate  $\{\mathbf{f}'_j\}^M$  as a *linear* combination of  $\{\mathbf{f}_i\}^N$ . This guarantees that even when the input sequence of set length  $N$  is divided into  $T$  smaller batches of set length  $N'$ ,  $\{\mathbf{f}'_j\}^M$  can be sequentially updated with batch-order invariance. This is due to our update rule, inspired by the order invariance property of the averaging operation, as in Eq. 3.8. When the inputs are sequential, we feed only the new features to CN and update the cached  $\{\mathbf{f}'_j\}^M$ . It achieves a similar effect as having used all previous features simultaneously. Fig. 3.2 shows the contrast with previous approaches. For readability, we will interchange the set notation  $\{\mathbf{f}_i\}^N$  with the matrix notation  $\mathbf{F} \in \mathbb{R}^{N \times C}$ .

In summary, the contributions of this paper include:

- A novel feature fusion framework for both large  $N$  feature fusion and efficient sequential inference. To our knowledge, this is the first approach to utilize linearly combined intermediate clusters to achieve batch-order invariance with intra-set relationship modeling.
- An task-driven clustering mechanism that can discover latent clustering centers that maximize the task performance. In our case, the task is FR. We achieve the task-driven clustering with an assignment algorithm using the global query and decoupled key and value structure.
- We show the superiority of CAFace in unconstrained face recognition on multiple datasets.

### 3.2 Related Work

**Feature Fusion (Unordered Set)** The simplest way of feature fusion is to average over a set of features  $\{\mathbf{f}_i\}^N$  [42, 184]. In this case, the features with larger norms play a bigger role, and it generally works since easy samples tend to show larger norms [172, 191]. To learn the weights, CFAN and QAN utilize the self-attention mechanism, a learned weighted averaging mechanism [78, 159]. The drawback of these approaches is the lack of an intra-set relationship during the weight calculation process.

Previous works that adopt the intra-set attention mechanism are Non-local Neural network and RSA [156, 247]. These works use intermediate feature maps  $\mathbf{U}_i$  of size  $\mathbb{R}^{C_M \times H \times W}$  during aggregation because feature maps provide rich and complementary information that can be refined by taking the spatial relationship into account. However, the drawback is in the heavy computation in the attention calculation. For a set of  $N$  features maps, an attention module involves making  $(N \times H \times W)^2$  sized affinity map. Our Cluster Network utilizes a compact style vector from SIM and makes  $N^2$  sized affinity map which greatly increases the computation efficiency in attention computation.

DAC [157] and MARN [77] propose RL-based and RNN-based quality estimators, respectively. Yet, they fail to be agnostic to input order, thus unsuitable for modeling long-range dependencies. Our method can split the  $N$  inputs into  $T$  smaller batches and still achieve batch-order invariance. Lastly, modeling the intra-set relationship with auxiliary context

(such as a body) is shown to be helpful [294].

**Video Recognition (Ordered Set)** The feature fusion for recognition has a resemblance to video-based recognition [155, 298], but set inputs cannot always expect the temporal dependencies to be available. Therefore, most video-based approaches for tasks such as action recognition or quality enhancement [13, 20, 127, 162, 176, 224, 288] focus on exploiting the relationship between nearby frames, whereas feature fusion approaches do not define them. In video-based FR, the general trend is to focus more on assessing the quality of individual frames as opposed to exploring the relationship among nearby frames. Some examples of video-based FR utilize n-order statistics [166], affine hulls [37, 97, 267], SPD matrices [106] and manifolds [84, 244]. Recently, probabilistic representation such as PFE [209] gained popularity [12, 39, 204, 209] since the variance in distribution serves as a quality estimation for individual frames. QSub-PM [293] bypassed the need for a single feature by representing a video with a subspace (matrix) and proposing a novel subspace comparison.

**Attention Mechanism** Multihead Self Attention (MSA) [236] is a widely adopted set-to-set function that models intra-set relationships via an affinity map. It is also a key component in transformer architectures which outperform CNNs in various vision tasks [36, 48, 61, 141, 160, 230, 282]. The underlying mechanism of MSA which uses the affinity of query and key to update the value is versatile in its application beyond recognition and has led to its usage in memory retrieval and grouping [32, 226, 263]. The unique property of the proposed Cluster Network is in the linear combination of value assignment which enables batch-order invariance using an incremental average update rule. Unlike MSA which requires concurrent inputs during inference for intra-set relationship, ours can split the inference and establish a connection across batches without decreasing the contribution of early inputs.

### 3.3 Proposed Approach

The Cluster and Aggregate paradigm seeks to divide the large  $N$  inference into partitioned inferences while still obtaining the result as seeing all inputs at the same time. This can be achieved if 1) each partitioned inference can update the intermediate representation with

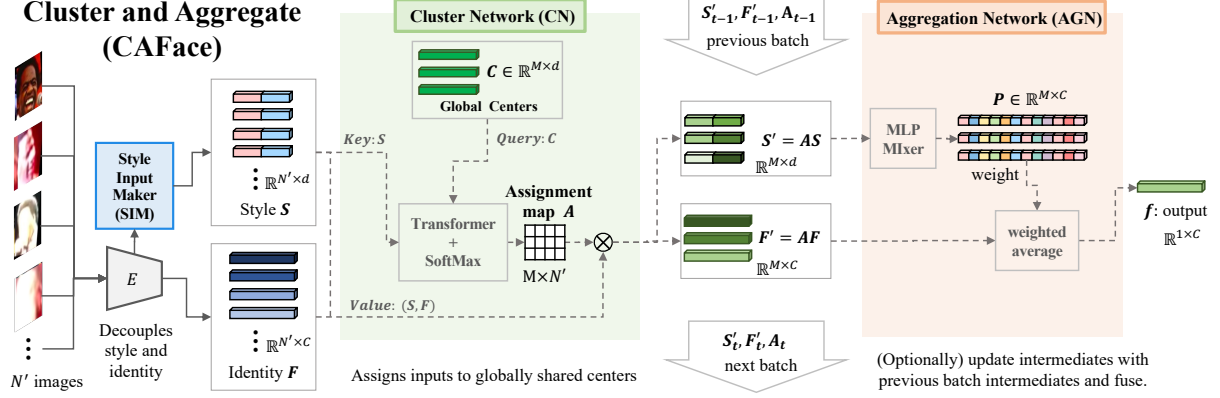


Figure 3.3 An overview of CAFace with cluster and aggregate paradigm. The task is to fuse a sequence of images to a single feature vector  $\mathbf{f}$  for face recognition. SIM is responsible for decoupling facial identity features  $\mathbf{F}$  from image style  $\mathbf{S}$  that carry information for feature fusion (Sec. 3.3.1). Cluster Network (CN) calculates the affinity of  $\mathbf{S}$  to the global centers  $\mathbf{C}$  and produces an assignment map  $\mathbf{A}$ . It will be used to map  $\mathbf{F}$  and  $\mathbf{S}$  to create fixed size representations  $\mathbf{F}'$  and  $\mathbf{S}'$ . Note that  $\mathbf{F}'$  and  $\mathbf{S}'$  are linear combinations of raw inputs  $\mathbf{F}$ ,  $\mathbf{S}$  respectively. This property ensures that the previous and current batch representations can be combined using weighted average, which is order-invariant. Lastly, AGN computes the intra-set relationship of  $\mathbf{S}'$  to estimate the importance of  $\mathbf{F}'$  for fusion. For interpretability, AGN produces the weights for averaging  $\mathbf{F}'$  to obtain  $\mathbf{f}$ .

necessary information and 2) the order of inference does not affect the final outcome, so the information in early batches is not forgotten. In essence, the intermediate representation serves as a communication channel across batches. We achieve this by designing a Cluster Network (CN) and Aggregation Network (AGN). Fig. 3.2 c) shows the proposed paradigm. In this section, we will elaborate on how we obtain the global assignment that is consistent across batches and how the update rule can be batch-order invariant. We formally layout a few assumptions for the Cluster and Aggregate paradigm in the face recognition (FR) task as shown in Fig. 3.3.

Let  $\{\mathbf{x}_i\}^N$  be a set of  $N$  facial images from the same person. The task is to produce a single feature vector  $\mathbf{f}$  from  $\{\mathbf{x}_i\}^N$  that is discriminative for the recognition task. We

assume that a single image based pretrained face recognition model  $E : \mathbf{x}_i \rightarrow \mathbf{f}_i$  is available following the settings of previous works [78, 156, 209]. For readability, we will interchange the set notation  $\{\mathbf{f}_i\}^N$  with the matrix notation  $\mathbf{F} \in \mathbb{R}^{N \times C}$  where  $\{\mathbf{f}_i\}^N$  is the input is a set of length  $N$  and  $\mathbf{F}$  simplifies equations. For clarity, we denote  $N$  to be the probe size (the number of images in a set) and  $N'$  to be the partitioned set size when  $N$  is large. During training, we fix the number of images for fusion as  $N'$ . Note that the shape of inputs during training would have one more dimension, training batch size  $B$ , *i.e.*  $\mathbf{F} \in \mathbb{R}^{B \times N' \times C}$ . Training batch size refers to the number of persons sampled in a mini-batch, different from the number of images per person,  $N'$ . We drop the training batch size dimension in equations for brevity.

### 3.3.1 Architecture

**Cluster Network (CN)** Cluster Network is responsible for mapping inputs  $\mathbf{F} \in \mathbb{R}^{N' \times C}$  of variable size  $N'$  to  $\mathbf{F}' \in \mathbb{R}^{M \times C}$  of a fixed size,  $M$ . A natural choice for the architecture would be Transformer [61, 236] as it is a set to set function. However, there are two problems with it. 1) It cannot handle large inputs due to the quadratic complexity of MSA. 2) When the inputs are partitioned and inferred sequentially, the intra-set information across the batch is lost, as MSA computes the affinity within the given inputs. CN solves this problem by modifying Transformer with 1) shared queries and 2) linear value mapping. These changes result in a clustering mechanism.

We first consider the following generic attention equation [236] with query  $\mathbf{Q}$ , key  $\mathbf{K}$  and value  $\mathbf{V}$ .

$$\text{Attn}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{SoftMax}_{\text{row}} \left( \frac{\mathbf{Q}\mathbf{W}_q (\mathbf{K}\mathbf{W}_k)^\top}{\sqrt{d}} \right) \mathbf{W}_v \mathbf{V}, \quad (3.1)$$

where  $\mathbf{W}_q, \mathbf{W}_k, \mathbf{W}_v$  are learnable weights and  $d$  is the channel dimension of  $\mathbf{K}$ . The row-wise Softmax ensures that the output is the weighted average of all projected values  $\mathbf{W}_v \mathbf{V}$  for each query index. We modify this to

$$\text{Assign}_C(\mathbf{K}, \mathbf{V}) = \text{SoftMax}_{\text{col}} \left( \frac{\mathbf{C}\mathbf{W}_q (\mathbf{K}\mathbf{W}_k)^\top}{\sqrt{d}} \right) \mathbf{V} = \mathbf{A}\mathbf{V}. \quad (3.2)$$

First, unlike  $\mathbf{K}$  and  $\mathbf{V}$  which are inputs, the query is now a shared learnable parameter  $\mathbf{C}$  initialized at the beginning of training. Secondly, removing  $\mathbf{W}_v$  and the column-wise Softmax

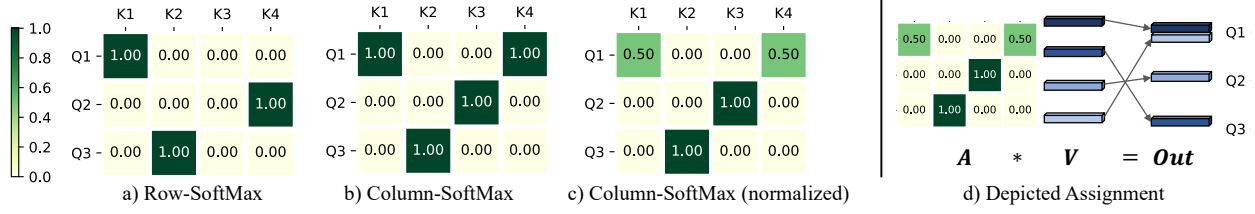


Figure 3.4 a) Row-SoftMax. The sum across the row should be 1.0. b) Column SoftMax. Each column sums to 1.0. c) Column SoftMax with row normalization (Eq. 3.3). d)

Depiction of how values are assigned to centers when  $A$  is multiplied to  $V$ . The matrix is deliberately made sparse for visualization but it can be soft-assignments.

ensure that for each query index of  $C$ , the output is the (soft) selection of values  $V$ . These two modifications result in a learned soft assignment mechanism where  $C$  serves as the global shared center. We name the assignment map as  $A$ . The difference between  $A$  from row and column SoftMax is shown in Fig. 3.4 a) and b). We then divide  $A$  by the weight of samples assigned to each center (row-sum of  $A$ ) as in

$$\text{Cluster}_C(K, V) = \frac{A}{\sum_j A_{i,j}} V, \quad \text{CN}(K, V) = \text{Cluster}_C(\text{Transformer}([K, C]), V). \quad (3.3)$$

Note that  $\text{Cluster}_C(K, V)$  is linear in  $V$ , while the prediction of  $A$  is nonlinear. To further add the nonlinearity of  $A$  to the Cluster Network, we first embed the keys  $K$  with shallow Transformer before clustering. The combined result  $\text{CN}(\cdot)$  is the learned soft assignment of values according to the affinity between keys and global queries.

**Style Input Maker (SIM)** So far, we have discussed the generic Cluster Network algorithm. For face recognition, we still need to decide keys  $K$  and values  $V$  for feature fusion. It is clear that  $V$  should be  $F$ , the facial identity features, as it is what we are interested in merging. It is possible to use  $F$  for  $K$  as well, but  $K$  should ideally contain useful information for fusion and be compatible with queries which are the global center  $C$ . However,  $f_i$  is optimized to be invariant to any characteristics other than the identity. Thus it lacks input image style which encompasses various image traits such as brightness, contrast, quality, pose, or a domain differences from the training data.

In light of the success of using first and second-order feature statistics as an image style [116, 136, 175], we propose SIM for extracting style information using the intermediate representations of feature extractors. The benefit of modeling keys  $\mathbf{K}$  in clustering with *style* over using *identity* is shown in Sec. 3.4.2. We also ablate the benefit of further including feature norm  $\|\mathbf{f}_i\|$  in  $\mathbf{K}$  as it is sometimes used to approximate the confidence of the prediction [122, 172].

Let  $\mathbf{U}_i \in \mathbb{R}^{C_M \times H \times W}$  be the intermediate feature. We capture image style by a style vector  $\boldsymbol{\gamma}_i \in \mathbb{R}^{64}$

$$\boldsymbol{\gamma}_i = \text{BatchNorm}(\text{FC}(\text{ReLU}(\text{AvgPool}(\mathbf{W}_s \odot \boldsymbol{\Gamma})))) \quad (3.4)$$

where  $\boldsymbol{\Gamma} = [\boldsymbol{\mu}_{\text{sty}}, \boldsymbol{\sigma}_{\text{sty}}]$ ,  $\boldsymbol{\mu}_{\text{sty}} = \text{SpatialMean}(\mathbf{U}_i)$ ,  $\boldsymbol{\sigma}_{\text{sty}} = \text{SpatialStd}(\mathbf{U}_i)$ .

A learnable matrix  $\mathbf{W}_s \in \mathbb{R}^{C_M \times 2}$  controls the importance of  $\boldsymbol{\mu}_{\text{sty}}$  and  $\boldsymbol{\sigma}_{\text{sty}}$  via element-wise multiplication  $\odot$ . Simply put, SIM is a shallow network on spatial mean and standard deviation of  $\mathbf{U}_i$ . One can take  $\mathbf{U}_i$  from more than one intermediate locations and in such a case, we concatenate them.

To verify whether the feature norm would further benefit the fusion process, we embed the feature norm  $\|\mathbf{f}_i\|_2$  to a 64-dim vector, following the convention of Sinusoidal conversion [236], which is analogous to the position embeddings in ViT [61]. The norm embedding  $\mathbf{n}_i$  is a 64-dim vector. Finally, the output SIM is the concatenation,  $\mathbf{s}_i = [\boldsymbol{\gamma}_i, \mathbf{n}_i]$  where  $\mathbf{s}_i \in \mathbb{R}^d$  and  $d = 64 + 64 = 128$ . For readability, we denote the set  $\{\mathbf{s}_i\}^{N'} \in \mathbb{R}^{N' \times 128}$  as  $\mathbf{S}$ .

In summary, we decouple style  $\mathbf{S}$  and identity  $\mathbf{F}$  and use  $\mathbf{S}$  as keys to map

$$\mathbf{F}' = \text{CN}(\text{key} = \mathbf{S}, \text{value} = \mathbf{F}), \quad \mathbf{S}' = \text{CN}(\text{key} = \mathbf{S}, \text{value} = \mathbf{S}), \quad (3.5)$$

which are the intermediates that will be used for subsequent fusion in AGN or stored for sequential inference. We also map  $\mathbf{S}$  to  $\mathbf{S}'$  using the same assignment. Fig. 3.3 shows the overall diagram.

**Aggregation Network (AGN)** The Aggregation Network is responsible for fusing a fixed number of  $M$  inputs,  $\mathbf{F}'$  and  $\mathbf{S}'$  into a single fused output  $\mathbf{f}$  with intra-set relationship. We

adopt MLP-Mixer [229] as it can efficiently propagate information for the fixed-size input. For interpretability, we predict weights  $\mathbf{P} \in \mathbb{R}^{M \times C}$  that combines  $\mathbf{F}'$  to  $\mathbf{f} \in \mathbb{R}^C$ . Specifically,  $\mathbf{f} = \text{AGN}(\mathbf{S}', \mathbf{F}')$  is

$$\mathbf{f} = \sum_M \mathbf{P} \odot \mathbf{F}', \quad \mathbf{P} = \text{SoftMax}(\text{MLPMixer}([\mathbf{S}', \mathbf{C}]]), \quad (3.6)$$

where  $[\mathbf{S}, \mathbf{C}]$  denotes the concatenation along the channel dimension. The magnitude of  $\mathbf{P}$  is an interpretable quantity showing the importance of each cluster during fusion. The final output  $\mathbf{f}$  is a weighted average of  $\mathbf{F}'$  whose weight is  $\mathbf{P}$ .

**Sequential Inference** A key characteristic of CAFace is its ability to divide the inputs into  $T$ -step sequential inference of smaller set length  $N'$  when  $N$  is large, and still achieve similar results as the concurrent inference. It is possible as the intermediates  $\mathbf{F}'$  and  $\mathbf{S}'$  are linear combinations of  $\mathbf{F}$ ,  $\mathbf{S}$  respectively, although estimating the combination weights  $\mathbf{A}$  is non-linear. This allows us to formulate the update rule as the incremental weighted average whose innate property is order-invariant.

Consider partitioned inputs  $\mathbf{F}_1, \dots, \mathbf{F}_T$ , with corresponding predicted weights  $\mathbf{A}_1, \dots, \mathbf{A}_T$ . Since by definition (Eq. 3.5),  $\mathbf{F}'_t = \mathbf{A}_t \mathbf{F}_t / \sum_j^{N'} \mathbf{A}_{t,(i,j)}$ , we can write the cumulative intermediate,  $\widehat{\mathbf{F}}'_T$  as

$$\widehat{\mathbf{F}}'_T = \frac{\mathbf{A}_1 \mathbf{F}_1 + \dots + \mathbf{A}_T \mathbf{F}_T}{\sum_{j=1}^{N'} \sum_{t=1}^T \mathbf{A}_{t,(i,j)}}. \quad (3.7)$$

This formulation requires storing all inputs of timestep  $1, \dots, T$ . We can easily convert this to

$$\widehat{\mathbf{F}}'_T = \frac{\mathbf{a}_{T-1} \widehat{\mathbf{F}}'_{T-1} + \sum_{j=1}^{N'} \mathbf{A}_{T,(i,j)} \mathbf{F}_T}{\mathbf{a}_{T-1} + \sum_{i=1}^{N'} \mathbf{A}_{T,(i,j)}}, \quad \text{where} \quad \mathbf{a}_{T-1} = \sum_{t=1}^{T-1} \sum_{i=1}^{N'} \mathbf{A}_{t-1,(i,j)} \quad (3.8)$$

which requires storing only the cumulative row-summed assignment map  $\mathbf{a}_{T-1}$  and a cumulative intermediate  $\widehat{\mathbf{F}}'_{T-1}$  of the previous time-step. The same logic applies to  $\mathbf{S}'$  as well. Note that this operation, by design, is invariant to inference order (batch-order) as the final result will always be the total weighted average. However, we do not obtain element-wise permutation invariance as the prediction of  $\mathbf{A}_t$  will change with different inputs. We test the susceptibility to element-wise permutation in Sec. 3.4.3 and it has minimal impact on the overall performance.

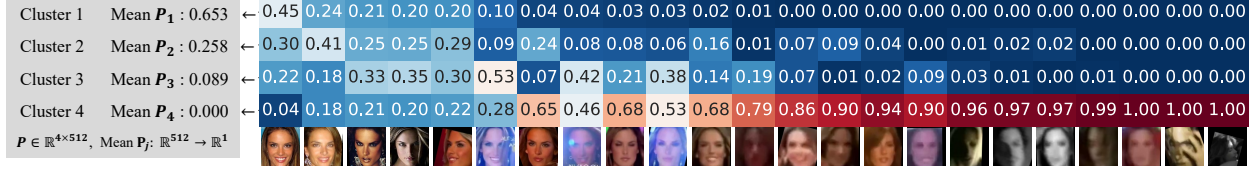


Figure 3.5 A plot of assignment map  $\mathbf{A} \in \mathbb{R}^{4 \times 23}$  (right) and the mean of cluster weights  $\mathbf{P}$  (left) for samples in IJB-B [253]. For each column in  $\mathbf{A}$ , the values sum up to 1.0.  $\mathbf{A}$  shows that 1) high quality images are assigned to clusters 1, 2 and 3, with large mean cluster weights  $\mathbf{P}$ ; low quality images are assigned to cluster 4 with near 0.0 weight. 2) There are variations among clusters 1, 2 and 3 as to which images have more influence, e.g., cluster 3 focuses on relatively blurred or occluded images.

### 3.3.2 Loss Function

**Template Loss** The objective is to make the fused output  $\mathbf{f}$  be close to the ground truth class center  $\mathbf{f}_{GT}$ . The task can be viewed as correctly inferring the true class center in the presence of low quality features  $\{\mathbf{f}_i\}^N$ . Here  $\mathbf{f}_{GT}$  is dependent on the pretrained feature extractor  $E$  and can be either taken from the last FC layer of  $E$  or computed using per-subject average of the embeddings  $\mathbf{f}_i$  in the training data. Our loss function can be viewed as the cosine distance version of the Center loss [251].

In training, we randomly sample  $B$  number of subjects and  $N'$  images per subject. Let the superscript in  $\mathbf{F}^{(b)}$  denote the  $b$ -th subject. The loss to increase the cosine similarity is,

$$\mathcal{L}_t = \frac{1}{B} \sum_{b=1}^B \left( 1 - \text{CosSim} \left( \text{AGN}(\mathbf{S}'^{(b)}, \mathbf{F}'^{(b)}), \mathbf{f}_{GT}^{(b)} \right) \right). \quad (3.9)$$

**Set Permutation Consistency Loss** The Cluster And Aggregate paradigm achieves batch-wise order invariance by formulation, but it does not achieve element-wise permutation invariance, as noted in Sec. 3.3.1. Therefore, we explore the element-wise permutation's added benefit using an additional loss function. Thus, the set permutation consistency loss  $\mathcal{L}_p$  is

$$\mathcal{L}_p = \frac{1}{B} \sum_{b=1}^B \left( 1 - \text{CosSim} \left( \text{AGN}(\mathbf{S}'^{(b)}, \mathbf{F}'^{(b)}), \text{AGN}(\widehat{\mathbf{S}}_T'^{(b)}, \widehat{\mathbf{F}}_T'^{(b)}) \right) \right). \quad (3.10)$$

It lets the splitted inference outcome similar to the concurrent inference. Sec. 3.4.2 shows the benefit of  $\mathcal{L}_p$  but it is small, meaning the batch-order invariance from model design is already powerful. The final loss is

$$\mathcal{L} = \mathcal{L}_t + \lambda_p \mathcal{L}_p. \quad (3.11)$$

where  $\lambda_p$  is the scaling terms for  $\mathcal{L}_p$  respectively.

## 3.4 Experiments

### 3.4.1 Datasets and Implementation Details

We use WebFace4M [300] as our training dataset. It is a large-scale dataset with 4.2M facial images from 205,990 identities. The single image based pretrained face recognition model  $E$  has been trained with the whole training dataset. To train the aggregation module, we use its randomly sampled subset, consisting of 813,482 images from 10,000 identities. We do not use VGG-2 [33] or MS1MV2 [55, 82] as they were withheld by their distributors due to privacy and other issues.

For the pretrained face recognition model  $E$ , we use the IResNet-101, trained with ArcFace loss [55]. Since the performance of the aggregation depends on the quality of  $E$ , we set  $E$  to be the same for *all* experiments.  $E$  produces an embedding vector  $\mathbf{f}_i \in \mathbb{R}^{512}$  for each image. To offer variations in the training data features, we randomly augment the dataset with cropping, blurring, and photometric augmentations.

We test on IJB-B [253], IJB-C [169] and IJB-S [112] datasets. IJB-B is a widely used FR test set containing both high-quality images and low-quality videos of celebrities (see Fig. 3.5 for examples). IJB-C is an updated version of IJB-B with more complex motions in the video. IJB-S is a surveillance video dataset, benchmarking extremely low-quality image/video face recognition. The probe and gallery set size can exceed 500,000. Within the set, there are many low-quality images, making IJB-S very challenging and suitable for measuring the feature fusion framework (see Fig. 3.6 for examples).

For IJB-S, we use protocols, Surv.-to-Single, Surv.-to-Booking and Surv.-to-Surv. The first/second word in the protocol refers to the probe/gallery image source. Surv. is the

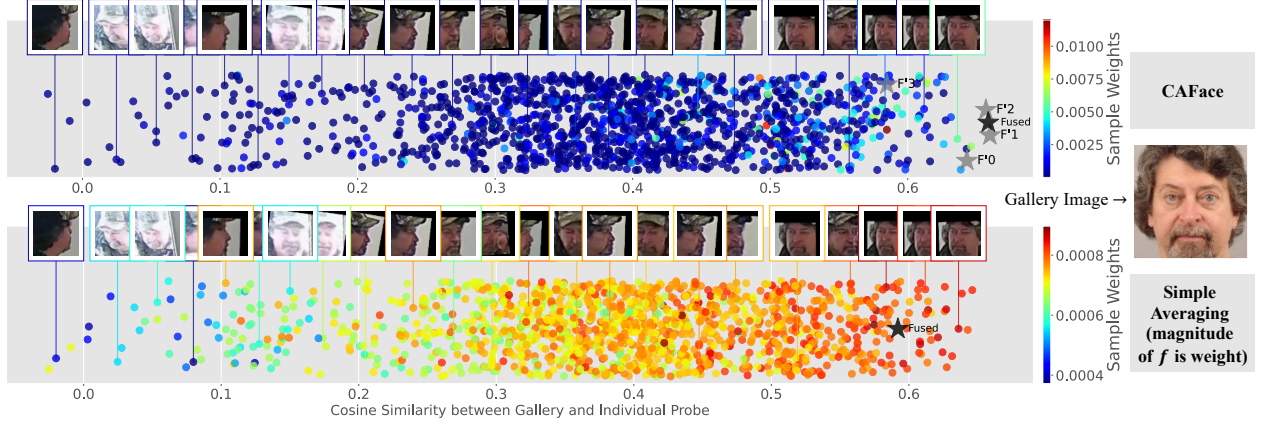


Figure 3.6 A plot of similarity of fused probe vs gallery. The circles represent individual probe images in IJB-S. The colors represent the contribution of each image during fusion. (top: CAFace, bottom: averaging). The x-axis is the cosine similarity with the gallery feature (closer to right: better the match). The black star represents the fused feature. For CAFace, we also plot 4 intermediate features  $\mathbf{F}'$  that go into AGN. (1) Compared to averaging scheme, in CAFace, only a select few are contributing to fusion (few red). Since most samples are low quality, a sparse selection of samples lead to better result. (2) Note that, out of 4 intermediates,  $\mathbf{F}'_3$  falls behind others. It is because CN tends to assign bad quality samples to one cluster, *e.g.*  $\mathbf{C}_3$ .

surveillance video, ‘Single’ is the frontal high-quality enrollment image and ‘Booking’ refers to the 7 high-quality enrollment images. For the ablation study, Sec. 3.4.2, we report the average of all 9 metrics listed in Tab. 3.4.

### 3.4.2 Ablation and Analysis

**Effect of Different Style Input** In Tab.3.1, we ablate the efficacy of various components in SIM which prepares the input for CN. The table shows that using  $\mathbf{f}_i$  for clustering is harmful to performance and increases the no. of parameters for CN. It also shows that using the additional norm embedding  $\mathbf{n}_i$  along with  $\mathbf{s}_i$  produces the best results for IJB-B and IJB-S datasets. However, the margin is small, so simply using  $\mathbf{s}_i$  would suffice for the setting that requires faster computation.

**Effect of  $\mathcal{L}_p$**  As noted in Sec. 3.3.2, we further propose to constrain the set permutation

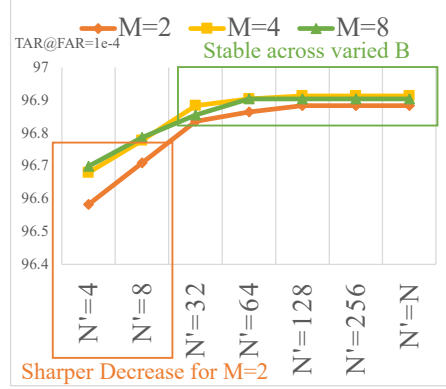


Figure 3.7 A plot of IJB-B performance of CAFace with varied temporal batch size  $N'$  for models with different numbers of clusters  $M$ .

$f_i$	$s_i$	$n_i$	# of Centers ( $M$ )	$\mathcal{L}_p$	# of Params	IJB-B (TAR@FAR=1e-3)	IJB-B (TAR@FAR=1e-4)	IJB-S (AVG)
✓	✓	✓	4	✓	16.28M	96.11	94.38	53.83
×	✓	×		✓	0.25M	96.81	95.53	57.42
×	✓	✓		✓	0.79M	<b>96.91</b>	<b>95.53</b>	<b>57.55</b>
×	✓	✓	4	✓	0.79M	<b>96.91</b>	<b>95.53</b>	<b>57.55</b>
×	✓	✓		×		96.86	95.52	57.36
×	✓	✓	1	✓	0.7860M	96.10	94.31	53.87
×	✓	✓	2	✓	0.7862M	96.88	95.52	57.11
×	✓	✓	4	✓	0.7865M	<b>96.91</b>	95.53	<b>57.55</b>
×	✓	✓	8	✓	0.7874M	96.90	<b>95.61</b>	57.33
Naive Average Fusion					0	96.10	94.30	54.12

Table 3.1 Ablation of varied inputs, loss functions and the number of centers.

consistency with the additional loss  $\mathcal{L}_p$ . The ablation between  $\lambda_p = 0$  and  $\lambda_p = 1$  is shown in Tab.3.1.

**Effect of Number of Clusters** In Tab.3.1, the effect of the number of clusters  $M$  is shown.

The IJB-S performance peaks when  $M = 4$ . When  $M = 2$ , the summary representations  $\mathbf{F}'$  and  $\mathbf{S}'$  have only two assignment options where one cluster takes the poor quality images with low weights. The behavior could be interpreted as performing an outlier detection, which is powerful enough to give high performance. When  $M > 2$ ,  $\mathbf{F}'$  and  $\mathbf{S}'$  have the capacity to store richer history of previous frames which would be beneficial in sequential inference. IJB-S has large  $N$  probes which require dividing the inference into batches. Higher IJB-S performance when  $M = 4$  indicates that the freedom to assign samples to different clusters is important in the sequential setting. A similar phenomenon is observed in Fig. 3.7. The performance gap widens for  $M = 2$  as we reduce the batch size to make more sequential steps

in inference.

**Weight Visualizations** An example of clustering assignments when  $M = 4$  could be viewed in Figs. 3.5 and 3.6. Fig. 3.5 shows how samples are soft-assigned to different clusters along with the weight estimation. The cluster weight is calculated by averaging along 512 dimensions of  $\mathbf{P} \in \mathbb{R}^{M \times 512}$ . Note that each column sums to 1 but  $\mathbf{F}'$  and  $\mathbf{S}'$  are calculated by averaging each row. Thus, the relative contribution of samples in each row is important. Fig. 3.6 shows the actual contribution of each sample during fusion. The contribution can be calculated by multiplying the magnitudes of  $\mathbf{A}$  and  $\mathbf{P}$ . Note that in the presence of many poor quality images, selecting a few good ones is very important and the sample weight of our method can effectively select a subset of samples during fusion.

### 3.4.3 Comparison with SoTA methods

To compare with prior feature aggregation methods, we use the same feature extractor  $E$  as in Sec. 3.4.1, for a fair comparison. Average is the conventional embedding  $\mathbf{f}_i$  averaging scheme that is adopted in the absence of a learned aggregation model. It is equivalent to the stand-alone ArcFace model performance. The rest of the methods learn an additional network for fusing the set of features.

In Tab. 3.2, we show the performance of various feature fusion methods on IJB-B. CAFace achieves a large performance gain in all TAR@FAR metrics. CFAN [78] and PFE [209] do not use any intra-set relationship, as they learn to predict the confidence of a single image. RSA [156] calculates intra-set attention of feature maps, which is computationally costly and incapable of sequential inference. CAFace obtains the best results with the least number of parameters. In Tab. 3.3, the performance in IJB-C dataset is also shown with the similar observation as in IJB-B. We also include an additional backbone, AdaFace [122] to highlight how CAFace can work across different backbones.

In Tab. 3.4, we compare feature aggregation models in the IJB-S dataset that has large  $N$  low quality images/videos in probes. RSA [156] cannot load all images in the probes concurrently for large  $N$ . As an alternative, we divide the probes into a manageable size of

Method	# of Params	Intra-set Att	Seq. Inference	FPS $\uparrow$	TAR@FAR=1e-3 $\uparrow$	TAR@FAR=1e-4 $\uparrow$	TAR@FAR=1e-5 $\uparrow$
Average	0	$\times$	$\checkmark$	-	96.10	94.30	89.53
PFE [209]	13.37M	$\times$	$\checkmark$	360.1 $\times$	96.37	94.82	91.02
CFAN [78]	12.85M	$\times$	$\checkmark$	<b>554.1<math>\times</math></b>	96.43	94.83	91.10
RSA [156]	2.62M	$\checkmark$	$\times$	3.1 $\times$	96.41	95.00	91.22
CAFace	0.79M	$\checkmark$	$\checkmark$	64.4 $\times$	<b>96.91</b>	<b>95.53</b>	<b>92.29</b>

Table 3.2 A performance comparison of recent methods on the IJB-B [253] dataset.

IJB-C [169]	Dataset		Backbone $E$		TAR@FAR=1e-3	TAR@FAR=1e-4	TAR@FAR=1e-5
Naive Average	WebFace4M	300	IResNet101+ArcFace	[55]	97.30	95.78	92.60
PFE [209]	WebFace4M	300	IResNet101+ArcFace	[55]	97.53	96.33	94.16
CFAN [78]	WebFace4M	300	IResNet101+ArcFace	[55]	97.55	96.45	94.40
RSA [156]	WebFace4M	300	IResNet101+ArcFace	[55]	97.49	96.49	94.58
CAFace	WebFace4M	300	IResNet101+ArcFace	[55]	<b>97.99</b>	<b>97.15</b>	<b>95.78</b>
Naive Average	WebFace4M	300	IResNet101+AdaFace	[122]	97.63	96.42	94.47
CAFace	WebFace4M	300	IResNet101+AdaFace	[122]	<b>98.08</b>	<b>97.30</b>	<b>95.96</b>

Table 3.3 A performance comparison of recent methods on the IJB-C [131] dataset. CAFE achieves the best result in IJB-C dataset. We also compare two different backbones ArcFace [55] and AdaFace [122].

$N' = 256$  and average the results. Since RSA does not have a sequential update mechanism, dividing large  $N$  probes reduces the performance, which shows why the sequential capacity is important. CAFE also divides the probes image into batch size of  $N' = 256$  images yet achieves a large margin improvement in IJB-S. It shows that our two-stage mechanism is very effective in the large  $N$  setting. Particularly, the performance gain in the hardest protocol, Surveillance to Surveillance, is the largest. We also randomly shuffle images within the probes 5 times and measure the mean and std. of the performance in the last row. The result shows that our model is robust to input ordering. We also include an experiment on a high quality image dataset, IJB-A [131] later, and note that the performance gain with feature fusion is negligible. As noted in Fig. 3.1 c), the improvement over the baseline (averaging) goes up with the increased number of images in the probe, which highlights the importance of large  $N$  scalability.

### 3.4.4 Resource and Computation Efficiency

Since CAFE is build on top of a single image feature extractor  $E$ , we show the relative FPS of CAFE with respect to the FPS of  $E$  in Tab. 3.2. The relative FPS reported in Tab. 3.2 is computed with the input sequence length  $N = 256$ . It shows that the single image

Method	Surveillance-to-Single			Surveillance-to-Booking			Surveillance-to-Surveillance		
	Rank-1	Rank-5	1%	Rank-1	Rank-5	1%	Rank-1	Rank-5	1%
Naive Average	69.26	74.31	57.06	70.32	75.16	56.89	32.13	46.67	5.32
PFE [209]	69.50	74.39	57.51	70.53	75.29	57.98	32.27	46.70	5.41
CFAN [78]	70.00	74.58	57.93	70.90	75.58	58.09	31.66	45.59	5.79
RSA [156]	63.04	67.33	51.62	63.54	68.23	51.89	16.82	31.80	0.75
CAFace	<b>71.61</b>	<b>76.43</b>	<b>62.21</b>	<b>72.72</b>	<b>77.41</b>	<b>62.68</b>	<b>36.51</b>	<b>49.59</b>	<b>8.78</b>
CAFace (Random Order)	71.65 $\pm 0.05$	76.37 $\pm 0.04$	62.27 $\pm 0.11$	72.77 $\pm 0.04$	77.37 $\pm 0.03$	62.70 $\pm 0.06$	36.43 $\pm 0.08$	49.40 $\pm 0.05$	8.89 $\pm 0.03$

Table 3.4 A performance comparison of recent methods on the IJB-S [112] dataset.

	Max $N$	$N = 16$	$N = 32$	$N = 64$	$N = 256$	$N = 512$
PFE	115, 200	21.8x	44.1x	86.3x	360.1x	2133.6x
CFAN	115, 200	82.6x	158.7x	268.8x	544.1x	664.2x
<b>CAFace</b>	<b>12,000</b>	<b>4.2x</b>	<b>8.2x</b>	<b>16.4x</b>	<b>64.4x</b>	<b>129.3x</b>
RSA	384	6.9x	13.1x	9.2x	3.1x	OOM

Table 3.5 A table of relative FPS of the fusion model with respect to the FPS of the backbone. We compare various fusion models with varied input size  $N$ . As  $N$  increases, it requires more GPU memory as well. Max  $N$  in the second column refers to the maximum number of images that can be in a set without causing the out of memory error (OOM). The third to the seventh columns represent the relative FPS under different set length  $N$ . The higher the relative FPS, the faster the fusion method.

based quality estimation methods, PFE and CFAN are the fastest. And RSA with intra-set attention is the slowest. CACFace can achieve a relatively good speed and obtain the best performance.

Another aspect of computation requirement is GPU memory usage. In the second column of Tab. 3.5, we show the maximum sequence length  $N$  that each method can take simultaneously to perform the feature fusion. It shows that RSA with the inner-set attention cannot handle a sequence length  $N$  larger than 384. This is a drawback that prevents the method from fusing large  $N$  features. On the other hand, CACFace can take a large  $N$  sequence upto 12,000 simultaneously. Note that for the sequence length larger than this can still be handled because CACFace has a sequential inference scheme as described in Sec. 3.3.1. In other words, we can divide the input into smaller set size  $N'$ , and the intermediate representation is updated to account for all elements in a set. Tab. 3.5 also shows the relative FPS of the fusion model compared to the backbone FPS under different sequence lengths  $N$ .

### 3.5 Implementation Details

To train the fusion network  $F$  which is comprised of SIM, CN and AGN, we set the batch size to be 512. We take the pretrained model  $E$ , which is IResNet-101 [55], trained on WebFace4M [300] with ArcFace loss [55] and freeze it without further tuning. For training CAFE, the number of images per identity  $N$  is randomly chosen between 2 and 16 during each step of training, and we take two sets per identity. The intermediate feature for the Style Input Component (SIM) is taken from the block 3 and 4 of the IResNet-101. The number of clusters in CN is varied in the ablation studies and fixed to be 4 for subsequent experiments. The number of layers  $L$  in CN is equal to 2.

We train the whole network end-to-end for 10 epochs with an AdamW optimizer [164]. The learning rate is set to  $1e-3$  and decayed by  $1/10$  at epochs 6 and 9. The weight decay is set to  $5e-4$ . For the loss term, we use  $\lambda_t = 1.0$  and  $\lambda_p = 1.0$  while the efficacy of  $\lambda_p = 1.0$  is ablated with  $\lambda_p = 0.0$  in the ablation studies. For  $\mathbf{f}_{GT}^{(p)}$ , we take the feature embeddings  $\mathbf{f}_i$  extracted from  $E$  for each labeled image in the training data, and average them per identity, with a flip augmentation.

### 3.6 Norm Embedding

For an embedding vector  $\mathbf{f}_i$ , the norm is a model dependent quantity, we L2 normalize the feature norm using batch statistics  $\boldsymbol{\mu}_f$  and  $\boldsymbol{\sigma}_f$  and convert it to a bounded integer between  $[-qk, qk]$ .

$$\widehat{\|\mathbf{f}_i\|} = \left\lfloor \left( q * \left( \left\lfloor \frac{\|\mathbf{f}_i\| - \boldsymbol{\mu}_f}{\boldsymbol{\sigma}_f} \right\rfloor_{-k}^k \right) \right) \right\rfloor. \quad (3.12)$$

Two hyper-parameters,  $q$  and  $k$  controll the concentration of the  $\widehat{\|\mathbf{f}_i\|}$  distribution and  $\lfloor \cdot \rfloor_{-k}^k$  refers to clipping the value between  $-k$  and  $k$ .  $\lfloor \cdot \rfloor$  refers to the floor operation to convert the quantity to an integer. Following the convention of Sinusoidal position embedding in [236], we let

$$\mathbf{n}_t(2t) = \sin(\widehat{\|\mathbf{f}_i\|}/10000^{\frac{2t}{c}}), \quad \mathbf{n}_t(2t+1) = \cos(\widehat{\|\mathbf{f}_i\|}/10000^{\frac{2t}{c}}), \quad (3.13)$$

where  $t$  is the channel index and  $c$  is the dimension of the norm embedding. The resulting  $\mathbf{n}_i \in \mathbb{R}^c$  is a 64-dim vector in our experiments.

### 3.7 Additional Performance Results

In this section, we provide additional performance results from IJB-A [131], IJB-B [253], IJB-C [169] and IJB-S [112] dataset with additional backbones.

<b>IJB-A [131]</b>	Dataset	Backbone $E$	TAR@FAR=0.001	TAR@FAR=0.01
Naive Average	VGGFace2(3.3M) [33]	ResNet50	$89.5 \pm 1.9$	$95.0 \pm 0.5$
QAN [159]	VGGFace2(3.3M) [33]	CNN256	$89.3 \pm 3.9$	$94.2 \pm 1.5$
NAN [266]	3M Web Crawl [266]	GoogLeNet	$88.1 \pm 1.1$	$94.1 \pm 0.8$
RSA [156]	VGGFace2(3.3M) [33]	ResNet50	$94.3 \pm 0.8$	$97.6 \pm 0.6$
Naive Average	WebFace4M [300]	IResNet101+ArcFace [55]	$98.5 \pm 0.6$	$99.1 \pm 0.2$
PFE [209]	WebFace4M [300]	IResNet101+ArcFace [55]	$98.5 \pm 0.6$	$99.1 \pm 0.2$
CFAN [78]	WebFace4M [300]	IResNet101+ArcFace [55]	$98.5 \pm 0.5$	$99.2 \pm 0.2$
RSA [156]	WebFace4M [300]	IResNet101+ArcFace [55]	$98.6 \pm 0.5$	$99.1 \pm 0.2$
CAFace	WebFace4M [300]	IResNet101+ArcFace [55]	<b><math>98.7 \pm 0.4</math></b>	<b><math>99.2 \pm 0.2</math></b>

Table 3.6 A performance comparison of recent methods on the IJB-A [131] dataset. The  $\pm$  sign refers to the standard deviation calculated from the official 10-fold cross validation splits from the dataset. For recent SoTA backbone models, the performance is saturated above 98.5.

<b>IJB-C [169]</b>	Dataset	Backbone $E$	TAR@FAR=1e-3	TAR@FAR=1e-4	TAR@FAR=1e-5
Naive Average	WebFace4M [300]	IResNet101+ArcFace [55]	97.30	95.78	92.60
PFE [209]	WebFace4M [300]	IResNet101+ArcFace [55]	97.53	96.33	94.16
CFAN [78]	WebFace4M [300]	IResNet101+ArcFace [55]	97.55	96.45	94.40
RSA [156]	WebFace4M [300]	IResNet101+ArcFace [55]	97.49	96.49	94.58
CAFace	WebFace4M [300]	IResNet101+ArcFace [55]	<b>97.99</b>	<b>97.15</b>	<b>95.78</b>
Naive Average	WebFace4M [300]	IResNet101+AdaFace [122]	97.63	96.42	94.47
CAFace	WebFace4M [300]	IResNet101+AdaFace [122]	<b>98.08</b>	<b>97.30</b>	<b>95.96</b>

Table 3.7 A performance comparison of recent methods on the IJB-C [131] dataset. CAFE achieves the best result in IJB-C dataset. We also compare two different backbones ArcFace [55] and AdaFace [122] (CVPR’22). The performance gain is observed in both backbones.

The size of the probes  $N$  in each dataset increases in the order of IJBA [131], IJBB [253], IJB-C [169] and IJB S [112]. As the probe size increases, the role of a feature fusion model also increases. As noted in Fig.1 c) of the main paper, previous methods either fail to model the intra-set relationship or scale to large  $N$ , which results in a suboptimal performance with

IJB-B [253]	Dataset	Backbone $E$	TAR@FAR=1e-3	TAR@FAR=1e-4	TAR@FAR=1e-5
Naive Average	WebFace4M [300]	IResNet101+ArcFace [55]	96.1	94.30	89.53
CAFace	WebFace4M [300]	IResNet101+ArcFace [55]	<b>96.91</b>	<b>95.53</b>	<b>92.29</b>
Naive Average	WebFace4M [300]	IResNet101+AdaFace [122]	96.66	94.84	90.86
CAFace	WebFace4M [300]	IResNet101+AdaFace [122]	<b>96.97</b>	<b>95.78</b>	<b>92.78</b>

Table 3.8 An additional performance on the IJB-B [131] dataset. We compare two different backbones ArcFace [55] and AdaFace [122] (CVPR’22).

Method	$E$	Surveillance-to-Single			Surveillance-to-Booking			Surveillance-to-Surveillance		
		Rank-1	Rank-5	1%	Rank-1	Rank-5	1%	Rank-1	Rank-5	1%
Naive Average	ArcFace	69.26	74.31	57.06	70.32	75.16	56.89	32.13	46.67	5.32
CAFace	ArcFace	<b>71.61</b>	<b>76.43</b>	<b>62.21</b>	<b>72.72</b>	<b>77.41</b>	<b>62.68</b>	<b>36.51</b>	<b>49.59</b>	<b>8.78</b>
Naive Average	AdaFace	70.42	75.29	58.27	70.93	76.11	58.02	35.05	48.22	4.96
CAFace	AdaFace	<b>72.91</b>	<b>77.14</b>	<b>62.96</b>	<b>73.39</b>	<b>78.04</b>	<b>63.61</b>	<b>39.25</b>	<b>50.47</b>	<b>7.65</b>

Table 3.9 An additional performance result on IJB-S [112] dataset with two different backbones, ArcFace [55] and AdaFace [122] (CVPR’22). AdaFace [122] combined with our proposed CAFE achieves a large margin improvement in IJB-S.

an increasing probe size. The plot of the relative performance increase over the naive average baseline shows that for CAFE, as the set size increases, the performance gain also increases. The relative performance gain for Fig.1 c) is calculated as  $\frac{Method - Naive}{Naive} * 100\%$  where the metrics for each dataset are TAR@FAR=0.001 for IJB-A, TAR@FAR=1e-4 for IJB-B and IJB-C, and the average of 9 metrics across all 3 protocols for IJB-S.

### 3.8 Resource and Efficiency Comparison

We report the FPS (frames per second) to give the estimation of how much resource the feature fusion framework takes with respect to the backbone  $E$ . For the table below, we use the backbone of IResNet-101 [55]. We measured the FPS with Nvidia RTX3090. It is equipped with a GPU memory of 24 GB. For measuring the time, we feed the random array as an input to the model and simulate the run for 1,000 times. In Tab. 3.10, we first show the FPS for the backbone  $E$ . The FPS increases with batch-size due to the efficiency of GPU architecture. We take 1,288 FPS as the FPS for the backbone and measure the relative FPS of the fusion models  $F$  with respect to the backbone, *i.e.*  $\frac{FPS(F)}{FPS(E)}$ .

In Tab. 3.11, we show  $\frac{FPS(F)}{FPS(E)}$  of various feature fusion models with the varied set size  $N$ . First, note that the feature fusion model’s inference speed is always faster than the

backbone model, *i.e.*  $\frac{FPS(F)}{FPS(E)} > 1$ . In practice, we would like the fusion time to be a fraction of the backbone inference time. Secondly, we show the maximum set size  $N$  each method can take. Note that methods without intra-set relationships, PFE [209] and CFAN [78], are computationally very fast and require little memory. Therefore, it can take many samples together (large  $N$ ) during inference. In contrast, the maximum set size  $N$  for RSA [156] is 384 because the intra-set attention with the feature map is a memory-intensive module. CAFace is fast and uses relatively little memory, allowing the maximum set number to be  $N = 12,000$ .

Note the ability to perform sequential inference is different from large  $N$ . For instance, with CAFace, we can split a set of size 64,000 with a batch size of 64 and run 1,000 sequential inferences, without sacrificing the performance. It is evident in the high performance of IJB-S dataset, where we adopt the batch size of 256.

	Batch Size	FPS
Backbone (Batchsize: 1)	1	91
Backbone (Batchsize: 256)	256	<b>1,288</b>

Table 3.10 FPS for the face recognition backbone model IResNet-101. Higher the FPS, the faster the inference speed per image.

$\frac{FPS(F)}{FPS(E)}$	Max $N$	$N = 16$	$N = 32$	$N = 64$	$N = 256$	$N = 512$
PFE	115,200	21.8x	44.1x	86.3x	360.1x	2133.6x
CFAN	115,200	82.6x	158.7x	268.8x	544.1x	664.2x
<b>CAFace</b>	<b>12,000</b>	<b>4.2x</b>	<b>8.2x</b>	<b>16.4x</b>	<b>64.4x</b>	<b>129.3x</b>
RSA	384	6.9x	13.1x	9.2x	3.1x	<b>OOM</b>

Table 3.11 A table of relative FPS of the fusion model w.r.t. the FPS of the backbone. We compare various fusion models with varied input size  $N$ . As  $N$  increases, it requires more GPU memory as well. Max  $N$  refers to the maximum number of images that can be in a set without causing the out of memory error (OOM). The higher the  $\frac{FPS(F)}{FPS(E)}$ , the faster the fusion method.

### 3.9 Training Progress and Learned Assignment

To see how the assignment behavior changes during training, we plot the entropy of the assignment map  $\mathbf{A} \in \mathbb{R}^{M \times N}$  over the training epochs. We note that each  $j$ -th cluster is a weighted average of individual  $N$  samples. Therefore, if all samples are contributing equally to the  $j$ -th cluster, then the entropy of  $A$  for each row would be high. When a few samples' contribution is larger than the others (*i.e.*,  $\mathbf{A}$  is sparse) then the entropy would be low. We use entropy as a proxy of how sparse is the influence of samples for each cluster.

The entropy is calculated as

$$\sum_{j=1}^M \sum_{i=1}^N -p_{j,i} \log(p_{j,i}),$$

where  $p_{j,i} = \mathbf{A}_{j,i} / \sum_{i=1}^N \mathbf{A}_{j,i}$ . In other words, it is the mean of the row-wise entropy of the normalized assignment map. Lower entropy value tells you that the cluster features are deviating from a simple average of all samples. In Fig. 3.8, we show the plot of the mean entropy over the training progression using the IJB-B dataset [253]. The value decreases steeply during the first few epochs, indicating that the clustering mechanism is quickly deviating from the simple averaging of the given samples.

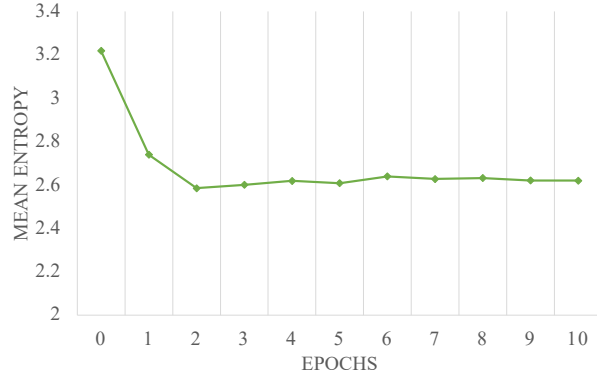


Figure 3.8 A plot of mean entropy during training. The samples used are random 200 probes taken from the IJB-B [253] dataset.

### 3.10 Weight Visualization

We show a few examples of the weight visualizations of different methods. The weights for CAFace are calculated as

$$w_i = \frac{\sum_j^M \mathbf{A}_{j,i} \sum_{c=1}^C (\mathbf{P}_{j,c}/C)}{z},$$

the sum of the contributions each sample makes to each cluster, weighed by the importance of the cluster.  $C$  is the dimension of  $\mathbf{f}$ , which is 512 in our backbone.  $M$  is the number of clusters.  $z$  is the normalization constant to make the  $\sum_{i=1}^N w_i = 1$ . For the Averaging, the weights are the normalized feature norms. For PFE and CFAN, the weights are the output of the respective modules. Note that RSA does not have a weight estimation, as it directly estimates the fused output as opposed to estimating the weights. The circles in the plot represent individual probe images in IJB-S and the color represents the magnitude of the weights. The horizontal axis represents the similarity of individual probe images to the gallery shown on the right. The vertical axis exists only to scatter the points. Note that for both PFE and CFAN, the weight estimation is based on a single image.

### 3.11 Comparison of Assignment Maps in Various Scenarios

To analyze the behavior of the assignment map  $\mathbf{A} \in \mathbb{R}^{4 \times N}$  of CAFace in varied scenarios, we show in Fig. 3.10, IJB-S [112] probe examples that come from 3 typical settings; mixed, poor and good quality image scenarios. The mixed-quality probe is comprised of both low and high quality images as illustrated in scenario 1. On the other hand, probes could contain all poor or all good quality images as illustrated by scenarios 2 and 3. Note that each column of  $\mathbf{A}$  sums to 1, and each row of  $\mathbf{A}$  are the relative weights responsible for creating each clustered vector in  $\mathbf{F}' \in \mathbb{R}^{4 \times 512}$ .

Note that cluster 4 works as a place where bad quality images are strongly assigned to. Since the mean  $\mathbf{P}_4$  is close to zero, all images assigned to cluster 4 have very little contribution to the final fused output  $\mathbf{f}$ . For scenario 2 where all of the images are of bad quality, a few *relatively* better images are still assigned to cluster 1, 2 and 3, making it possible to perform feature fusion with bad quality images. This is possible because CAFace incorporates intra-set

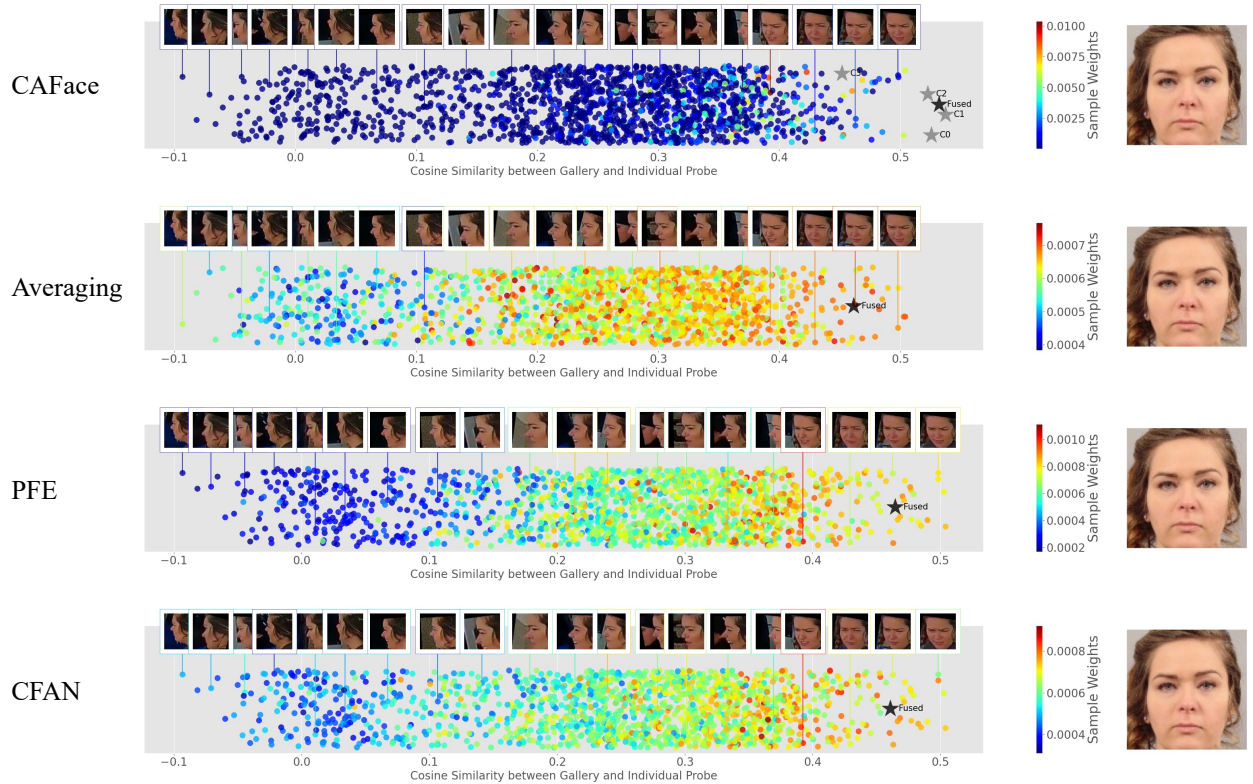


Figure 3.9 Visualization of importance weights.

relationships that allow the information to communicate among the inputs to determine which features are more usable than the others. For scenario 3, we can observe that most of the images are quite similar to one another, providing duplicating information. Therefore, the assignments are learned to discard many of the duplicating images, as shown by the high (red) values in the last row of scenario 3.

### 3.12 Effect of Sequence Length

In Fig. 3.11, to illustrate the importance of using all video sequences, we show how the IJB-S performance of CAFE changes as we divide the probe videos into 10 partitions and use first  $1:k$  partitions. The increasing trend reveals that longer video sequences can provide more information for fusion.



is designed to linearly combine  $N$  inputs to  $M$  global cluster centers, whose assignment is invariant to the batch-order. The aggregation stage efficiently produces a fused output from  $M$  clustered features, while utilizing the intra-set relationship. We show our proposed CAFace outperforms baselines on unconstrained face datasets such as IJB-B and IJB-S.

**Limitations** Cluster and Aggregate is a feature fusion framework that learns the weights of individual inputs, given a fixed feature extractor  $E$ . Weight estimation, in other words, is an *interpolation* among the given set of features which is a double-edged sword, as it gives the interpretability, but is not capable of *extrapolation*. Therefore, when the given feature extractor  $E$  is sub-optimal, it could be favorable to relax the constraint and let the model extrapolate for better performance.

**Potential Negative Societal Impacts** We believe that the machine learning community as a whole should strive to minimize the negative societal impacts. Large scale face recognition training datasets inevitably comprise web-crawled images which are without formal consent, or IRB review. We refrained from using any dataset withdrawn by its creators such as VGG-2 [33] or MS1MV2 [82] to avoid any known copyright issues. We hope that the FR community can collectively move toward collecting datasets with informed consent, fostering R&D without societal concern.

## CHAPTER 4

### DCFACE: SYNTHETIC FACE GENERATION WITH DUAL CONDITION DIFFUSION MODEL

Generating synthetic datasets for training face recognition models is challenging because dataset generation entails more than creating high fidelity images. It involves generating multiple images of same subjects under different factors (*e.g.*, variations in pose, illumination, expression, aging and occlusion) which follows the real image conditional distribution. Previous works have studied the generation of synthetic datasets using GAN or 3D models. In this work, we approach the problem from the aspect of combining subject appearance (ID) and external factor (style) conditions. These two conditions provide a direct way to control the inter-class and intra-class variations. To this end, we propose a Dual Condition Face Generator (DCFace) based on a diffusion model. Our novel Patch-wise style extractor and Time-step dependent ID loss enables DCFace to consistently produce face images of the same subject under different styles with precise control. Face recognition models trained on synthetic images from the proposed DCFace provide higher verification accuracies compared to previous works by 6.11% on average in 4 out of 5 test datasets, LFW, CFP-FP, CPLFW, AgeDB and CALFW. Code Link

#### 4.1 Introduction

What does it take to create a good training dataset for visual recognition? An ideal training dataset for recognition tasks would have 1) large inter-class variation, 2) large intra-class variation and 3) small label noise. In the context of face recognition (FR), it means, the dataset has a large number of unique subjects, large intra-subject variations, and reliable subject labels. For instance, large-scale face datasets such as WebFace4M [300] contain over 1M subjects and large number of images/subject. Both the number of subjects and the number of images per subject are important for training FR models [55, 122]. Also, datasets amassed by crawling the web are not free from label noise [33, 300].

In various domains, synthetic datasets are traditionally used to help generalize deep



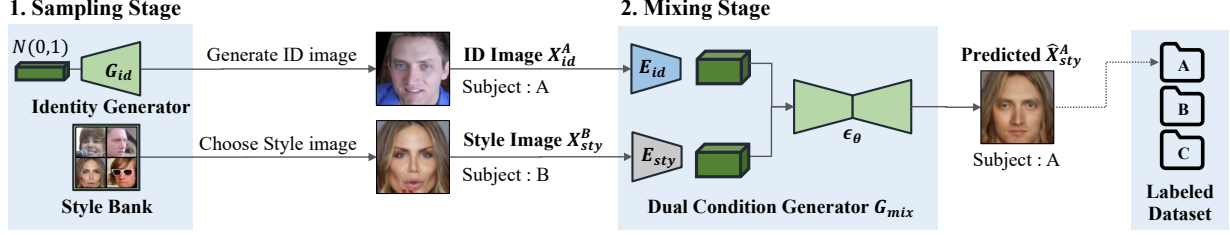


Figure 4.2 Two stage dataset generation paradigm. In the sampling stage, 1)  $G_{id}$  generates a high-quality face image  $\mathbf{X}_{id}$  that defines how a person looks and 2) the style bank selects a style image  $\mathbf{X}_{sty}$  that defines the overall style of the final image. The mixing stage generates image with identity from  $\mathbf{X}_{id}$  and style from  $\mathbf{X}_{sty}$ . Repeating this process multiple times, one can generate a labeled synthetic face dataset.

generated by DiscoFaceGAN is less than 500, a finding that will be discussed in Sec. 4.3.1. The recent state of the art (SoTA), DigiFace [17], can generate 1M large-scale synthetic face images with many unique subjects based on 3D parametric model rendering. However, it falls short in matching the quality and style of real face images.

We propose a new data generation scheme that addresses all three criteria, *i.e.* the large number of novel subjects (*uniqueness*), real dataset style matching (*diversity*) and label consistency (*consistency*). In Fig. 4.1, we illustrate the high-level idea by showcasing some of our generated face samples. The key motivation of our paper is that the synthetic dataset generator needs to control the number of unique subjects, match the training dataset’s style distribution and be consistent in the subject label.

In light of this, we formulate the face image generation as a dual condition inverse problem, retrieving the unknown image  $\mathbf{Y}$  from the observable Identity condition  $\mathbf{X}_{id}$  and Style condition  $\mathbf{X}_{sty}$ . Specifically,  $\mathbf{X}_{id}$  specifies how a person looks and  $\mathbf{X}_{sty}$  specifies how  $\mathbf{X}_{id}$  should be portrayed in an image.  $\mathbf{X}_{sty}$  contains identity-independent information such as pose, expression, and image quality.

Our choice of dual conditions (identity and style) is important in how we generate a synthetic dataset as ID and style conditions are controllable factors that govern the dataset’s characteristics. To achieve this, we propose a two-stage generation paradigm. First, we

generate a high-quality face image  $\mathbf{X}_{id}$  using a face image generator and sample a style image  $\mathbf{X}_{sty}$  from a style bank. Secondly, we mix these two conditions using a dual condition generator which predicts an image that has the ID of  $\mathbf{X}_{id}$  and a style of  $\mathbf{X}_{sty}$ . An illustration is given in Fig. 4.2.

Training the mixing generator in stage 2 is not trivial as it would require a triplet of  $(\mathbf{X}_{id}^A, \mathbf{X}_{sty}^B, \mathbf{X}_{sty}^A)$  where  $\mathbf{X}_{sty}^A$  is a hypothetical combination of the ID of subject  $A$  and the style of subject  $B$ . To solve this problem, we propose a new dual condition generator that can learn from  $(\mathbf{X}_{id}^A, \mathbf{X}_{sty}^A)$ , a tuple of same subject images that can always be obtained in a labeled dataset. The novelty lies in our style condition extractor and ID loss which prevents the training from falling into a degenerate solution. We modify the diffusion model [91, 213] to take in dual conditions and apply an auxiliary time-dependent ID loss that can control the balance between sample diversity and label consistency.

We show that our Dual Condition Face Dataset Generator (DCFace) is capable of surpassing the previous methods in terms of FR performance, establishing a new benchmark in face recognition with synthetic face datasets. We also show the roles dataset subject uniqueness, diversity and consistency play in face recognition performance.

The followings are the contributions of the paper.

- We propose a two-stage face dataset generator that controls subject uniqueness, diversity and consistency.
- For this, we propose a dual condition generator that mixes the two independent conditions  $\mathbf{X}_{id}$  and  $\mathbf{X}_{sty}$ .
- We propose uniqueness, consistency and diversity metrics that quantify the respective properties of a given dataset, useful measures that allow one to compare datasets apart from the recognition performance.
- We achieve SoTA in FR with 0.5M image synthetic training dataset by surpassing the previous methods by 6.11% on average in 5 popular test datasets.

## 4.2 Related Works

**Face Recognition** Face Recognition (FR) is the task of matching query imagery to an enrolled identity database. SoTA FR models are trained on large-scale web-crawled datasets [55, 82, 300] with margin-based softmax losses [55, 102, 122, 154, 240]. The FR performance is measured on various benchmark datasets such as LFW [100], CFP-FP [202], CPLFW [296], AgeDB [174] and CALFW [297]. These datasets are designed to measure factors such as pose changes and age variations. Performance on these datasets for models trained on large-scale datasets such as WebFace260M is well above 97% [122] in verification accuracy.

**Synthetic Face Generation** Recent advances in generative models allow high fidelity synthetic face image generations [30, 47, 91, 115–117, 215]. GANs have been widely used to manipulate, animate or enhance face images [47, 59, 96, 143, 187, 221, 231, 262]. They typically learn disentangled representations in GAN latent space that control desired face properties. On the contrary, some works leverage the 3D face prior from 3D datasets (*e.g.*, 3DMM [24]) for controllable synthesis [52, 73, 75, 119, 170, 178, 185, 207]. These methods have advantages in the fine-grained control over face generation and 3D consistency yet lack in style or domain variation.

Recent advances in the latent variable models such as diffusion or score-based models have shown great success in high-quality image generation with a more stable and simple objective of MSE loss [91, 179, 213, 215–218]. Diffusion models have advanced the conditional image generation in tasks such as text-conditional image generation, inpainting, *etc* [25, 190, 196, 246]. We adopt the diffusion model as a backbone and explore how the two image characteristics, namely ID and style images, can control complementary information, the subject appearance and the style of an image.

**Face Recognition with Synthetic Dataset** Synthetic training datasets offer an advantage over real datasets with regards to ethical issues and class imbalance problems as large-scale face datasets have been criticized for lacking informed consent and reflecting racial biases [17, 55, 274, 300]. Despite the benefit, use of synthetic datasets as the sole training data is

not widely adopted due to the resulting low recognition performance. In various domains such as face recognition [17, 150, 188], fingerprint recognition [64, 260], and anti-spoofing [158, 219], synthetic datasets have been shown to improve recognition when combined with real images.

In the face domain, SynFace [188] studied the efficacy of using DiscoFaceGAN [59] for synthetic face generation. Recently, DigiFace-1M [17] studied the efficacy of 3D model based face rendering in combination with image augmentations to create a synthetic dataset. We propose a face dataset generation method that can generate both a large number of subjects and diverse styles that are close to the real dataset.

### 4.3 Proposed Approach

We propose Dual Condition Face Dataset Generator (DCFace), a two-stage dataset generator (see Fig. 4.2). Stage 1 is the Condition Sampling Stage, generating a high-quality ID image ( $\mathbf{X}_{id}$ ) of a novel subject and selects one arbitrary style image ( $\mathbf{X}_{sty}$ ) from the bank of real training data. Stage 2 is the Mixing Stage which combines the two images using the Dual Condition Generator.

For trainable models in each stage, Stage 1 requires training an ID image generator  $G_{id}$ . For the style bank, we can conveniently use any real face dataset that we wish generated samples to follow. Stage 2 requires training a dual condition mixer  $G_{mix}$ . Both  $G_{id}$  and  $G_{mix}$  are based on diffusion models [91]. We describe each component and the associated training procedure in the following subsections.

#### 4.3.1 Preliminary

Diffusion models [91, 213] are a class of denoising generative models that are trained to predict an image from random noise through a gradual denoising process. One notable difference from the class of GAN-based generators [79] is in the objective function and the sampling procedure. The forward process as expressed in Eq. 4.1 corrupts the input  $\mathbf{X}$  using variance controlled Gaussian noise over  $t$  time-steps,

$$q(\mathbf{X}_t | \mathbf{X}_{t-1}) = \mathcal{N}\left(\mathbf{X}_t; \sqrt{1 - \beta_t} \mathbf{X}_{t-1}, \beta_t \mathbf{I}\right), \quad (4.1)$$

and the denoising is done by training a model  $\epsilon_\theta(\mathbf{X}_t, t)$  to predict the initial noise  $\epsilon$  with an

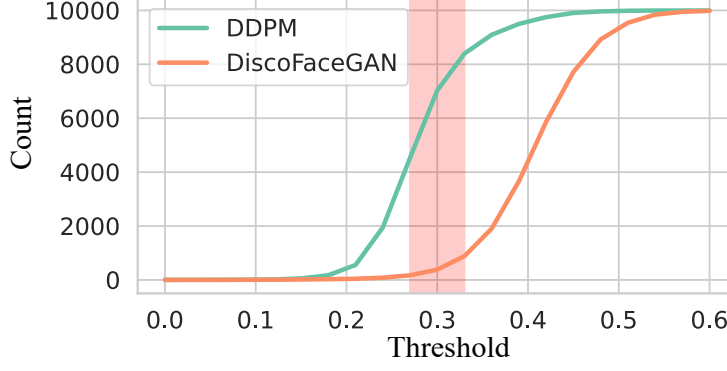


Figure 4.3 Comparison of the number of unique subjects generated by DiscoFaceGAN [59] and unconditional DDPM [91]. Uniqueness is the number of unique subjects measured by a face recognition model. By varying the threshold which determines a match between two subjects, we plot the number of unique subjects as defined in Eq. 4.11. Unconditional DDPM and DiscoFaceGAN are trained on FFHQ [116] and each generates 10,000 samples. The ability to generate novel subjects is larger for DDPM.

$L_2$  objective,

$$\mathcal{L} = \mathbb{E}_{t, \mathbf{X}_0, \epsilon} \left[ \left\| \underbrace{\epsilon_{\theta}(\sqrt{\alpha_t} \mathbf{X}_0 + \sqrt{1 - \alpha_t} \epsilon, t)}_{\mathbf{X}_t} - \epsilon \right\|_2^2 \right]. \quad (4.2)$$

$\beta_t$  and  $\alpha_t$  are pre-set variance scheduling scalars. The denoising diffusion model (DDPM) has shown success in producing diverse samples in text-conditioned image generation [190]. We find that in unconditional face generation, DDPM is also capable of generating many unique subjects. For instance, Fig. 4.3 compares DiscoFaceGAN [59] with DDPM [91] in their capacity to generate different subjects for every sample. It shows that DDPM [91] is a good model choice for  $G_{id}$  and  $G_{mix}$  as it can generate many unique subjects. For  $G_{id}$ , we adopt the unconditional DDPM trained on FFHQ [116], having observed that it is capable of generating a large number of unique subject images.

#### 4.3.2 Dual Condition Generator $G_{mix}$

The two-stage data generation requires Dual Condition Generator  $G_{mix}$  which is a conditional DDPM. Specifically, two conditions  $\mathbf{X}_{id}$  and  $\mathbf{X}_{sty}$  are injected into the denoiser  $\epsilon_{\theta}(\mathbf{X}_t, t, E_{id}(\mathbf{X}_{id}), E_{sty}(\mathbf{X}_{sty}))$  using trainable feature extractors  $E_{id}$  and  $E_{sty}$  and cross-

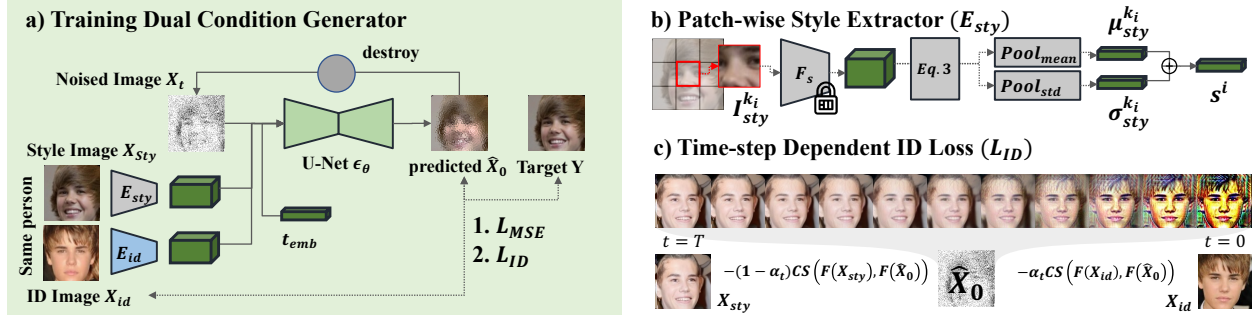


Figure 4.4 a) A diagram of  $G_{mix}$  during training. At each step, we draw two labeled images from the labeled training dataset and use them as  $\mathbf{X}_{id}$  and  $\mathbf{X}_{sty}$ . We ensure  $\mathbf{X}_{id}$  to be the good-quality frontal view image.  $t_{emb}$  is the time-step embedding in DDPM [91].  $\mathbf{X}_{sty}$  also serves as a target image and we apply Gaussian noise  $\epsilon$  to  $\mathbf{X}_{sty}$  to create  $\mathbf{X}_t$  as DDPM specifies. Then  $\epsilon_{\theta}(\mathbf{X}_t, t, \mathbf{X}_{id}, \mathbf{X}_{sty})$  is trained to predict  $\epsilon$  using  $L_{MSE}$ , conceptually equivalent to the reconstruction loss to recover  $\mathbf{X}_{sty}$ . We also apply  $L_{ID}$  as in Eq. 4.10 for the dependence on  $\mathbf{X}_{id}$ . b) Patch-wise Style Extractor generates style vectors from small patches of images. Style vectors are architecturally constrained from containing full ID information. c) Time-step dependent ID Loss is a linear interpolation between the  $\mathbf{X}_{id}$  and  $\mathbf{X}_{sty}$  in the recognition feature space. It forces  $\epsilon_{\theta}$  to rely on  $\mathbf{X}_{id}$  to extract the subject’s appearance and gradually shift the style to  $\mathbf{X}_{sty}$ .

attentions.  $G_{mix}$  is responsible for the operation  $\mathbf{X}_{id}^A + \mathbf{X}_{sty}^B \rightarrow \mathbf{X}_{sty}^A$ , a mixing of an image of a novel subject  $A$  and an arbitrary style image of different subject  $B$ .

Naive training would require the reference image  $\mathbf{X}_{sty}^A$ , an image of subject  $A$  in the style of  $\mathbf{X}_{sty}^B$ . This reference is absent in the labeled training dataset. As such, we modify the operation to  $\mathbf{X}_{id}^A + \mathbf{X}_{sty}^A \rightarrow \mathbf{X}_{sty}^A$ , using two different images from the same subject as illustrated in Fig. 4.4(a). But this formulation is prone to a trivial solution of ignoring  $\mathbf{X}_{id}^A$ , making the ID condition unused during test time. To mitigate this issue, we propose the following two elements.

**Patch-wise Style Extractor  $E_{sty}$**  The motivation of Style Extractor is to map an image  $\mathbf{X}_{sty}$  to a feature that contains little ID information, forcing  $G_{mix}$  to rely on  $\mathbf{X}_{id}$  for ID information. In prior works such as StyleGAN, 1<sup>st</sup> and 2<sup>nd</sup> order statistics of a feature are

shown to resemble the image style [116, 123, 136]. Yet, resulting statistics are reduced in spatial dimensions and consequently without spatially local informations such as pose.

We propose a module that can extract style information without losing spatial information. Specifically, consider a pretrained and fixed face recognition model  $F_s$  and its intermediate feature  $F_s(\mathbf{X}_{sty}) = \mathbf{I}_{sty} \in \mathbb{R}^{C \times H \times W}$ . We divide the feature into a  $k \times k$  grid. For each element in the grid  $\mathbf{I}_{sty}^{k_i} \in \mathbb{R}^{C \times \frac{H}{k} \times \frac{W}{k}}$ , we perform non-linear mapping on the mean and variance of  $\mathbf{I}_{sty}^{k_i}$ . Specifically,

$$\hat{\mathbf{I}}^{k_i} = \text{BN}(\text{Conv}(\text{ReLU}(\text{Dropout}(\mathbf{I}_{sty}^{k_i})))), \quad (4.3)$$

$$\boldsymbol{\mu}_{sty}^{k_i} = \text{SpatialMean}(\hat{\mathbf{I}}^{k_i}), \quad \boldsymbol{\sigma}_{sty}^{k_i} = \text{SpatialStd}(\hat{\mathbf{I}}^{k_i}), \quad (4.4)$$

$$\mathbf{s}^{k_i} = \text{LN}((\mathbf{W}_1 \odot \boldsymbol{\mu}_{sty}^{k_i} + \mathbf{W}_2 \odot \boldsymbol{\sigma}_{sty}^{k_i}) + \mathbf{P}_{emb}), \quad (4.5)$$

$$E_{sty}(\mathbf{X}_{sty}) := \mathbf{s} = [\mathbf{s}^1, \mathbf{s}^2, \mathbf{s}^{k_i} \dots, \mathbf{s}^{k \times k}, \mathbf{s}'], \quad (4.6)$$

where  $\mathbf{s}'$  corresponds to  $\mathbf{I}_{sty}^{k_i}$  being a global feature, where  $k = 1$ . The final output  $\mathbf{s}$  is a concatenation of all style vectors for each patch. Each  $\mathbf{s}^{k_i}$  is a mean and variance of local information which is constrained from containing full pixel-level details with the ID information. And  $\mathbf{P}_{emb}$  is a learned position embedding to let the model differentiate different patch locations. BN and LN are BatchNorm [107] and LayerNorm [16].  $F_s$  is a shallow CNN taken from the early layers of a pretrained FR model. It is fixed and not updated to prevent it from optimizing  $\mathbf{I}_{sty}$ , serving only to create style information. By varying the grid size  $k \times k$ , we can represent style at different spatial locations. An illustration of  $E_{sty}$  can be found in Fig. 4.4(b).

**Time-step Dependent ID Loss** To train Dual Condition Generator  $G_{mix}$ , the original DDPM objective of  $L_2$  loss, Eq. 4.2 is not sufficient to guarantee the consistency in subject identity between the ID condition  $\mathbf{X}_{id}$  and the prediction,  $\hat{\mathbf{X}}_0$ . To ensure the ID consistency, one could devise a loss function to maximize the similarity between  $\mathbf{X}_{id}$  and the predicted denoised image  $\hat{\mathbf{X}}_0$ , in the ID feature space using a pretrained FR model,  $F$ . Specifically,

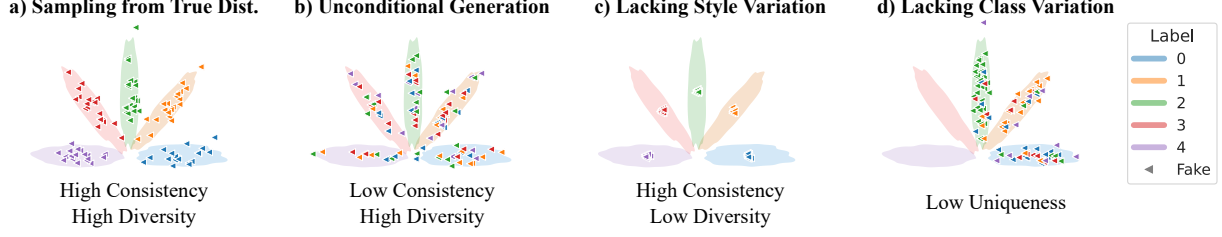


Figure 4.5 Illustration of conditional distributions in 2D space. Colored regions represent the true data distribution with individual colors representing different labels. Colored triangles represent generated samples with corresponding labels. For each scenario except (a), the generated distribution does not follow the true distribution. Consistency, diversity and uniqueness analysis can quantify the shortcomings.

following the Eq.15 of DDPM [91], one-step prediction of the original image is

$$\hat{\mathbf{X}}_0 = (\mathbf{X}_t - \sqrt{1 - \bar{\alpha}_t} \epsilon_\theta(\mathbf{X}_t, t, \mathbf{X}_{id}, \mathbf{X}_{sty})) / \sqrt{\bar{\alpha}_t}. \quad (4.7)$$

A simple ID loss to increase cosine similarity (CS) is

$$L_{\text{naive1}} = -\text{CS} \left( F(\mathbf{X}_{id}), F(\hat{\mathbf{X}}_0) \right). \quad (4.8)$$

However, this loss is in conflict with MSE loss and is empirically observed to reduce the predicted image quality. This is because the FR model,  $F$  is not invariant to image style; some style of  $\mathbf{X}_{id}$  has to match in order to completely reduce  $L_{\text{naive1}}$ . In contrast, one could also use

$$L_{\text{naive2}} = -\text{CS} \left( F(\mathbf{X}_{sty}), F(\hat{\mathbf{X}}_0) \right), \quad (4.9)$$

as during training the label of  $\mathbf{X}_{sty}$  and  $\mathbf{X}_{id}$  are the same. However,  $L_{\text{naive2}}$  causes the model to depend on  $\mathbf{X}_{sty}$  for ID information. Thus, during evaluation, when  $\mathbf{X}_{sty}$  and  $\mathbf{X}_{id}$  are different subjects, the label consistency in the generated dataset is compromised. We show this in Tab. 4.2.

Instead, we propose to interpolate between  $F(\mathbf{X}_{id})$  and  $F(\mathbf{X}_{sty})$  across diffusion time-steps. Specifically,

$$\begin{aligned} L_{\text{ID}} = & -\gamma_t \text{CS} \left( F(\mathbf{X}_{id}), F(\hat{\mathbf{X}}_0) \right) \\ & - (1 - \gamma_t) \text{CS} \left( F(\mathbf{X}_{sty}), F(\hat{\mathbf{X}}_0) \right), \end{aligned} \quad (4.10)$$

where  $\gamma_t = \frac{t}{T}$  is a time-dependent weight that linearly changes from 0 to 1. When  $t=T$ ,  $\epsilon_\theta$  is predicting  $\mathbf{X}_{t-1}$  from random noise, and we let the model fully exploit the ID information of  $\mathbf{X}_{id}$ . Gradually as  $t$  increases, we let the model’s prediction walk into the direction of  $\mathbf{X}_{sty}$ . Note that during training, the actual label of  $\mathbf{X}_{sty}$  and  $\mathbf{X}_{id}$  are the same. So the interpolation in the loss forces the prediction to be the same in identity but gradually shifting in style toward  $\mathbf{X}_{sty}$ . This loss allows  $\epsilon_\theta(\mathbf{X}_t, t, \mathbf{X}_{id}, \mathbf{X}_{sty})$  to play different roles depending on  $t$ . For  $t \approx T$ ,  $\epsilon_\theta$  will exploit  $\mathbf{X}_{id}$  to infer front-view ID rich image. And as  $t \rightarrow 0$ , it will change the image’s style to match the style of  $\mathbf{X}_{sty}$ . The final loss is  $L_{MSE} + \lambda L_{ID}$  with  $\lambda$  as a scaling parameter.

**$E_{id}$  and Conditioning Mechanism** Following the success text-conditional image generation and inpainting using DDPM [186, 190, 246], we adopt a similar architecture for inserting conditions into the model. We concatenate  $E_{id}(\mathbf{X}_{id})$  and  $E_{sty}(\mathbf{X}_{sty})$  and put in  $\epsilon_\theta$  using cross-attention and adaptive group normalization layers (AdaGN) [186].  $E_{id}$  is a CNN, with the same architecture as a small FR model (*e.g.* ResNet50). And  $E_{id}$  is trained end-to-end with  $\epsilon_\theta$  to extract useful ID feature for  $\epsilon_\theta$ .

### 4.3.3 Condition Sampling Strategy

**ID Image Sampling** For sampling ID images, we generate 200,000 facial images from  $G_{ID}$ , from which we remove faces that are wearing sunglasses or too similar to the subjects in CASIA-WebFace with the Cosine Similarity threshold of 0.3 using  $F_{eval}$ . We are left with 105,446 images. Then we narrow them down to 62,570 images that are unique according to uniqueness, Eq. 4.11 using  $F_{eval}$  and  $r = 0.3$ . Then we explore two different options, 1) random sampling and 2) gender/ethnicity balanced sampling as  $G_{id}$  has a skewed distribution towards White subjects as shown in Tab. 4.1. We use [9] to classify the ethnicity and use [109, 257] to detect sunglasses. We denote the sampling option 1 as *random* and 2 as *balance*.

**Style Image Sampling** For style sampling, for each  $\mathbf{X}_{id}$ , we randomly sample  $\mathbf{X}_{sty}$  from the style bank. We denote this option as *random*. We also explore the option of sampling

	White	Asian	Others	Black	Indian
CASIA-WebFace	0.634	0.144	0.074	0.074	0.072
DDPM $G_{id}$	0.660	0.209	0.034	0.046	0.048
Balanced Ethnicity	0.200	0.200	0.200	0.200	0.200

Table 4.1 Ethnicity Distribution of CASIA-WebFace. Ethnicity prediction is made using [9]. DDPM  $G_{id}$  is trained on FFHQ [116].

$\mathbf{X}_{sty}$  from the pool of images whose gender/ethnicity matches that of  $\mathbf{X}_{id}$ . We denote this option as *match*.

#### 4.4 Dataset Evaluation

In evaluating the synthesized dataset, one often adopts 1) FID [90] for evaluating the distribution similarity to the real images and 2) subsequent recognition performance. In this section, we propose three class-dependent metrics that aid us in understanding the property of generated labeled datasets. We let  $F_{eval}$  be a recognition model used for evaluating synthesized face datasets. Note that this is different from  $F$  in ID loss.  $F$  is a model for training loss and  $F_{eval}$  is for evaluating metrics. The more generalizable  $F_{eval}$  is, the more accurate the metrics become in capturing the identity and diversity of the synthesized dataset.

Let  $y_c$  be a class label, and  $f_i = F_{eval}(\mathbf{X}_i)$ . Let  $d(f_i, f_j)$  be the distance between two images in  $F_{eval}$  feature space.

**Uniqueness** Consider the following non-overlapping  $r$ -ball in  $F_{eval}$  space,

$$U = \{f_i : d(f_i, f_j) > r, j < i, i, j \in \{1, \dots, N\}\}, \quad (4.11)$$

where  $d(f_i, f_j)$  is the cosine distance. Then  $|U|$  is the count of unique subjects determined by the threshold  $r$  in an unlabeled dataset. Note that the set  $U$  is equivalent to sequentially adding a  $r$ -ball into  $F_{eval}$ -space until you cannot add more without collision.  $|U|$  is subject to both  $r$  and  $F_{eval}$ . In FR,  $r$  is a threshold in the FR model that is set to determine match or non-match.

For a labeled synthetic dataset, one generates multiple feature sets  $\{f_i^c\}$  for the same label. To count the number of unique subjects, we calculate the number of unique centers,  $f^c = \frac{1}{N_c} \sum_i^{N_c} f_i^c$  for  $c \in \{1, \dots, C\}$ , where  $C$  is the number of subjects and  $N_c$  is the number of images per subject. Then we define the number of unique subjects in a labeled dataset

with  $|U_c|$  where  $U_c$  is

$$U_c = \{f_c : d(f^{c_n}, f^{c_m}) > r, m < n, n, m \in \{1, \dots, C\}\}, \quad (4.12)$$

For the metric, we use  $U_{class} = |U_c|/C$ , the ratio between the number of unique subjects and the number of labels.

**Intra-class Consistency** It measures how consistent the generated samples are in adhering to the label condition, as

$$C_{intra} = \frac{1}{C} \sum_{c=1}^C \frac{1}{N_c} \sum_{i=1}^{N_c} d(f_i^c, f^c) < r, \quad (4.13)$$

which is the ratio of individual features  $f_i^c$  being close to the class center  $f^c$ . For a given threshold  $r$ , higher values of  $C_{intra}$  mean the samples are more likely to be the same subject under the same label.

**Intra-class Diversity** It measures how diverse the generated samples are under the same label condition. Note that the diversity is in the style of an image, not in the subject's identity. We define the style space as a vector space defined by Inception Network [198] features pretrained on ImageNet [53] following the convention of [134], denoting the real and generated image inception vectors as  $\{s_i^c\}, \{\hat{s}_j^c\}$ .

For intra-class diversity, we measure how many real images fall into the style space manifold defined by the generated images under the same label condition. We compute this by extending the Improved Recall Metric [134], from comparing the unconditional distributions of real and fake images to comparing the label-conditional distributions. Specifically, for a set of real and generated feature vectors  $\{s_i^c\}, \{\hat{s}_j^c\}$  under the same label condition  $y_c$ , we define  $k$ -nearest feature distance  $r_k$  as  $r_k = d(\hat{s}_j^c - \text{NN}_k(\hat{s}_j^c, \{s_i^c\}))$ , where  $\text{NN}_k$  returns the  $k$ -nearest feature vector in  $\{s_i^c\}$  and

$$\mathbf{I}(s_i^c, \{\hat{s}_j^c\}) = \begin{cases} 1, \exists \hat{s}_j^c \in \{\hat{s}_j^c\} \text{ s.t. } d(s_i^c - \hat{s}_j^c) \leq r_k \\ 0, \text{ otherwise.} \end{cases} \quad (4.14)$$

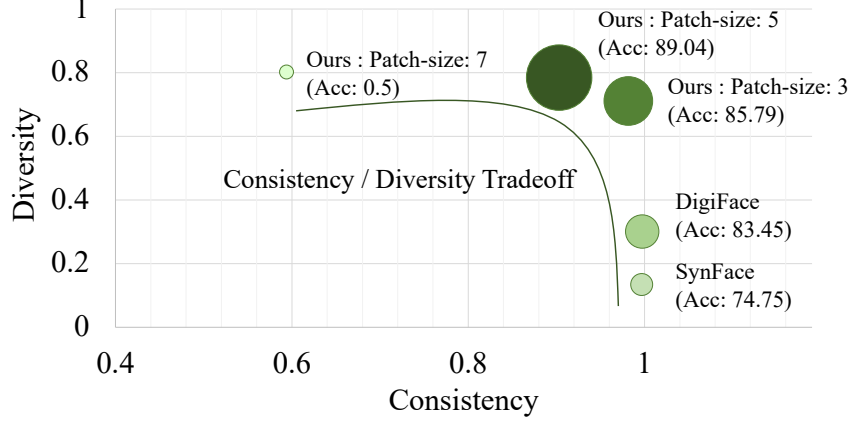


Figure 4.6 A plot of FR performance on 5 synthetic datasets with respect to Consistency and Diversity metrics. Color intensity and circle size denotes the FR accuracy.

$d(\cdot)$  is an Euclidean distance. Then diversity is defined by

$$D_{intra} = \frac{1}{C} \frac{1}{N} \sum_{c=1}^C \sum_{i=1}^{N_c} \mathbf{I}(s_i^c, \{\hat{s}_j^c\}), \quad (4.15)$$

which is the fraction of real image styles manifold covered by the generated image style manifold as defined by  $k$ -nearest neighbor ball. If the style variation is small, then  $r_k$  becomes small, reducing the chance of  $d(s_i^c - \hat{s}_j^c) \leq r_k$ . We compute the recall per class to capture style variation conditional on the subject label.

In Fig. 4.5, we illustrate different scenarios of conditional generation and how these metrics can capture the shortcomings in each scenario. In Sec. 4.5 and Fig. 4.6, we measure the metrics on our generated datasets and compare with previous synthetic datasets [17, 188]. We find that FR performance is at best when consistency and diversity are balanced. Also, we find SynFace and DigiFace have high  $C_{intra}$  and low  $D_{intra}$  compared to our method in Fig. 4.5.

## 4.5 Experiments

For  $G_{id}$  which generates ID images, we adopt the publicly released unconditional DDPM [91] trained on FFHQ [116]. For  $G_{mix}$ , we train it on CASIA-WebFace [99] after initializing weights from  $G_{id}$ . Although using all of CASIA-WebFace is a valid setting, we split it into a 95-5 split between train and validation sets. The validation set is used as a real dataset in

Grid Size	Loss	Loss Model	$U_{class}$	$C_{intra}$	$D_{intra}$	FR Perf.
SynFace	-	-	0.080	0.9966	0.131	74.75
DigiFace	-	-	0.178	0.9973	0.297	83.45
$1 \times 1$	$L_{ID}$	$F$	<b>0.978</b>	<b>0.9987</b>	0.4418	79.28
$3 \times 3$			0.956	0.9809	0.7030	85.79
<b><math>5 \times 5</math></b>			0.924	0.9035	0.7734	<b>89.04</b>
$7 \times 7$			0.690	0.5937	<b>0.7950</b>	50.00
$5 \times 5$	$L_{naive1}$	$F$	<b>0.988</b>	<b>0.9996</b>	0.6546	84.75
	$L_{naive2}$		0.866	0.8046	0.7835	50.00
	<b><math>L_{ID}</math></b>		0.924	0.9035	<b>0.7734</b>	<b>89.04</b>
$5 \times 5$	$L_{ID}$	<b><math>F</math></b>	0.924	0.9035	0.7734	89.04
		$F_{bigger}$	<b>0.954</b>	<b>0.9197</b>	0.7715	<b>89.89</b>

Table 4.2 Model Ablation. For FR performance, we generate a synthetic dataset of  $10K$  subjects with 50 images per subject using (*random*, *random*) ID and style sampling strategy. Blue color indicates the adopted setting for subsequent experiments.

measuring the uniqueness, consistency and diversity metrics.  $G_{mix}$  is trained for 10 epochs with a batch-size of 256 using AdamW Optimizer [129, 164] with the learning rate of 0.001. Training takes 8 hours using two A100 GPUs. Once  $G_{mix}$  is trained, we use  $G_{id}$ ,  $G_{mix}$  and a style bank to generate a synthetic labeled dataset. The style bank is the CASIA-WebFace training set. For sampling, we use DDIM [215] with 200 intervals. Generating 500K samples takes about 20 hours using one A100 GPU.

To train FR models, for a fair comparison, we adopt the training scheme of [17, 188] using IR-SE-50 [55] as a backbone and AdaFace [122] as a loss function. We evaluate the trained FR models on five datasets, LFW [100], CFP-FP [202], CPLFW [296], AgeDB [174] and CALFW [297]. CFP-FP and CPLFW are designed to measure the FR in the large pose variation and AgeDB and CALFW are for the large age variation. To measure the consistency, diversity and uniqueness during evaluation, we adopt  $F_{eval}$  as IR101 [55] model trained on WebFace4M [300] with AdaFace [122] loss.

#### 4.5.1 Model Ablation

To show the efficacy of our proposed modules, we ablate on 1) the grid size in Style extractor  $E_{sty}$ , 2) Time-step dependent ID loss and 3) the ID loss backbone  $F$ 's. The number of samples we generate for the ablation are  $10K$  subjects with 50 images per subject, similar to CASIA-WebFace image counts. We report the FR performance with the synthetic data by

averaging the 5 validation set verification accuracies. To measure  $U_{class}$ ,  $C_{intra}$  and  $D_{intra}$ , we use 500 subjects with 20 real images from the held-out validation set of CASIA-WebFace and generate an equivalent number of images from each method.

**Grid Size** We choose 4 grid sizes ranging from  $1 \times 1$  to  $7 \times 7$ . Note that  $1 \times 1$  corresponds to the style vector of a whole image. We expect to see higher spatial control in  $\mathbf{X}_{sty}$  as the grid size increases. In Tab. 4.2, we report the three metrics  $U_{class}$ ,  $C_{intra}$  and  $D_{intra}$ . As the grid size increases,  $E_{sty}$  features contain more fine-grained information, possibly related to ID, lowering the consistency. However, the diversity increases, making the conditional distribution similar to the real dataset. The subsequent FR performance using the model is the best in the setting  $5 \times 5$ , which is a good compromise between consistency and diversity. In Fig. 4.7, we show the effect of the grid size with examples.

**ID Loss** For ID loss, we compare  $L_{ID}$  with  $L_{naive1}$  and  $L_{naive2}$  in Tab. 4.2. Using  $L_{naive1}$  or  $L_{naive2}$  both suffer from lower FR performance, but for different reasons.  $L_{naive1}$  has low diversity because it is optimized to be similar to  $\mathbf{X}_{id}$  of front-view high quality face images.  $L_{naive2}$  has low consistency because of the lack of dependence on  $\mathbf{X}_{id}$ , making the resulting dataset with random labels. FR performance of 0.5 means the model diverged and is returning random predictions.  $L_{ID}$ , a linear interpolation of the  $L_{naive1}$  and  $L_{naive2}$  across time-steps results in the best performance.

**ID Loss Backbone  $F$**  ID Loss requires a pretrained FR model,  $F$ . For all of our experiments, we use  $F$  as IR50 trained on CASIA-WebFace. But, we are curious if there is a benefit to have a better representation from  $F$ . For this, we ablate  $F_{bigger}$ , a model pretrained on a larger dataset, WebFace4M [300]. Tab. 4.2 shows that a better FR backbone induce the generator to synthesize better datasets, even without explicitly showing WebFace4M images to generators. But for fairness in comparing to the real CASIA-WebFace dataset, we do not use  $F_{bigger}$  for subsequent analysis.

ID	Style	LFW	CFPFP	CPLFW	AGEDB	CALFW	AVG
<i>random</i>	<i>random</i>	98.05	84.17	82.20	89.38	91.40	89.04
<i>random</i>	<i>match</i>	98.28	84.61	82.32	89.12	91.28	89.12
<i>balance</i>	<i>random</i>	98.30	83.27	81.60	89.40	91.27	88.77
<i>balance</i>	<i>match</i>	98.38	84.06	82.45	89.30	91.38	89.11
<i>balance</i>	<i>over smpl</i>	<b>98.55</b>	<b>85.33</b>	<b>82.62</b>	<b>89.70</b>	<b>91.60</b>	<b>89.56</b>

Table 4.3 Sampling Ablation. We generate a synthetic dataset of 10K subjects with 50 images per subject, using the setting indicated by the blue text in Tab. 4.2. *over smpl* is over-sampling  $\mathbf{X}_{id}$  during training for showing more front-view faces.

Methods	Venue	# images (# IDs× # imgs/ID)	LFW	CFP-FP	CPLFW	AgeDB	CALFW	Avg	Gap to Real
SynFace	ICCV21	0.5M (10K × 50)	91.93	75.03	70.43	61.63	74.73	74.75	26.58
DigiFace	WACV23	0.5M (10K × 50)	95.4	<b>87.4</b>	78.87	76.97	78.62	83.45	13.39
DCFace (Ours)	-	0.5M (10K × 50)	<b>98.55</b>	85.33	<b>82.62</b>	<b>89.70</b>	<b>91.60</b>	<b>89.56</b>	<b>5.65</b>
DigiFace	WACV23	1.2M (10K × 72 + 100K × 5)	96.17	<b>89.81</b>	82.23	81.10	82.55	86.37	9.55
DCFace (Ours)	-	1.0M (20K × 50)	<b>98.83</b>	88.4	84.22	90.45	92.38	90.86	4.14
DCFace (Ours)	-	1.2M (20K × 50 + 40K × 5)	98.58	88.61	<b>85.07</b>	<b>90.97</b>	<b>92.82</b>	<b>91.21</b>	<b>3.74</b>
CASIA-WebFace (Real)		0.49M (approx. 10.5K × 47)	99.42	96.56	89.73	94.08	93.32	94.62	0.0

Table 4.4 Verification accuracies of FR models trained with SoTA synthetic training datasets. SynFace [188] is a GAN-based dataset with a latent space mixup technique. DigiFace [17] is a 3D model-based dataset with heavy image augmentation. DCFace uses the model setting from the ablation study, Tab. 4.2, 4.3 indicated by blue colors. FR backbone is IR-SE50 [55] + AdaFace [122] to match the setting of DigiFace.

#### 4.5.2 Sampling Ablation

Using the sampling strategy defined in Sec. 4.3.3, we ablate on the ID sampling options (*random*, *balance*) and style sampling methods (*random*, *match*) in Tab. 4.3. We find that either balancing the gender/ethnicity distribution or making the gender/ethnicity of style image equal to that of ID images does not bring significant performance gain.

On the other hand, to compensate for lower label consistency compared to the real dataset, we include the same  $\mathbf{X}_{id}$  for 5 additional times for each label. This has the effect of oversampling  $\mathbf{X}_{id}$  during training FR model. When we add the oversampling option to (*balance*, *match*) setting, we observe an average verification accuracy of 89.56%, 0.52% increase over the (*random*, *random*) setting.

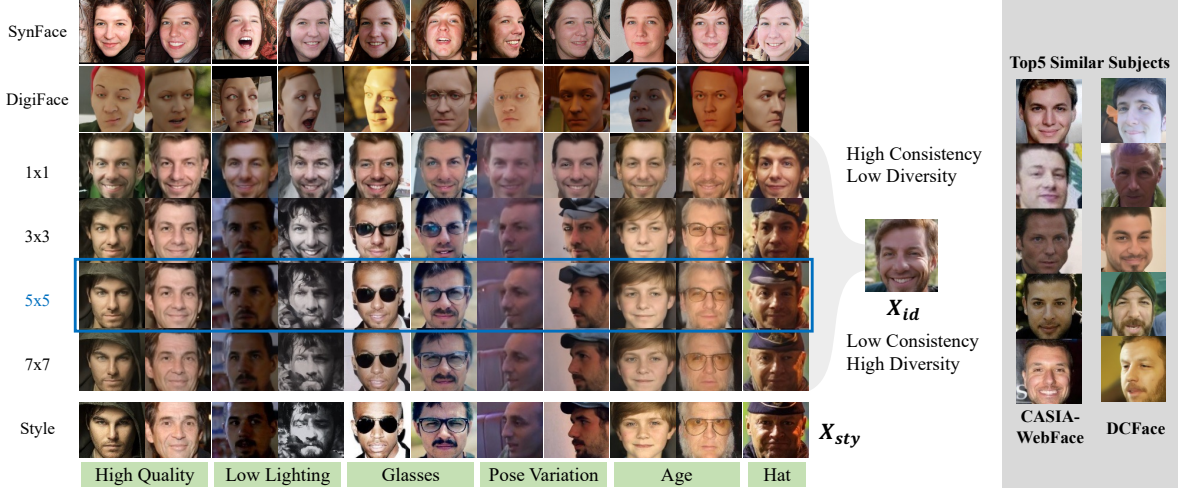


Figure 4.7 An example of SynFace and DigiFace in rows 1-2 and DCFace with different grid size settings in rows 3-7. SynFace (DiscoFaceGAN) generates mostly frontal-view high-quality images and DigiFace contains synthetic face images with unrealistic texture compared to real images. Our grid size ablation changes the contribution of  $\mathbf{X}_{sty}$  and  $\mathbf{X}_{id}$ . A good FR performance is a compromise in-between,  $5 \times 5$ . Note that our method can have diverse styles such as low lighting, pose, glasses, hat, etc. Using  $\mathbf{X}_{id}$  to query subjects in CASIA-WebFace and DCFace datasets returns top 5 most similar subjects. We see  $\mathbf{X}_{id}$  sufficiently different from other (real or fake) subjects.

#### 4.5.3 Comparison with Previous Methods

For training FR models with synthetic datasets, we compare with SynFace [188] and DigiFace [17]. We compare 0.5M and 1.2M image count settings. The first setting corresponds to the size of the CASIA-WebFace real dataset. The second setting is to evaluate the effect of increasing the training dataset size. In Tab. 4.4, we show the verification accuracies of 5 validation sets. In 0.5M regime, our DCFace can surpass DigiFace in 4 out of 5 datasets with an improvement of 6.11% on average. In CFP-FP dataset with extremely large pose variation, DigiFace performs better, showing the merit of 3D consistent face synthesis using 3D models. DCFace has a good balance of consistency and diversity with many unique subjects, leading to a better FR performance in general. Note the larger style variation compared to SynFace and DigiFace in Fig. 4.7.

The last column of Tab. 4.4 shows the gap between synthetic and real, calculated as  $(\text{REAL} - \text{SYN}) / \text{SYN}$ , e.g.  $5.65\% = \frac{94.62 - 89.56}{89.56}$ . It indicates how much improvement is needed to be on par with the real dataset. In 0.5M setting, DCFace reduces the gap to real performance by 57% over the SoTA. When we use more synthetic data as in 1.2M regime, the synthetic dataset performance comes closer to that of the real dataset (3.74% in gap), a 60.9% improvement from the previous method (9.55% in gap).

## 4.6 Training Details

### 4.6.1 Architecture Details

The dual condition generator  $G_{mix}$  is a modification of DDPM [91] to incorporate two conditions. We insert two conditions  $\mathbf{X}_{id}$  and  $\mathbf{X}_{sty}$  into the denoising U-Net  $\epsilon_\theta(\mathbf{X}_t, t, \mathbf{X}_{id}, \mathbf{X}_{sty})$ . Conditioning images  $\mathbf{X}_{sty}$  and  $\mathbf{X}_{id}$  are mapped to features using  $E_{sty}$  and  $E_{id}$ , respectively. According to Eq. 6 of the main paper, the style information  $E_{sty}(\mathbf{X}_{sty})$  is the concatenation of style vectors at different  $k \times k$  patch locations,

$$E_{sty}(\mathbf{X}_{sty}) := \mathbf{s} = [\mathbf{s}^1, \mathbf{s}^2, \mathbf{s}^{k_i}, \dots, \mathbf{s}^{k \times k}, \mathbf{s}'] \in \mathbb{R}^{(k^2+1) \times C}. \quad (4.16)$$

On the other hand, ID information is a concatenation of features extracted from a trainable CNN (e.g. ResNet50 [86]), which produces an intermediate feature  $\mathbf{I}_{id}$  of shape  $\mathbb{R}^{7 \times 7 \times 512}$  and a feature vector  $\mathbf{f}_{id}$  of shape  $\mathbb{R}^{512}$ . Specifically,

$$E_{id}(\mathbf{X}_{id}) := \mathbf{i} = [\text{Flatten}(\mathbf{I}_{id}), \mathbf{f}_{id}] + \mathbf{P}_{emb} \in \mathbb{R}^{50 \times C}, \quad (4.17)$$

where Flatten refers to removing the  $H \times W$  spatial dimension and  $\mathbb{R}^{50 \times C}$  is from concatenating features of length  $7 \times 7$  and 1.  $\mathbf{P}_{emb}$  is a learnable position embedding for distinguishing each feature position for the subsequent cross-attention operation. Detailed illustrations of  $E_{sty}(\mathbf{X}_{sty})$  and  $E_{id}(\mathbf{X}_{id})$  are shown in Fig. 4.8.  $C$  for the channel dimension of  $E_{sty}(\mathbf{X}_{sty})$  and  $E_{id}(\mathbf{X}_{id})$  is 512.

When  $E_{sty}(\mathbf{X}_{sty})$  and  $E_{id}(\mathbf{X}_{id})$  is prepared, they together form  $(k^2 + 1) + 50$  vectors of shape 512. These can be injected into the U-Net  $\epsilon_\theta$  by following the convention of the DDPM based text-conditional image generators [190]. Specifically, cross attention operation can be

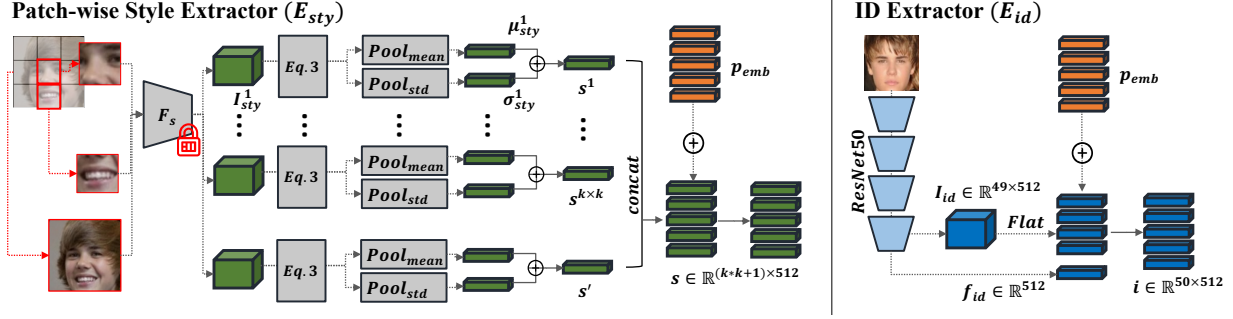


Figure 4.8 Left: An illustration of  $\mathbf{X}_{sty}$ . The key property of  $\mathbf{X}_{sty}$  is in restricting the information in  $\mathbf{X}_{sty}$  from flowing freely to the next layer. The fixed feature encoder  $F_s$  and the patch-wise spatial mean-variance operation destroy the detailed ID information while preserving the style of an image. We create an output of size  $\mathbb{R}^{(k^2+1) \times C}$ . Right: A simple CNN based on ResNet50. We take intermediate representation and the last feature vector and concatenate them together to create a output of size  $\mathbb{R}^{50 \times C}$ .

written as a modification of attention equation [236] with query  $\mathbf{Q}$ , key  $\mathbf{K}$  and value  $\mathbf{V}$  with additional query  $\mathbf{Q}_c$ , key  $\mathbf{K}_c$ .

$$\text{Attn}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{SoftMax} \left( \frac{\mathbf{Q}\mathbf{W}_q (\mathbf{K}\mathbf{W}_k)^\top}{\sqrt{d}} \right) \mathbf{W}_v \mathbf{V}, \quad (4.18)$$

$$\text{Cross-Attn}(\mathbf{Q}, \mathbf{K}, \mathbf{V}, \mathbf{K}_c, \mathbf{V}_c) = \text{SoftMax} \left( \frac{\mathbf{Q}\mathbf{W}_q ([\mathbf{K}, \mathbf{K}_c]\mathbf{W}_k)^\top}{\sqrt{d}} \right) \mathbf{W}_v [\mathbf{V}, \mathbf{V}_c], \quad (4.19)$$

where  $\mathbf{W}_q$ ,  $\mathbf{W}_k$  and  $\mathbf{W}_v$  are learnable weights and  $[\cdot]$  refers to concatenation operation. In our case,  $\mathbf{Q} = \mathbf{K} = \mathbf{V}$  are an arbitrary intermediate feature in the U-Net. And  $\mathbf{K}_c = \mathbf{V}_c$  are conditions generated by  $E_{sty}(\mathbf{X}_{sty})$  and  $E_{id}(\mathbf{X}_{id})$ , concatenated together. This operation allows the model to update the intermediate features with the conditions if necessary. We insert the cross-attention module in the last two DownSampling Residual Blocks in the U-Net, as shown in Fig. 4.9.

To increase the effect of  $\mathbf{X}_{id}$  in the conditioning operation, we also add  $\mathbf{f}_{id}$  to the time-step embedding  $\mathbf{t}_{emb}$ . As shown in the right side of Fig. 4.9, the Residual Block in the U-Net modulates the intermediate features according to the scaling vector provided by  $\mathbf{f}_{id} + \mathbf{t}_{emb}$ . GNorm [259] refers to Group Normalization and SiLU refers to Sigmoid Linear Units [63]. Adding  $\mathbf{f}_{id}$  to  $\mathbf{t}_{emb}$  for the Residual Block allows more paths for  $\mathbf{X}_{id}$  to change the output of

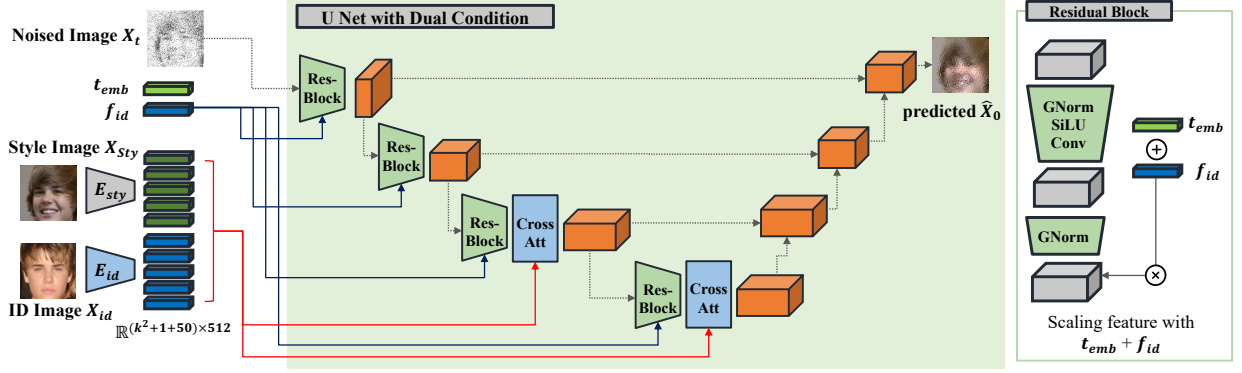


Figure 4.9 Illustration of DDPM U-Net with conditioning operations highlighted. The red arrow indicates how the dual conditions are injected into the intermediate features of U-Net using cross-attention layers. For clarity, up-sampling stages are not illustrated, but they are symmetric to the down-sampling stages. On the right is a detailed illustration of the Residual Block with timestep and ID condition.  $t_{emb}$  and  $f_{id}$  from  $E_{id}$  are added together and used to scale the output of the Residual Block.

U-Net.

#### 4.6.2 Training Hyper-Parameters

The final loss for training the model end-to-end is  $L_{MSE} + \lambda L_{ID}$  with  $\lambda$  as a scaling parameter. We set  $\lambda = 0.05$  to compensate for the different scale between L2 and Cosine Similarity. All our input image sizes are  $112 \times 112$ , following the convention of SoTA face recognition model datasets [55, 99, 300]. And our code is implemented in Pytorch.

### 4.7 More Experiment Results

#### 4.7.1 Adding Real Dataset

We include additional experiment results that involve adding real images. Although the motivation of the paper is to use an only-synthetic dataset to train a face recognition model, the performance comparison with an addition of a subset of the real dataset has its merits; it shows 1) whether the synthetic dataset is complementary to the real dataset and 2) whether the synthetic dataset can work as an augmentation for real images.

Tab. 4.5 shows the performance comparison between DigiFace [17] and our proposed

DCFace when 1) a few real images are added and 2) both synthetic datasets are combined. The performance gap for DigiFace is large, jumping from 86.37 to 92.67 on average when 2K real subjects with 20 images per subject are added. In contrast, ours show a relatively less dramatic gain, 91.21 to 92.90 when few real images are added. This indicates that DigiFace [17] is quite different from the real images and ours is similar to the real images. This is in-line with our expectation as we have created a synthetic dataset that tries to mimic the style distribution of the training dataset, whereas DigiFace simulates image styles using 3D models.

#### 4.7.2 Combining Multiple Synthetic Datasets

In the second to the last row of Tab. 4.5, when we combined the two synthetic datasets without the real images, the performance is the highest, reaching 93.06 on average. This result indicates that different synthetic datasets can be complementary when they are generated using different methods.

	# Synthetic Imgs	# Real Imgs	LFW	CFPFP	CPLFW	AGEDB	CALFW	AVG	Gap to Real
DigiFace	1.2M (10K×72+100K×5)	0	96.17	89.81	82.23	81.10	82.55	86.37	8.72
DigiFace	1.2M (10K×72+100K×5)	2K×20	99.17	94.63	88.1	90.5	90.97	92.67	2.06
DCFace	1.2M (20K×50+40K×5)	0	98.58	88.61	85.07	90.97	92.82	91.21	3.61
DCFace	1.2M (20K×50+40K×5)	2K×20	98.97	94.01	86.78	91.80	92.95	92.90	1.82
DCFace+DigiFace (2.4M)		0	99.20	93.63	87.25	92.25	92.95	93.06	<b>1.65</b>
CASIA	0	0.5M	99.42	96.56	89.73	94.08	93.32	94.62	0

Table 4.5 Verification accuracies of FR models trained with synthetic datasets and subset of real datasets. In all settings, the backbone is set to IR50 [55] model with AdaFace loss [122] for a fair comparison.

## 4.8 Analysis

§C.1 Unique Subject Counts In Fig. 4.10, we plot the number of unique subjects that can be sampled as we increase the sample size. The blue curve shows that the number of unique samples that can be generated by a DDPM of our choice does not saturate when we sample 200,000 samples. At 200,000 samples, the unique subjects are about 60,000. And by extrapolating the curve, we estimate the number might reach 80,000 with more samples.

Our DDPM of choice is trained on FFHQ [116] dataset which contains 70,000 unlabeled high-quality images. The orange line shows the number of unique samples that are sufficiently different from the subjects in the CASIA-WebFace dataset. The green line shows the number of unique samples left after filtering images that contain sunglasses. The flat region is due to the filtering stage reducing the total candidates. The plot shows that DDPM trained on FFHQ dataset can sufficiently generate a large number of unique and new samples that are different from CASIA-WebFace dataset. However, with more samples, eventually there is a limit to the number of unique samples that can be generated. When the number of total generated samples is 100,000, one additional sample has approximately 24% chance of being unique, whereas, at 200,000, the probability is 15%. The rate of sampling another unique subject decreases with more samples. The model used for evaluating the uniqueness is IR101 [55] trained on the WebFace4M [300] dataset. And we use the threshold of 0.3. We would like to note a typo in Sec. 3.3 of the main paper, where the number of unique subjects should be corrected from 62,570 to 42,763.

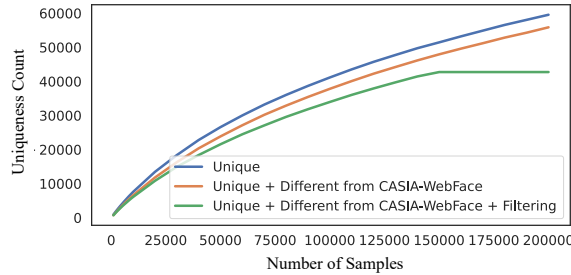


Figure 4.10 Plot of unique subject count as the number of samples from  $G_{id}$  is increased from 1000 to 200,000. At 200,000, one additional sample has approximately 15% chance of being unique. And the rate decreases with more samples.

**§C.2 Feature Plot** In Fig. 4.11, we show the 2D t-SNE [235] plot of synthetic images generated by 3 different methods (DiscoFaceGAN [59], DigiFace [17] and proposed DCFace). The red circles represent real images from CASIA-WebFace. We extract the features from each image using a pre-trained face recognition model, IR101 [55] trained on WebFace4M [300]. We show two settings we sample (a) 50 subjects with 1 image per subject and (b) 1 subject

with 50 images per subject. Note that the proximity of DCFace image features is closer to CASIA-WebFace image features, highlighted in a circle. For each setting, we show the features extracted from an intermediate layer of IR101 and the last layer. As the layer becomes deeper, the features become suitable for recognition, as shown in the last column of the figure.

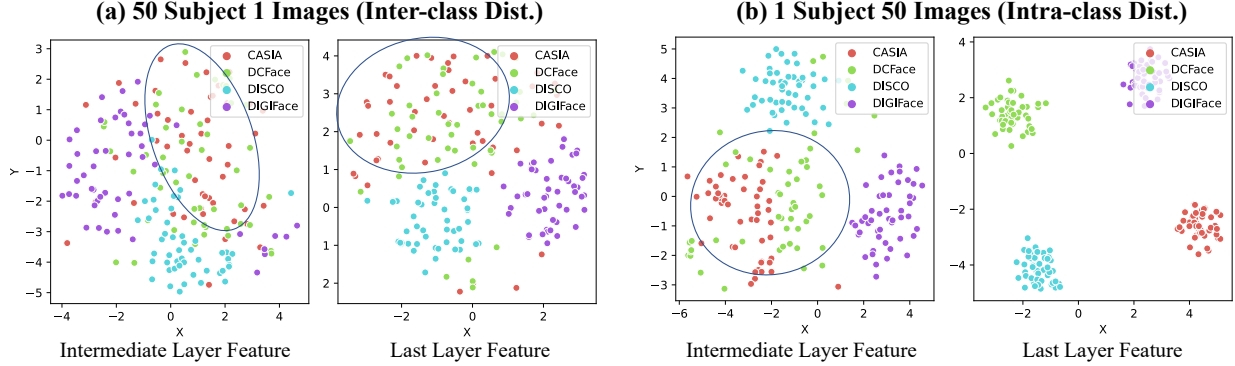


Figure 4.11 (a) the t-SNE plot of features from synthetic and real datasets of 50 subjects per dataset. It shows how 50 randomly sampled subjects from each dataset are distributed. The distribution between real (red) and DCFace (green) is the closest. (b) the t-SNE plot of features from synthetic and real datasets of 1 subject per dataset with 50 images. We randomly sample 1 subject from each dataset. The last layer features are well separated as the model is a face recognition model that separates the features of different subjects.

### §C.3 Comparison with Classifier Free Guidance

When  $\epsilon(x_t, c)$  learns to use the condition  $c$ , the difference  $\epsilon(x_t, c) - \epsilon(x_t)$  can give further guidance during sampling to increase the dependence on  $c$ . But, in our case, the ID condition is the fine-grained facial difference that is hard to learn with MSE loss. Proposed Time-dependent ID loss,  $L_{ID}$  helps the model learn this directly. Row 3 vs 4 of Tab. 4.6 shows that  $L_{ID}$  is more effective than CFG.

Interestingly, with a large guidance scale, CFG becomes harmful. CFG decreases diversity as pointed out by [92]. We observe that guidance with  $X_{id}$  leads to consistent ID but with little facial variation, the same phenomenon in DCFace with grid-size 1x1 in  $E_{sty}$ , in Tab. 2

	Conditions	Train Loss	Sampling	FR.Perf $\uparrow$
1	CNN( $X_{id}$ ), CNN( $X_{sty}$ )	MSE	+ Guide	73.38
2	CNN( $X_{id}$ ), $E_{sty}(X_{sty})$	MSE	$\times$	82.30
3	CNN( $X_{id}$ ), $E_{sty}(X_{sty})$	MSE	+ Guide	84.05
4	CNN( $X_{id}$ ), $E_{sty}(X_{sty})$	MSE+ $L_{ID}$	$\times$	<b>89.56</b>

Table 4.6 Green  $E_{sty}$  and  $L_{ID}$  indicates the novelty of our paper. For guidance, we adopt 10% condition masking during training and the guidance scale of 3 during sampling. FR.Perf is an average of 5 face recognition performances as in the main paper.

(main). Good FR datasets need both large intra and inter-subject variability and we combine  $E_{sty}$  and  $L_{ID}$  to achieve this.

§C.4 FID Scores Note that our generated data is not high-res images like FFHQ when compared to how SynFace is similar to FFHQ. (Tab. 4.7 row 5 vs 6). But, we point out that our aim is not to create HQ images but to create a *database* with realistic inter/intra-subject variations. In that regard, we have successfully approximated the distribution of the popular FR training dataset CASIA-WebFace (FID=13.67).

	Generator Train Data	Source (real/syn)	Target (real)	FID $\downarrow$
1	-	CASIA (train)	CASIA (val)	<b>9.57</b>
2	CASIA (train)	DCFace	CASIA (val)	<b>13.67</b>
3	FFHQ+3DMM	SynFace	CASIA (val)	38.48
4	3D Face Capture	DIGIFACE1M	CASIA (val)	71.65
5	CASIA (train)	DCFace	FFHQ (train+val)	35.45
6	FFHQ+3DMM	SynFace	FFHQ (train+val)	<b>21.75</b>
7	3D Face Capture	DIGIFACE1M	FFHQ (train+val)	68.67

Table 4.7 FID scores of synthetic vs real datasets. For synthetic datasets, we randomly sampled 10,000 images. See Line 630 for Casia-WebFace Train and Val set split. All images are aligned and cropped to  $112 \times 112$  to be in accordance with CASIA-WebFace.

Having said this, we note FID is not comprehensive in evaluating labeled datasets. It cannot capture the label consistency nor directly relate to the FR performance. As such, SynFace/DigiFace do not report FID. We propose U,D,C metrics that enable holistic analysis of labeled datasets.

§C.5 Does DCFace change gender? DCFace combines  $X_{ID}$  and  $X_{sty}$ , while adhering to the subject ID as defined by a pre-trained FR model. Factors weakly related to ID, such as age and hair style, can vary. Biometric ambiguity can occur due to makeup, wig, weight change, *etc.* even in real life. The perceived gender may change, but changes such as hair are less relevant to subject ID for the FR model.

§C.6 Why DCFace is better in U,D,C metrics? We note DCFace is not better in all U,D,C. Fig. 6 (main) shows SynFace has the highest consistency (C). But, DCFace excels in the tradeoff between C and D. In other words, style similarity to the real dataset (*i.e.* D) is lacking in other datasets and it is as important as ID consistency. As such, U,D,C metrics reveal weak/strong points of synthetic datasets.

## 4.9 Visualizations

### 4.9.1 Time-step Visualizaton

Fig. 4.12 shows how DDPM generates output at each time-step. The far left column shows  $X_{sty}$ , the desired style of an image. The far right column shows  $X_{id}$ , the desired ID image of choice. In early time-steps, the network reconstructs the front-view image with an ID of  $X_{id}$ . And gradually, it interpolates the image into the desired style of  $X_{sty}$ . The gradual transition can be in the pose, hair-style, expression, *etc.*



Figure 4.12 A plot of DCFace outputs at each time-step.

### 4.9.2 Interpolation

In Fig. 4.13, we show the plot of interpolation in  $\mathbf{X}_{sty}$ . While keeping the same identity  $\mathbf{X}_{id}$ , we take two style images  $\mathbf{X}_{sty1}$  and  $\mathbf{X}_{sty2}$ . We interpolate with  $\alpha$  in  $\alpha E_{stry}(\mathbf{X}_{sty1}) + (1 - \alpha)E_{stry}(\mathbf{X}_{sty2})$  with  $\alpha$  increasing linearly from 0 to 1. The interpolation is smooth, creating an intermediate pose and expression that did not exist before.



Figure 4.13 A plot of DCFace output with style interpolation.

### 4.10 Miscellaneous

**Similarity threshold** Threshold=0.3 is based on FR evaluation model having a threshold of 0.3080 for verification with TPR@FPR=0.01% : 97.17% on IJB-B [253]. FPR=0.01% is widely used in practice and the scale of similarity is  $(-1, 1)$ . At threshold=0.3, FFHQ has 200 (2%) more unique subjects than DDPM, signaling a similar level of uniqueness.

**Style Extracting Model** We use the early layers of face recognition model for style extractor backbone. Our rationale for adopting the early layers of the FR model, as opposed to that of the ImageNet-trained model is that the early layers extract low-level features and we wanted features optimized with the face dataset. But, it is possible to take other models as long as it generates low-level features.

**Evaluation on Harder Datasets** We evaluate on IJB-B [253] (TPR@FPR=0.01%: 75.12) and TinyFace [46] (Rank1: 41.66). We include this result for future works to evaluate on harder datasets.

**Real and Generated Similarity Analysis** In addition to Fig.7 matching  $\hat{X}_{id}$  with CASIA-WebFace, matching all  $\hat{X}_0$  (generated) images against CASIA-WebFace at threshold=0.3, we get 0.0026% FMR. This implies that only a small fraction of CAISA-WebFace images are similar to the generated images.

#### 4.11 Societal Concerns

We believe that the Machine Learning and Computer Vision community should strive together to minimize the negative societal impact. Our work falls into the category of 1) image generation using generative models and 2) synthetic labeled dataset generation. In the field of image generation, unfortunately, there are numerous well-known malicious applications of generative models. Fake images can be used to impersonate high-profile figures and create fake news. Conditional image generation models make the malicious use cases easier to adapt to different use cases because of user controllability. Fortunately, GAN-based generators produce subtle artifacts in the generated samples that allow the visual forgery detection [14, 76, 245, 279]. With the recent advance in DDPM, the community is optimistic about detecting forgeries in diffusion models [203]. It is also known that proactive treatments on generated images increase the forgery detection performance [14], and as generative models become more sophisticated, proactive measures may be advised whenever possible.

Synthetic dataset generation is, on the other hand, an effort to avoid infringing the privacy of individuals on the web. Large-scale face dataset is collected without informed consent and only a few evaluation datasets such as IJB-S [112] has IRB compliance for safe and ethical research. Collecting large-scale datasets with informed consent is prohibitively challenging and the community uses web-crawled datasets for the lack of an alternative option. Therefore, efforts to create synthetic datasets with synthetic subjects can be a practical solution to this

problem. In our method, we still use real images to train the generative models. We hope that research in synthetic dataset generation will eventually replace real images, not just in the recognition task, but also in the generative tasks as well, removing the need for using real datasets in any form.

#### 4.12 Implementation Details and Code

The code will be released at <https://github.com/mk-minchul/dcfacer>. For preprocessing the training data CASIA-WebFace [99], we reference AdaFace [122] and use MTCNN [285] for alignment and cropping faces. For the backbone model definition, TFace [3] and for evaluation of LFW [100], CFP-FP [202], CPLFW [296], AgeDB [174] and CALFW [297], we use AdaFace repository [122].

#### 4.13 Conclusion

This paper presents a method for creating a synthetic training dataset for face recognition. Dataset generation is studied from the perspective of generating many unique subjects with large style diversity and label consistency. We propose the Dual Condition Face Generator to this end and show its large FR performance gain over previous methods on synthetic dataset generation. We believe our approach takes one step towards matching the performance of real training datasets with synthetic training datasets.

**Limitations** This work addresses the problem of generating label consistent and diverse datasets for face recognition model training. In our model ablation, we find that sacrificing label consistency for diversity to some degree is beneficial for the FR model training. However, this is not ideal; for instance, our synthetic face generator lacks 3D consistency across pose, which is an advantage of generative models with 3D priors. Secondly, the goal of our research is to release a synthetic face dataset that alleviates the dependence on large-scale web-crawled images. As shown in our experiments, there is still some performance gap between real and synthetic training datasets. In this work, we take one step towards the goal and hope that the continued research will introduce a standalone synthetic face dataset.

## CHAPTER 5

### KEYPOINT RELATIVE POSITION ENCODING FOR FACE RECOGNITION

In this paper, we address the challenge of making ViT models more robust to unseen affine transformations. Such robustness becomes useful in various recognition tasks such as face recognition when image alignment failures occur. We propose a novel method called KP-RPE, which leverages key points (e.g. facial landmarks) to make ViT more resilient to scale, translation, and pose variations. We begin with the observation that Relative Position Encoding (RPE) is a good way to bring affine transform generalization to ViTs. RPE, however, can only inject the model with prior knowledge that nearby pixels are more important than far pixels. Keypoint RPE (KP-RPE) is an extension of this principle, where the significance of pixels is not solely dictated by their proximity but also by their relative positions to specific keypoints within the image. By anchoring the significance of pixels around keypoints, the model can more effectively retain spatial relationships, even when those relationships are disrupted by affine transformations. We show the merit of KP-RPE in face and gait recognition. The experimental results demonstrate the effectiveness in improving face recognition performance from low-quality images, particularly where alignment is prone to failure. Code and pre-trained models are available.

#### 5.1 Introduction

Geometric alignment has shown to be highly effective for certain computer vision problems, such as face, body and gait recognition [55, 56, 100, 122, 128, 131, 139, 149, 154, 169, 172, 174, 202, 240, 253, 289, 290, 292, 296]. Alignment is the process of transforming input images, to a consistent and standardized form, often by scaling, rotating, and translating. This standardization helps recognition models learn the underlying patterns and features more effectively. As a result, many state-of-the-art (SoTA) face recognition models [55, 122, 172, 240] rely on well-aligned datasets [54, 55, 82, 300] to achieve high accuracy.

Fig. 5.1 shows a toy example with a training dataset MNIST [58] and test set AffNIST [197] which is in unseen affine transformation of MNIST. Using a shallow ViT [61] model, one can

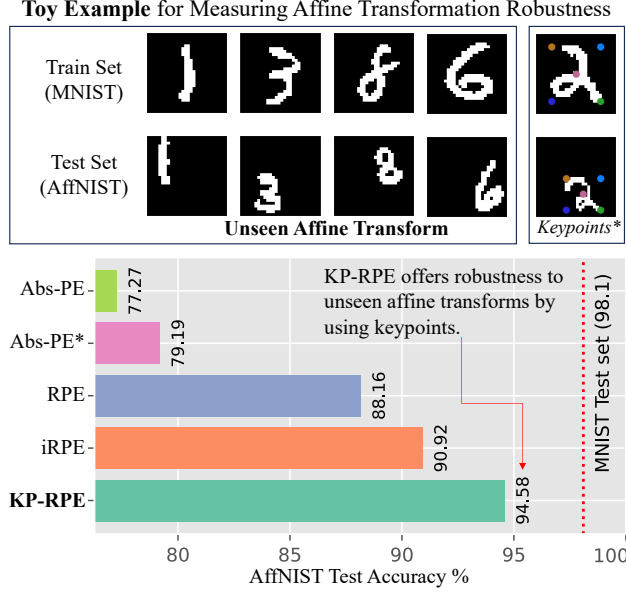


Figure 5.1 Toy Example illustrating how different Position Embeddings impact the ViT’s robustness to unseen affine transforms. Abs-PE refers to the learnable Absolute Position Embedding. RPE and iRPE refers to Relative Position Embedding adopted to ViT [105, 256]. Keypoints in MNIST is arbitrarily defined to be the four corners of a box that covers a digit. Abs-PE\* is drawing the keypoints onto the input image. KP-RPE uses the keypoints to adjust the RPE.

easily achieve 98.1% accuracy in the MNIST test set. However, in AffNIST, ViT with the original Absolute Position Embedding obtains only 77.27% accuracy. Such a sharp decrease in performance with unseen affine transform causes problems in applications that rely on accurate input alignments.

In face recognition, alignment can be imperfect, especially in low-quality images where accurate landmark detection is difficult [54, 148]. Thus, images with low resolution or taken in poor lighting may result in misalignment during testing. Given the interplay between alignment and recognition, it becomes crucial to proactively handle potential alignment failures, which often result from, *e.g.*, low-quality images. In other words, there is a need for a recognition model that is robust to scale, rotation, and translation variations.

We revisit the Relative Position Encoding (RPE) concept used in ViT [61] and find

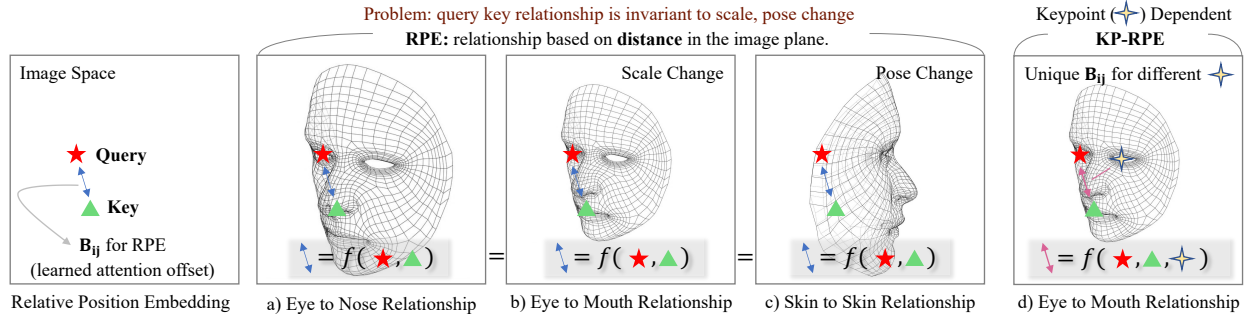


Figure 5.2 Illustration of RPE [205] and proposed KP-RPE. The blue arrow represents the learned attention offset  $\mathbf{B}_{ij}$  between a query  $i$  and key  $j$  of attention in RPE. The query-key relationship at the same  $i$  and  $j$  should represent different relationships as the scale or pose change. But  $\mathbf{B}_{ij}$  does not change in RPE. KP-RPE addresses this issue by incorporating the *distance to the keypoints* when calculating the learned attention offset in RPE.

that RPE can be useful for introducing affine transform robustness. RPE [205] enables the model to capture the relative spatial relationships among regions of an image, learning the positional dependencies without relying on absolute coordinates. As shown in Fig. 5.1, adding RPE to ViT increases the performance in AffNIST. With RPE [205], queries and keys of self-attention [236] at closer distances can be assigned different attention weights compared to those at a greater distance. While RPE allows the model to exploit relative positions, it has a limitation: even if an image changes in terms of scaling, shifting, or orientation, the significance of the key-query position in RPE stays the same. This static behavior is illustrated in Figs. 5.2 a)-c). Notably, the key-query relationship is the same regardless of the corresponding pixels’ semantic meaning changes.

We hypothesize that an RPE which dynamically adapts based on image keypoints, such as facial landmarks, could improve the model’s comprehension of spatial relationships in the image. By leveraging the spatial relationships with respect to these keypoints, the model can adapt to variations in scale, rotation, and translation, resulting in a more robust recognition system capable of handling both aligned and misaligned datasets. Fig. 5.2 d) highlights a keypoint-dependent query-key relationship.

To this end, we introduce KeyPoint RPE (KP-RPE), a method that dynamically adapts

the spatial relationship in ViT based on the keypoints present in the image. Our experiments demonstrate that incorporating KP-RPE into ViT significantly enhances the model’s robustness to misaligned test datasets while maintaining or even improving performance on aligned test datasets. We show the usage of KP-RPE in face recognition and gait recognition as the inputs share the same topology (face or body) that allows the keypoints to be defined. Finally, KP-RPE is an order of magnitude faster than iRPE [256], a widely used RPE that depends on the image content.

In summary, the contributions of this paper include:

- The insight that RPE (or its variants) can improve the robustness of ViT to unseen affine transformations.
- The development of Keypoint RPE (KP-RPE), a novel method that dynamically adapts the spatial relationship in Vision Transformers (ViT) based on the keypoints in the image, significantly enhancing the model’s robustness to misaligned test datasets while maintaining or improving performance on aligned test datasets.
- Comprehensive experimental validation demonstrating the effectiveness of our proposed KP-RPE, showcasing its potential for advancing the field of recognition by bringing model’s robustness to geometric transformation. We improve the recognition performance across unconstrained face datasets such as TinyFace [46] and IJB-S [112] and even non-face datasets such as Gait3D [67, 292].

## 5.2 Related Works

**Relative Position Encoding in ViT** Relative Position Encoding (RPE) is first introduced by Shaw *et al.* [205] as a technique for encoding spatial relationships between different elements in a sequence. By adding relative position encodings into the queries and keys, the model can effectively learn positional dependencies without relying on absolute coordinates. Subsequent works, such as those by Dai *et al.* [50] and Huang *et al.* [105], refine and expand upon the concept of RPE, demonstrating its effectiveness in natural language processing (NLP) tasks.

The adoption of RPE in Vision Transformers [61] has been explored by several researchers.

For instance, Ramachandran *et al.* [189] propose a 2D RPE method that computes the  $x$ ,  $y$  distance in an image plane separately to include directional information. A notable RPE method in ViT is iRPE [256], which considers directional relative distance modeling as well as the interactions between queries and relative position encodings in a self-attention mechanism.

Despite the success of these RPE methods in various vision tasks, they do not specifically address the challenges associated with scale, rotation, and translation variations in face recognition applications. This shortcoming highlights the need for RPE methods that can better handle these variations, which are common in real-world low-quality face recognition scenarios. To address this, we propose KP-RPE, which incorporates keypoint information during the network’s feature extraction, significantly enhancing the model’s ability to generalize across affine transformations.

**Keypoints and Spatial Reasoning** Keypoint detection, often associated with landmarks, has been fundamental in various vision tasks such as human pose estimation [35, 177], face landmark detection [31, 132, 224, 285], and object localization [183]. These keypoints serve as representative points that capture the essential structure or layout of an object, facilitating tasks like alignment, recognition, and even animation.

Face landmark detection is commonly carried out alongside face detection. MTCNN [285] is a widely-used method for combined face detection and facial landmark localization, utilizing cascaded CNNs (P-Net, R-Net, and O-Net) that collaborate to detect faces and landmarks in an image. RetinaFace [54], on the other hand, is a single-stage detector [144, 153] based landmark localization algorithm, demonstrating strong performance when trained on the annotated WiderFace [269] dataset. TinaFace [299] further enhances detection capabilities by incorporating SoTA generic object detection algorithms. MTCNN and RetinaFace are often used for aligning face datasets.

Recent advances in keypoint detection techniques, particularly using deep neural networks, have led to using keypoints to improve the performance of recognition tasks [220, 265]. For instance, [83] proposes a keypoint-based pooling mechanism and shows promising results

in skeleton-based action recognition and spatio-temporal action localization tasks. Albeit its benefit, many models including ViTs do not have pooling mechanisms. KP-RPE is the first attempt at incorporating keypoints into the RPE which can be easily inserted into ViT models.

### 5.3 Proposed Method

#### 5.3.1 Background

**Self-Attention** Self-attention is a crucial component of transformers [236], which is a popular choice for a wide range of NLP tasks. ViT [61] applies the same self-attention mechanism to images, treating images as sequences of non-overlapping patches. The self-attention mechanism in Transformers calculates attention weights based on the compatibility between a query and a set of keys. Given a set of input vectors, the Transformer computes query ( $\mathbf{Q}$ ), key ( $\mathbf{K}$ ), and value ( $\mathbf{V}$ ) matrices through linear transformations:

$$\mathbf{Q}_i = \mathbf{x}_i \mathbf{W}_Q, \quad \mathbf{K}_j = \mathbf{x}_j \mathbf{W}_K, \quad \mathbf{V}_j = \mathbf{x}_j \mathbf{W}_V, \quad (5.1)$$

where  $\mathbf{x}_i$  is the  $i$ -th input vector, and  $\mathbf{W}_Q$ ,  $\mathbf{W}_K$ , and  $\mathbf{W}_V$  are learnable weight matrices.

The self-attention mechanism computes attention weights as the dot product between the query and key vectors, followed by a softmax normalization:

$$e_{ij} = \frac{\mathbf{Q}_i \mathbf{K}_j^T}{\sqrt{d_k}}, \quad a_{ij} = \frac{\exp(e_{ij})}{\sum_{j=1}^N \exp(e_{ij})}, \quad (5.2)$$

where  $d_k$  is the dimension of the key vectors. Finally, the output matrix  $\mathbf{Y}$  is computed as the product of the attention weight matrix and the value matrix:  $\mathbf{Y}_i = \sum_{j=1}^N a_{ij} \mathbf{V}_j$ .

#### Absolute Position Encoding

Transformers are inherently order invariant, as their self-attention mechanism does not consider input token positions. To address this, absolute position encoding is introduced [74, 236], which adds fixed, learnable positional embeddings to input tokens:

$$\mathbf{x}'_i = \mathbf{x}_i + \text{PE}(i), \quad (5.3)$$

where  $\mathbf{x}'_i$  is the updated input token with positional information,  $\mathbf{x}_i$  is the original input token, and  $\text{PE}(i)$  is the positional encoding for the  $i$ -th position. These embeddings, generated using

sinusoidal functions or learned directly, enable the model to capture the absolute positions of elements.

### Relative Position Encoding (RPE)

RPE, introduced by Shaw *et al.* [205] and refined by Dai *et al.* [50] and Huang *et al.* [105], encodes relative position information, essential for tasks focusing on input element relationships. Unlike absolute position encoding, RPE considers query-key interactions based on sequence-relative distances. The modified self-attention calculation for RPE is:

$$e'_{ij} = \frac{(\mathbf{Q}_i + \mathbf{R}_{ij}^Q)(\mathbf{K}_j + \mathbf{R}_{ij}^K)^T}{\sqrt{d_k}}, \mathbf{Y}_i = \sum_{j=1}^n a_{ij}(\mathbf{V}_j + \mathbf{R}_{ij}^V). \quad (5.4)$$

Here,  $\mathbf{R}_{ij}^Q$ ,  $\mathbf{R}_{ij}^K$ , and  $\mathbf{R}_{ij}^V$  are relative position encoding between the  $i$ -th query and  $j$ -th key with shape  $\mathbb{R}^{d_z}$ . Each  $\mathbf{R}$  is a learnable matrix of  $\mathbb{R}^{K \times d_z}$ , where  $\mathbf{R}_{i,j}$  corresponds to the relative position encoding for distance  $d(i, j) = k$  and  $K$  is the maximum possible value of  $d(i, j)$ . To obtain relative position encoding, we index the  $\mathbf{R}$  matrix using the computed distance  $\mathbf{R}[d(i, j)]$ . Common choices for  $d$  are quantized Euclidean distance, separate  $x, y$  cross distance [189]. [256] uses a quantized  $x, y$  product distance, which encodes direction information. Note, query location  $i$  is a 2D point  $(i_x, i_y)$ . Fig. 5.3 a) and b) illustrate the distance between  $i$  and all possible  $j$  with different distance functions. For KP-RPE, we modify [256] and allow the RPE to be keypoint dependent.

#### 5.3.2 Keypoint Relative Position Encoding

Building upon the general formulation of [256], we begin with the following RPE formulation:

$$e'_{ij} = \frac{\mathbf{Q}_i \mathbf{K}_j^T + \mathbf{B}_{ij}}{\sqrt{d_k}}. \quad (5.5)$$

Here,  $\mathbf{B}_{ij}$  is a scalar that adjusts the attention matrix based on the query and key indices  $i, j$ . Assuming a set of keypoints  $\mathbf{P} \in \mathbb{R}^{N_L \times 2}$  is available for each  $\mathbf{x}$ , our goal is to make  $\mathbf{B}_{ij}$  dependent on  $\mathbf{P}$ . For face recognition,  $\mathbf{P}$  is the five facial landmarks (two eyes, nose, mouth tips). For gait recognition,  $\mathbf{P}$  is 17 points from the joint locations of skeleton predictions. For the MNIST toy example,  $\mathbf{P}$  is five keypoints from the four corners and the center of the

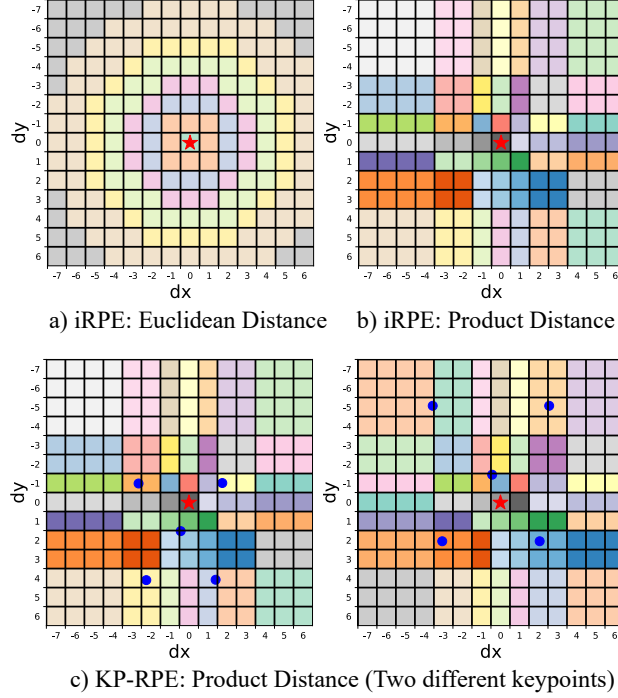


Figure 5.3 Depiction of key-query combinations in an image, given a query location  $i = (7, 7)$  ( $\star$ ). Distinct colors represent varying attention offset values in RPE based on the distance between  $i$  and  $j$ . We are showing  $\mathbf{B}_{i=(7,7),j}$  for all  $j \in (14 \times 14)$ . a) The distance function is a quantized Euclidean distance. b) Product distance proposed in iRPE accounts for direction. c) We adopt b) and allow  $\mathbf{B}_{i,j}$  to vary based on keypoint locations ( $\bullet$ ).

minimum cover box of a foreground image. As such  $\mathbf{P}$  can be defined for objects with shared topology.

The novelty of KP-RPE lies in the design of  $\mathbf{B}_{ij}$ . Since

$$\mathbf{B}_{ij} = \mathbf{W}[d(i, j)] \in \mathbb{R}^1, \quad (5.6)$$

comprises of a learnable table  $\mathbf{W}$  and a distance function  $d(i, j)$ , we can make  $\mathbf{W}$  or  $d(i, j)$  depend on the keypoints. At a first glance,  $d(i, j, \mathbf{P})$ , conditioning the distance on  $\mathbf{P}$  seems plausible. However, we find that it leads to inefficiencies, as distance caching, which is precomputing  $d(i, j)$  for a given input size, is only feasible when  $d(i, j)$  is independent of the input. Therefore, we propose an alternative where the bias matrix  $\mathbf{W}$ , is a function of  $\mathbf{P}$ :

$$\mathbf{B}_{ij} = f(\mathbf{P})[d(i, j)]. \quad (5.7)$$

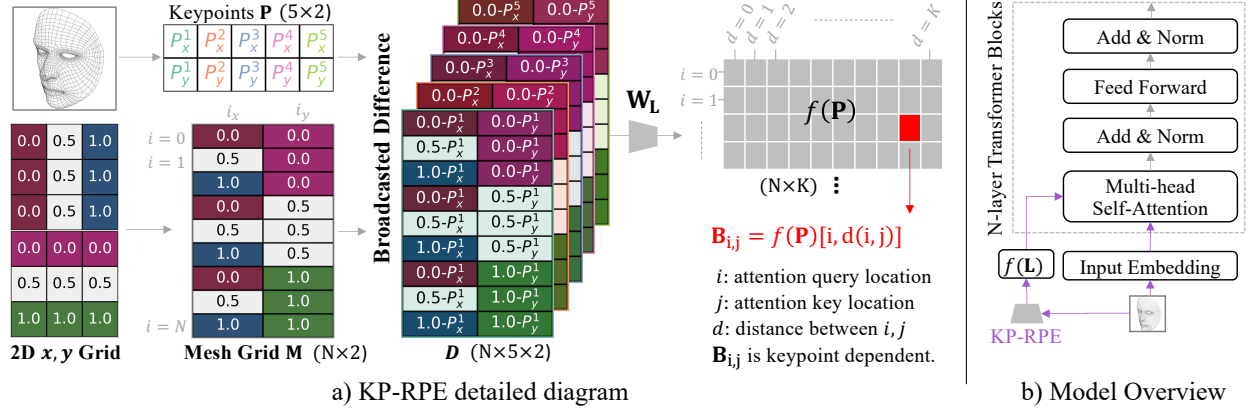


Figure 5.4 a) Illustration of KP-RPE. First a mesh grid  $\mathbf{M}$  and an image-specific keypoints  $\mathbf{P}$  are generated. Then the broadcasted difference  $\mathbf{D}$  is calculated, and we linearly map  $\mathbf{D}$  to  $f(\mathbf{P})$ . Finally for a given  $i, j$ , we can find the  $\mathbf{B}_{ij} = f(\mathbf{P})[i, d(i, j)]$ , which is used to adjust the attention map in self-attention. b) Backbone contains multiple transformer blocks followed by an MLP for classification. KP-RPE is used where multi-head attention modules exist. KP-RPE is efficient as  $f(\mathbf{P})$  is computed once.

We propose three variants of  $f(\mathbf{P})$  building up from the simplest solution.

**Absolute  $f(\mathbf{P})$**  Let  $\mathbf{P} \in \mathbb{R}^{N_L \times 2}$  be the normalized keypoints between 0 and 1. First, the simplest way to model the indexing table is to linearly map  $\mathbf{P}$  to the desired shape.  $f(\mathbf{P}) = \mathbf{P}'\mathbf{W}_L$  where  $\mathbf{P}' \in \mathbb{R}^{1 \times (2N_L)}$  is reshaped keypoints  $\mathbf{P}$  and  $\mathbf{W}_L \in \mathbb{R}^{(2N_L) \times K}$  is a learnable matrix.  $K$  is the maximum distance value in  $d(i, j)$ . For each distance between  $i$  and  $j$ , we learn a keypoint adaptive offset value. However, this  $f(\mathbf{P})$  only works with the absolute position information of  $\mathbf{P}$  and the relative distance between  $i$  and  $j$ . It is missing the relative distance between  $\mathbf{P}$  and  $(i, j)$ .

**Relative  $f(\mathbf{P})$**  To improve,  $f(\mathbf{P})$  can be adjusted to work with the position of keys and queries *relative* to the keypoints. In other words, so that the query-key relationship in  $\mathbf{B}_{ij}$  depends on the query-landmark relationship. To achieve this, we generate a mesh grid  $\mathbf{M} \in \mathbb{R}^{N \times 2}$  of patch locations containing all possible combinations of  $i_x$  and  $i_y$ .  $N$  represents the number of patches. We then compute the element-wise difference between the normalized

grid and keypoints  $\mathbf{P}$  to obtain a grid of  $i, j$  relative to the keypoints:

$$\mathbf{D} = \text{Expand}(\mathbf{M}, \text{dim}=1) - \text{Expand}(\mathbf{P}, \text{dim}=0), \quad (5.8)$$

where  $\mathbf{D}$  is the broadcasted tensor difference of shape  $\mathbb{R}^{N \times N_L \times 2}$ . Finally, we reshape  $\mathbf{D}$  and linearly project it with  $\mathbf{W}_L$ . Specifically,

$$\mathbf{D}' = \text{Reshape}(\mathbf{D}) \in \mathbb{R}^{N \times (2N_L)} \quad (5.9)$$

$$f(\mathbf{P}) = \mathbf{D}'\mathbf{W}_L \in \mathbb{R}^{N \times K} \quad (5.10)$$

$$\mathbf{B}_{ij} = f(\mathbf{P})[i, d(i, j)] \in \mathbb{R}^1. \quad (5.11)$$

In other words, the offset value  $\mathbf{B}_{ij}$  is determined with respect to the positions of the keypoints and is unique for each query location. This approach allows for more expressive control of the query-key relationships with the keypoint locations. An illustration of this is shown in Fig. 5.4.

**Multihead Relative  $f(\mathbf{P})$**  Lastly, we can further enhance our method by tailoring the query-keypoint relationship for each head in the attention mechanism. When there are  $H$  heads, we simply expand the dimension of  $\mathbf{W}_L$  to  $\mathbf{W}_L \in \mathbb{R}^{(2N_L) \times HK}$ . By reshaping  $f(\mathbf{P})$ , we obtain  $f(\mathbf{P})^h$  for each head. Furthermore, considering the multiple self-attentions in ViT which entails multiple RPEs, we can individualize  $f(\mathbf{P})$  for each self-attention by additionally increasing the dimension of  $\mathbf{W}_L$  to  $\mathbf{W}_L \in \mathbb{R}^{(2N_L) \times N_d HK}$ , where  $N_d$  represents the transformer’s depth. Since  $f(\mathbf{P})$  is computed only once per forward pass, this modification introduces negligible computational overhead compared to other operations. In Sec. 5.4.2, we evaluate and compare the various KP-RPE versions (basic, relative keypoint, multiple relative keypoint), demonstrating the superior performance of the multiple relative keypoint approaches.

## 5.4 Face Recognition Experiments

### 5.4.1 Datasets and Implementation Details

To validate the efficacy of KP-RPE, we train our model using aligned face training data and evaluate on three distinct types of datasets: 1) aligned face data, 2) intentionally

Method	Low Quality Aligned Dataset				High Quality Aligned Dataset		High Quality Unaligned Dataset	
	TinyFace [46]		IJB-S [112]		CFPFP [202]	IJB-C [169]	CFPFP [202]	IJB-C [169]
	Rank-1	Rank-5	Rank-1	Rank-5	Verification	TAR@0.01%	Verification	TAR@0.01%
ViT	68.24	72.96	59.60	68.31	96.11	92.22	72.81	21.62
ViT + iRPE	69.05	73.10	62.49	70.50	<b>97.01</b>	92.72	77.91	34.73
ViT+KP-RPE	<b>69.88</b>	<b>74.25</b>	<b>63.44</b>	<b>72.04</b>	96.60	<b>94.20</b>	<b>93.56</b>	<b>91.85</b>

Table 5.1 Ablation of RPE on ViT-small. Aligned is the standard protocol with raw face images (detector bounding box) aligned by RetinaFace [54] and resized to  $112 \times 112$ .

Unaligend takes the raw face images and simply resizes it to  $112 \times 112$ . Aligned setting always shows better performances and Unaligned is for simulating alignment failure. Low Quality Aligned dataset may have alignment failures.

Method	Low Quality Aligned Dataset				High Quality Aligned Dataset		High Quality Unaligned Dataset	
	TinyFace [46]		IJB-S [112]		CFPFP [202]	IJB-C [169]	CFPFP [202]	IJB-C [169]
	Rank-1	Rank-5	Rank-1	Rank-5	Verification	TAR@0.01%	Verification	TAR@0.01%
KP-RPE Absolute $f(\mathbf{P})$	68.11	72.42	9.97	69.13	96.51	90.96	68.09	14.91
KP-RPE Relative $f(\mathbf{P})$	69.42	73.71	62.51	70.77	<b>96.74</b>	<b>94.28</b>	89.70	85.22
KP-RPE MultiHead $f(\mathbf{P})$	<b>69.88</b>	<b>74.25</b>	<b>63.44</b>	<b>72.04</b>	96.60	94.20	<b>93.56</b>	<b>91.85</b>

Table 5.2 Ablation of KP-RPE with three different formulations of keypoint dependent RPE tables  $f(\mathbf{P})$ . The sharp increase in Unaligned setting shows the robustness to unseen affine transform manifests with Relative  $f(\mathbf{P})$ . Multihead  $f(\mathbf{P})$  further improves the performance.

unaligned face data, and 3) low-quality face data containing misaligned images. For the evaluation, aligned face datasets include CFPFP [202], AgeDB [174], and IJB-C [169]. For unaligned face data, we intentionally use the raw CFPFP [202] and IJB-C [169] datasets without aligning them. Raw images, as provided by their respective creators, are equivalent to images cropped based on face detection bounding boxes. Lastly, we assess the model’s robustness on low-quality face datasets, specifically TinyFace [46] and IJB-S [112], which are prone to alignment failures. This comprehensive setup enables us to examine the effectiveness of our proposed method across diverse data conditions.

The training datasets MS1MV2 [55] MS1MV3 [57] and WebFace4M [300] are released as aligned and resized to  $112 \times 112$  by RetinaFace [54] whose backbone is ResNet50 model trained on WiderFace [269]. For keypoint detection in KP-RPE, we also use RetinaFace [54] but with lighter backbone MobileNetV2 for faster inference. Given the sensitivity of ViTs to hyperparameters, we report the exact settings for learning rate, weight decay, and other parameters in the later section. For ablation dataset, we take the MS1MV2 subset dataset as

used in [122].

Following the training conventions of [122, 230], we adopt RandAug [49], repeated augmentation [94], random resized crop, and blurring. We utilize the AdaFace [122] loss function to train all models. For ablation, we employ ViT-small, while for SoTA comparisons, we use ViT-base models. The AdamW [164] optimizer and Cosine Learning Rate scheduler [163, 254] are used. In WebFace4M trained models, we adopt PartialFC [10, 11] to reduce the classifier’s dimension.

#### 5.4.2 Ablation Analysis

Row 1 in Tab. 5.1 shows results on the baseline ViT. Row 2 and 3 show results on the baseline ViT with iRPE and our proposed KP-RPE. KP-RPE demonstrates a substantial performance improvement on unaligned and low-quality datasets, without compromising performance on aligned datasets. Last row highlights the difference between ViT and ViT+KP-RPE. Also, Fig. 5.5 shows the sensitivity to the affine transformation, *i.e.*, how the performance changes when one interpolates the affine transformation from the face detection images to the aligned images in CFPFP dataset.

Tab. 5.2 further investigates the effect of modifications to KP-RPE. By making KP-RPE dependent on the difference between the query and keypoints (row 2), we observe a significant improvement in unaligned dataset performance. Also, by allowing unique mapping for each head and module in ViT (row 3), we achieve a further improvement. In other words, more expressive KP-RPE is beneficial for learning complex RPE that depends on the keypoints of an image. Overall, the ablation study highlights the necessity of each component in KP-RPE and the effectiveness of KP-RPE in enhancing the robustness of face recognition models, particularly with unaligned and low-quality datasets.

#### 5.4.3 Computation Analysis

In this section, we analyze the computational efficiency of our proposed KP-RPE in terms of FLOPs, throughput, and the number of parameters. Tab. 5.3 shows that KP-RPE is highly efficient, with only a small increase in the computational cost (FLOPs) compared to

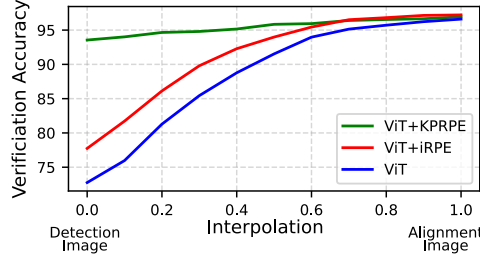


Figure 5.5 Plot of Verification Accuracy in CFPFP [202]. On the X-axis, we interpolate the affine transformation from raw data (Detection Image) to canonical alignment (Alignment Image). Note KP-RPE is robust to affine transformations, while all models have been trained on the aligned image dataset.

	GFLOP	$\Delta$ in GFLOP	Eval Throughput	Train Throughput	% $\Delta$ in Train Throughput	# Param
IResNet50	12.62	-	1432.72 img/s	337.93 img/s	-	43.59M
ViT Small	17.42	①	1303.15 imgs/s	333.17 img/s	①	95.95M
ViT Small + iRPE	18.13	①+0.71	832.12 imgs/s	186.55 img/s	①-44.01%	96.07M
ViT Small + <b>KP-RPE</b>	17.44	①+0.02	1145.90 imgs/s	302.70 img/s	①-9.15%	96.00M
ViT Small + <b>KP-RPE (+ Ldmk)</b>	17.58	①+0.16	1085.22 imgs/s	302.70 img/s	①-9.15%	96.49M
IResNet101	24.19	-	773.12 imgs/s	189.74 img/s	-	65.15M
ViT Base	24.83	②	644.10 imgs/s	162.94 img/s	②	114.87M
ViT Base + iRPE	26.25	②+1.42	337.32 imgs/s	79.40 img/s	②-51.27%	114.98M
ViT Base + <b>KP-RPE</b>	24.90	②+0.07	502.57 imgs/s	136.15 img/s	②-16.44%	115.08M
ViT Base + <b>KP-RPE (+ Ldmk)</b>	25.04	②+0.21	489.37 imgs/s	136.15 img/s	②-16.44%	115.56M

Table 5.3 Computation resource comparison. GFLOP refers to Giga Floating Operating per Second. We measure it as [193]. Throughput refers to the number of images processed per second during the train/eval iteration.

the backbone: 0.02 GFLOP increase for ViT Small and 0.07 GFLOP increase for ViT Base (ViT vs ViT+KP-RPE). Notably, KP-RPE is considerably more efficient than iRPE, which incurs an increase of 0.71 GFLOP for ViT Small and 1.42 GFLOP for ViT Base.

Considering training throughput, which factors in computation time during training (with backpropagation), KP-RPE’s efficiency is more pronounced. It only reduces throughput by 9.15% for ViT Small and 16.44% for ViT Base, as opposed to iRPE’s larger decrease. Also, we show the GFLOP and throughput with the landmark detection time included. Landmark detection time is negligible compared to the total feature extraction time.

Also, our method introduces a negligible increase in the number of parameters: just 0.05M for ViT Small and 0.21M for ViT Base. Hence, incorporating KP-RPE into the model

Method	Backbone	Train Data	Low Quality Dataset				High Quality Dataset			
			TinyFace [46]		IJB-S [112]		AgeDB [174]	CFPFP [202]	IJB-C [169]	TAR@FAR=0.01%
			Rank-1	Rank-5	Rank-1	Rank-5	Verification Accuracy	Accuracy		
PFE [208]	CNN64	MS1MV2 [55]	-	-	50.16	58.33	-	-	-	-
ArcFace [55]	ResNet101	MS1MV2 [55]	-	-	57.35	64.42	98.28	98.27	96.03	
URL [210]	ResNet101	MS1MV2 [55]	63.89	68.67	59.79	65.78	-	98.64	96.60	
CurricularFace [102]	ResNet101	MS1MV2 [55]	63.68	67.65	62.43	68.68	<b>98.32</b>	98.37	96.10	
AdaFace [55]	ResNet101	MS1MV2 [55]	68.21	71.54	65.26	70.53	98.05	98.49	96.89	
AdaFace [55]	ResNet101	MS1MV3 [57]	67.81	70.98	67.12	72.67	98.17	99.03	97.09	
AdaFace [122]	ViT	MS1MV3 [57]	72.05	74.84	65.95	71.64	97.87	99.06	97.10	
AdaFace [122]	<b>ViT+KP-RPE</b>	MS1MV3 [57]	<b>73.50</b>	<b>76.39</b>	<b>67.62</b>	<b>73.25</b>	97.98	<b>99.11</b>	<b>97.16</b>	
ArcFace [55]	ResNet101	WebFace4M [300]	71.11	74.38	69.26	74.31	<b>97.93</b>	99.06	96.63	
AdaFace [122]	ResNet101	WebFace4M [300]	72.02	74.52	70.42	75.29	97.90	<b>99.17</b>	<b>97.39</b>	
AdaFace [122]	ViT	WebFace4M [300]	74.81	77.58	71.90	77.09	97.48	98.94	97.14	
AdaFace [122]	ViT+IRPE	WebFace4M [300]	74.92	77.98	71.93	77.14	97.15	99.01	97.01	
AdaFace [122]	<b>ViT+KP-RPE</b>	WebFace4M [300]	<b>75.80</b>	<b>78.49</b>	<b>72.78</b>	<b>78.20</b>	97.67	99.01	97.13	
AdaFace [122]	ResNet101	WebFace12M [300]	72.42	74.81	71.46	77.04	98.00	99.24	97.66	
AdaFace [122]	<b>ViT+KP-RPE</b>	WebFace12M [300]	<b>76.18</b>	<b>78.97</b>	<b>72.94</b>	<b>77.46</b>	<b>98.07</b>	<b>99.30</b>	<b>97.82</b>	

Table 5.4 SoTA comparison on low-quality and high-quality datasets. ViT models are ViT-Base sized.

achieves enhanced performance without a substantial rise in computational cost or model complexity.

#### 5.4.4 Comparison with SoTA Methods

In this section, we position ViT+KP-RPE, against SoTA face recognition methodologies with large-scale datasets and large models. We undertake a comprehensive evaluation, covering both high-quality and low-quality image datasets. The results, as shown in Tab.5.4, underscore the strengths of KP-RPE. Notably, the inclusion of KP-RPE does not impair the performance on high-quality datasets, a testament to its applicability to both low and high-quality datasets.

This becomes particularly compelling when we observe the performance on low-quality datasets. Consistent with the findings of our ablation study, the introduction of KP-RPE leads to an appreciable improvement in these challenging scenarios. This supports our thesis that face recognition models with robust alignment capabilities can indeed enhance performance on low-quality datasets. In summary, our model with KP-RPE not only maintains competitive performance on high-quality datasets but also brings significant improvements on low-quality ones, marking it a valuable contribution to the field of face recognition.

#### 5.4.5 Note on the Landmark Predictor

KP-RPE in all experiments uses our own MobileNet [199] based RetinaFace [54] to predict landmarks for KP-RPE. We train MobileNet version for computation efficiency. However, the original landmark predictor used for aligning the test datasets is ResNet50-RetinaFace [54]. We also report the KP-RPE performance with the officially released ResNet50-RetinaFace. We report this to compare KP-RPE on the same ground with other models by using the same landmark used to pre-align the testset. The face recognition performance of KP-RPE+Official is similar to KP-RPE+Ours (75.86 vs 75.80 in TinyFace Rank1). Our MobileNet-RetinaFace is improved to perform similarly to ResNet50 in landmark prediction by applying additional tricks while training. Therefore, the face recognition performances are also similar. Unlike vanilla RetinaFace on face alignment, ours is fully differentiable during inference and named Differentiable Face Aligner.

#### 5.4.6 Scalability on Larger Training Datasets

We train the ViT+KP-RPE model on a larger WebFace12M [300] dataset to demonstrate the potential of KP-RPE in its scalability and applicability in real-world, data-rich scenarios. Tab.5.4’s last row shows that the performance continues to increase with WebFace12M dataset.

**Discussion** Why are noisy keypoints more useful in KP-RPE than in simple alignment? The short answer is that not all predicted points are noisy in an image while alignment as a result of one or more noisy point impacts all pixels.

### 5.5 Gait Recognition Experiments

KP-RPE is a generic method that can generalize beyond face recognition to any task with keypoints. We apply KP-RPE to gait recognition using body joints as the keypoints.

**Dataset.** We train and evaluate on Gait3D [292], an in-the-wild gait video dataset. In our experiments, we use silhouettes and 2D keypoints preprocessed and released by the authors directly. Following SMPLGait [273, 292], we use rank- $n$  accuracy ( $n = 1, 5, 10$ ), mean Average Precision (mAP), and mean Inverse Negative Penalty (mINP) for evaluation.

Model	Rank-1	Rank-5	mAP	mINP
GaitSet [40]	36.7	58.3	30.01	17.30
MTSGait [291]	48.7	67.1	37.63	21.92
DANet [167]	48.0	69.7	—	—
GaitGCI [62]	50.3	68.5	39.5	24.3
GaitBase [67]	64.6	81.5	55.31	31.63
HSTL [242]	61.3	76.3	55.48	34.77
DyGait [243]	66.3	80.8	56.40	<b>37.30</b>
SwinGait-2D [66]	67.1	83.7	58.76	34.36
+ KP-RPE	<b>68.2</b>	<b>84.4</b>	<b>60.81</b>	36.19

Table 5.5 KP-RPE performance on Gait3D [292] compared with the baseline. KP-RPE boosts all metrics by a large margin.

**Implementation Details** We implement SwinGait-2D [66] as the baseline in our experiments. SwinGait-2D is chosen over SwinGait-3D [66] because we focus on exploiting the geometric information in gait recognition. SwinTransformer [160] uses vanilla relative positional encoding for each windowed self-attention. To incorporate KP-RPE into the SwinTransformer, we modify the 2D grid  $\mathbf{M}$  to be the size of the window as opposed to the image size. Following the default configuration of [292], we use an AdamW [164] optimizer with a learning rate  $3 \times 10^{-4}$  and weight decay  $2 \times 10^{-2}$ , accompanied by an SGDR [163] scheduler. We train our models for 60,000 iterations, sampling 32 subjects and 4 sequences per subject in a batch.

**Results and Analyses** In Tab. 5.5, we compare to SoTA approaches, including SwinGait-2D [66], with and without KP-RPE. We can see that the KP-RPE shows a significant improvement over SwinGait-2D, with 1.1 % and 0.7 % improvement on rank-1 and -5 accuracies, respectively. mAP has improved by 2.05 % and mINP by 1.23 % of the baseline) compared to SwinGait-2D. We believe that a great portion of the improvement comes from KP-RPE exploiting the gait information contained in 2D skeletons. Gait skeletons contain identity-related information, such as body shape and walking posture. This demonstrates that KP-RPE is both effective and generalizable to gait recognition.

## 5.6 Training Details

Training code will be released for reproducibility. Our experiments were conducted using the PyTorch deep learning framework. Detailed information pertaining to the training

parameters, configurations, and specifics can be referred to in Tab. 5.6. We employed the Vision Transformer (ViT) model architectures as implemented in the InsightFace GitHub repository, ensuring a well-established and tested model foundation. When measuring the throughput of our KeyPoint Relative Position Embedding (KPRPE), we utilized an NVIDIA RTX3090 GPU.

	<b>Ablation Experiments</b>	<b>Large Scale Experiments</b>
Backbone	ViT Small	ViT Large
LR	0.001	0.0001
Batch Size	512	1024
Epoch	34	36
Momentum		0.9
Weight Decay		0.05
Scheduler		Cosine
Optimizer		AdamW
Warmup		3
AdaFace Loss Margin		0.4
AdaFace Loss $h$		0.333
Augmentation	Flip, Brightness, Contrast, Scaling, Translation, RandAug [49](magnitude:14/31), Blur, Cutout, Rotation (20°)	
PartialFC	None	sampling rate 0.6
RepeatedAug Prob	0.5	0.1

Table 5.6 Details for training face recognition models with or without KPRPE.

## 5.7 Supplementary Performance Analysis

### 5.7.1 Performance Across Various Loss Functions

In our extensive evaluation, we have employed three popular loss functions: AdaFace [122], CosFace [240], and ArcFace [55], to train the Vision Transformer (ViT) in combination with our proposed KeyPoint Relative Position Embedding (KPRPE). As demonstrated by the results in Tab. 5.7 rows 3-6, our method exhibits consistent performance improvements on lower quality datasets across all three loss functions when compared to the standalone ViT. This signifies the versatility of KPRPE in synergizing with a variety of loss functions to enhance the robustness of face recognition models to less-than-optimal image quality.

Method	Backbone	Train Data	Low Quality Dataset				High Quality Dataset		
			TinyFace [46]		IJB-S [112]		AgeDB	CFPFP	IJB-C
			Rank-1	Rank-5	Rank-1	Rank-5	Verification	Accuracy	0.01%
AdaFace [122]	<b>ViT</b>	WebFace4M [300]	74.81	77.58	71.90	77.09	97.48	98.94	97.14
AdaFace [122]	<b>ViT+KPRPE</b>	WebFace4M [300]	<b>75.80</b>	78.49	72.78	78.20	<b>97.67</b>	99.01	97.13
ArcFace [55]	<b>ViT+KPRPE</b>	WebFace4M [300]	75.62	<b>78.57</b>	<b>73.04</b>	<b>78.62</b>	97.57	<b>99.06</b>	<b>97.21</b>
CosFace [240]	<b>ViT+KPRPE</b>	WebFace4M [300]	75.48	78.30	72.22	77.67	97.45	98.94	96.98

Table 5.7 SoTA comparison on low-quality and high-quality datasets. IJB-C [253] reports TAR@FAR=0.01%.

### 5.7.2 Performance with Different Number of Keypoints

We include the impact of the number of keypoints in KP-RPE. We initiated the analysis with 5 keypoints, the maximum available in RetinaFace. And gradually reduce the number of points.

Number of Keypoints	TinyFace Rank1	TinyFace 5	AgeDB	CFPFP
5	<b>69.88</b>	<b>74.25</b>	<b>95.92</b>	96.60
4	69.58	73.63	95.65	96.57
3	69.66	73.95	95.77	<b>96.80</b>
2	69.26	73.42	95.73	95.97
No Keypoints (Vanilla ViT)	68.24	72.96	95.57	96.11

Table 5.8 Performance by changing the number of keypoints.

For datasets characterized by lower image quality like TinyFace, the performance diminishes as the number of keypoints reduce. But it does not diminish compared to not using the keypoints. It could be that the information about the scale and rotation of an image could still be captured by few points as 2 or 3. Interestingly, in high-resolution datasets, the trend is absent and the performance remains relatively consistent regardless of the number of keypoints used. More keypoints can be adopted with other landmark detectors but they are not trained with low quality images in WiderFace as the dataset only provides 5 points.

### 5.7.3 Sensitivity to Landmark Error in KPRPE

To test the sensitivity of KPRPE to the landmark prediction error, we take the prediction of the landmark predictor and perturb it by the following equation,

$$\mathbf{L}_{pert} = \mathbf{L} + \alpha \mathbf{L}. \quad (5.12)$$

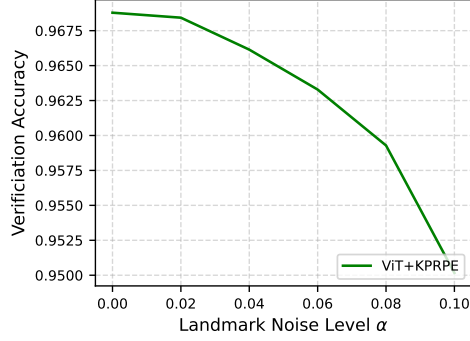


Figure 5.6 Verification accuracy measured in CFPFP dataset with added noise in landmark predictions.

$\alpha$  is a parameter that changes the level of noise in the prediction. We change  $\alpha$  from 0 to 0.1 after noting that 0.1 makes the NME score to be 0.12 which is far worse than the NME score of 0.05 in WiderFace which is a harder dataset. Therefore,  $\alpha = 0.1$  is an extreme scenario where all of the inputs have failed to the level which exceeds the average level of failure in WiderFace by two times.

Note that as we add noise into the landmark prediction, the performance goes down, signaling that KPRPE is dependent on the landmark prediction. However, the amount of performance drop within the range of realistic noise level is not too much (about 1.5%). Fig 5.11 shows the experiment setting in a diagram.

#### 5.7.4 Why are noisy keypoints more useful in KP-RPE than in simple alignment?

The short answer is that not all predicted points are noisy in an image while alignment as a result of one or more noisy point impacts all pixels. For a more concrete example, in Fig. 5.7, we have taken images from WiderFace which contains human-annotated ground truth keypoints and compared them with RetinaFace prediction. Fig. 5.7 (a) shows a well aligned scenario. (b) and (c) show that when one or two landmarks (red color) deviate from the ground truth (GT), the resulting alignment changes dramatically. For KP-RPE, this is a less severe problem because individual landmarks affect the RPE independently in the landmark space (0-1). On the other hand, when affine transformation is regressed to align the image to a canonical space, individual landmark error becomes correlated and amplified.

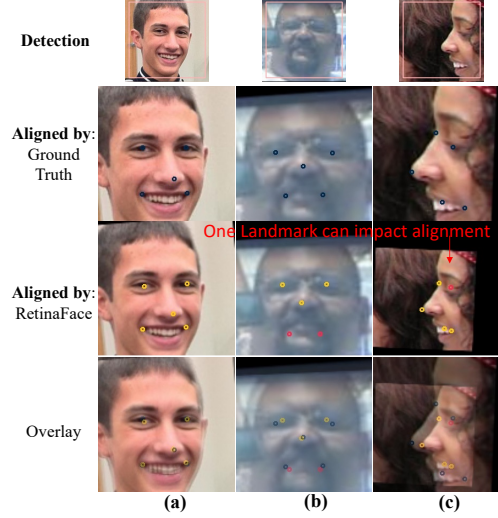


Figure 5.7 Keypoints in Aligned images. Blue: ground truth keypoints. Yellow/Red: RetinaFace keypoints with less/more than 5% error from GT. Overlay of (b,c) shows how small deviation from one or two points can lead to significant scale, translation change.

## 5.8 Training Landmark Detector (MobileNet-RetinaFace)

RetinaFace [54], a single-stage face detector, is built upon Feature Pyramid Network [144] (FPN) and Single Shot MultiBox Detector [153]. It is originally designed for detecting multiple faces using anchor boxes in each location in an image. However, in our case, we assume the presence of one face, and we leverage this constraint to improve the landmark detection performance and efficiency of the model. This assumption is valid if a face detector crops out a face, which is a standard practice in face recognition. With this assumption, we can modify the RetinaFace to predict more accurate landmarks when the input image is cropped. We adopt few training techniques and a faster aggregation technique and name it Differentiable Face Aligner (DFA). The name suggests that with the modifications we propose, the face alignment network is differentiable (unlike RetinaFace because of NMS and CPU based cropping), making it potentially useful for other applications in computer vision.

**Training Data Adaptation** We adapt the training data WiderFace [269] for our Differentiable Face Aligner (DFA) by cropping out facial images using the ground truth bounding boxes. And we resize the input to be 160x160. This change in data size and distribution

allows the model to specialize in localizing landmarks for single faces, ultimately improving its performance.

**Aggregation Network** The motivation for the aggregation network is to eliminate the Non-Maximum Suppression (NMS) and output a single landmark prediction from multiple anchor boxes. We design a network that takes in the output of FPN and aggregates it to a single prediction. The architecture of the aggregation network consists of MixerMLP [229]. Specifically, let  $\mathbf{X}$  be an image, and let  $\mathbf{F}_{bbox}$ ,  $\mathbf{F}_{score}$  and  $\mathbf{F}_{ldmk}$  be the set of the output of FPN followed by the corresponding multitask head (bounding box, face score and landmark prediction) for each anchor box. For example, when an image is sized  $160 \times 160$ , there are 1050 anchor boxes. Based on these outputs, we predict the weights for fusing the outputs. Specifically,

$$\mathbf{O} = \text{Concat}(\mathbf{F}_{bbox}, \mathbf{F}_{score}, \mathbf{F}_{ldmk}) \in \mathbb{R}^{1050 \times (C_{bbox} + C_{score} + C_{ldmk})} = \mathbb{R}^{1050 \times (4+1+10)}, \quad (5.13)$$

$$\mathbf{w} = \text{Softmax}(\text{MixerMLP}(\mathbf{O})) \in \mathbb{R}^{1050}, \quad (5.14)$$

$$\mathbf{L} = \mathbf{w}^T \mathbf{F}_{ldmk}. \quad (5.15)$$

The final output  $\mathbf{L}$  is the weighted average of the landmarks in all anchor boxes. The aggregation network is trained end to end with the rest of the detection model with the smooth L2 Loss [194] between  $\mathbf{L}$  and the ground truth landmark  $\mathbf{L}^{GT}$ .

By incorporating these modifications, we show in Sec. 5.8.1 that our DFA achieves superior landmark detection performance compared to the RetinaFace while using a more efficient backbone architecture.

**Training Details** For the training of our Differentiable Face Aligner (DFA), we incorporated specific training settings to optimize the performance. We used an input image size of 160 pixels, with a batch size of 320. Training was conducted for 750 epochs, ensuring that the model had adequate exposure to learn and generalize from the dataset. Training was

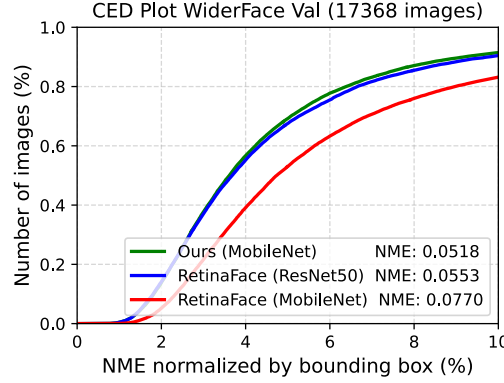


Figure 5.8 Cumulative Error Distribution curve and the corresponding NME for models evaluated on WiderFace [269] validation set.

performed using the WiderFace training dataset, with images cropped using the ground truth bounding boxes and a padding of 0.1.

### 5.8.1 Landmark Detection Performance

In this section, we evaluate the performance of our proposed Differentiable Face Aligner (DFA) in terms of landmark detection. We use the Normalized Mean Error (NME) as the metric and evaluate on WiderFace validation set [269] as in RetinaFace [54].

Fig. 5.8 shows an improvement in NME when using DFA compared to the baseline RetinaFace. The RetinaFace with MobileNet backbone achieves an NME of 0.077, while the one with ResNet50 achieves 0.0553. In contrast, our DFA achieves 0.0518, demonstrating its superiority in landmark detection.

Moreover, the DFA model benefits from the introduction of the aggregation network, which eliminates the need for the NMS stage. The improvement in NME due to the aggregation network is from 0.0527 to 0.0518. This not only simplifies the overall pipeline but also contributes to the enhanced performance of the DFA model in the landmark detection. With a straightforward modification in the training data and an aggregation stage that assumes a single-face image, a lightweight backbone with better performance can be trained.

## 5.9 IJB-S Evaluation Method

IJB-S [112] is a video-based dataset that defines probe and gallery templates according to its predefined video clip of arbitrary length. This naturally implies one must perform feature aggregation (fusion) when frame-level features are predicted. Since the backbone predicts a unit-norm feature vector for one image, the simplest method would be to average all the features within the template. The most popular method is to utilize norm-weighted average, where the features are averaged before normalization [122]. This only works if the norm is a good proxy for the prediction quality. However, in certain cases, depending on various factors such as dataset, learning rate, backbone, optimizer, etc., that go into the training of a model, this may not be the case. Also, in our experience, ViT+KPRPE was not the case.

Therefore, we propose a proxy that could easily replace the norm with another quantity that can be found within a model. Since DFA predicts the landmarks  $\mathbf{L}$  and a face score  $\mathbf{F}_{score}$ , we derive a fusion score using those quantities. First, let us review the conventional norm-weighted feature fusion equation for a set of  $N$  number of feature vectors  $\{f_i\}_N$  where  $f_i = \|f_i\|_2 \cdot \bar{f}_i$  decomposes  $f_i$  into the norm and the unit length feature.

$$f_{\text{norm weighted}} = \frac{\sum_{i=1}^N \|f_i\|_2 \cdot \bar{f}_i}{N}. \quad (5.16)$$

In the equation above,  $f_i$  represents the  $i$ -th frame-level feature, and  $N$  is the total number of frames. Now, for KPRPE, we propose a new feature fusion method, incorporating the face score and the Euclidean distance between predicted landmarks  $\mathbf{L}$  and the canonical landmark  $\hat{\mathbf{L}}$ , which is a known set of landmarks that the training images are aligned to. This distance score,  $d$ , is computed as:

$$d_i = \frac{h - \min(\|\mathbf{L}_i - \hat{\mathbf{L}}\|_2, h)}{h}, \quad (5.17)$$

where  $h = 0.2$  is a fixed constant that allows the score to be bounded between 0 and 1. The face score  $\mathbf{F}_{score}^i$  represents the quality of the image, and  $d_i$  assigns more weight to well-aligned images. Proposed feature fusion equation, hence, becomes:

$$f_{\text{KPRPE}} = \frac{\sum_{i=1}^N (d_i \cdot \mathbf{F}_{score}^i) \cdot \bar{f}_i}{N}. \quad (5.18)$$

This method allows for the aggregation of features even when the feature norm does not serve as a good proxy for the quality of an image. In computing IJB-S result for ViT+KPRPE, we use this fusion method.

However, for a fair comparison in IJB-S, it is important to apply this fusion method to previous methods. Therefore, we include the breakdown of with and without landmark score based fusion. For single image based datasets such as TinyFace, AgeDB or CFPFP, feature fusion is not needed.

Training Data: MS1MV3	Feature Fusion Method	IJB-S Rank1	IJB-S Rank5	TinyFace Rank1
ViT Base+IRPE	Average	62.49	70.50	69.05
ViT Base+IRPE	Landmark based	63.81	71.30	69.05
ViT Base+KPRPE	Average	63.44	72.04	69.88
ViT Base+KPRPE	Landmark based	64.68	72.33	69.88
Training Data: WebFace4M	Feature Fusion Method	IJB-S Rank1	IJB-S Rank5	TinyFace Rank1
ViT Large+IRPE	Average	71.32	76.22	74.92
ViT Large+IRPE	Landmark based	71.93	77.14	74.92
ViT Large+KPRPE	Average	65.95	71.64	75.80
ViT Large+KPRPE	Landmark based	72.78	78.20	75.80

Table 5.9 Breakdown of with and without fusion method in various backbones and datasets.

The performance of ViT+KP-RPE consistently surpasses ViT+iRPE, both in scenarios using Averaging or Landmark-based fusion. This affirms the efficacy of KP-RPE in enhancing performance, even in single image contexts like TinyFace. Importantly, while the keypoint detection step is integral to KP-RPE, it isn’t incorporated within iRPE, making a direct comparison based on this score less straightforward for iRPE.

Interestingly, average fusion does not synergize well with ViT+KP-RPE. Contrary to typical observations where feature magnitude positively correlates with image quality [122], with ViT+KP-RPE, a higher feature magnitude actually suggests reduced image quality. It remains unclear why this inverse relation emerges in our model. Through empirical observations, the relationship between feature magnitude and image quality appears contingent on the chosen training dataset and model architecture. For instance, models based on the ResNet architecture consistently exhibit a positive correlation between feature magnitude and image quality.

### 5.10 Alignment Visualizations

TinyFace [46] and IJBS [112], which are prone to alignment failures. In Fig 5.9 we show some success and failure cases in alignment. These images are taken from the released aligned dataset itself.



Figure 5.9 Actual examples of aligned and mis-aligned images from TinyFace [46] (row1,3) and IJB-S [112] (row2,4) datasets. These are shown as processed and used by [122]. Lines are placed on the eyes for a visual guide for an alignment.

### 5.11 Comparison with SoTA Off-the-Shelf Landmark Detector

We evaluate the off-the-shelf landmark detector SLPT [261] (CVPR2022), which delivers strong performance on the high-quality WFLW [258] dataset. However, its performance dips significantly on the WiderFace dataset, populated with lower-quality images, as demonstrated in Tab. 5.10. This evaluation is not aimed at drawing a direct comparison between SLPT and DFA, as DFA is trained specifically on WiderFace. Instead, it serves to underline the performance variations of landmark detectors when trained on diverse datasets, stressing the importance of training dataset selection. Additionally, DFA boasts a magnitude faster speed than SLPT.

Since SLPT predicts 98 landmarks compared to 5 landmarks in DFA, we convert the SLPT landmarks by selecting indices that represent the left eye, right eye, nose, left mouth, and right mouth. An example is shown in Fig. 5.10.

Models	Train Data	NME	FLOP	Params
DFA MobileNet	WiderFace [269]	0.0518	0.14 GFLOP	0.49M
SLPT [261] 6 Layer	WFLW [258]	0.1104	8.40 GFLOP	13.19M

Table 5.10 Comparison of DFA to SoTA Landmark detector. Note that NME is evaluated on on WiderFace Validation set. DFA is trained on WiderFace training set. SLPT is trained on WFLW. Direct NME comparison is not fair as the training dataset is different.

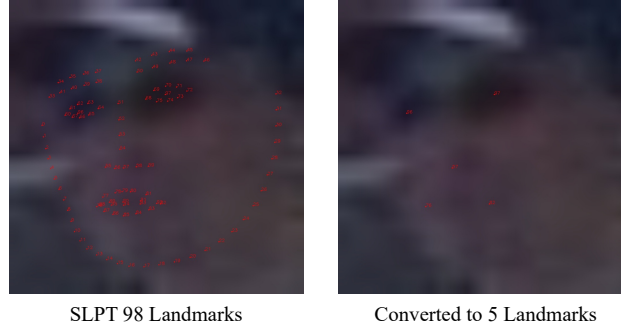


Figure 5.10 For converting 98 points landmarks from SLPT output, we choose indices 96, 97, 57, 76, 82.

### 5.12 Pipeline Detail

In this section, we elaborate on the inference scenarios involved in evaluation pipelines. A face recognition pipeline could be simplified to the following diagram. For a given raw image (a), the face detector crops out an image containing a face region (b). Then a conventional alignment algorithm (MTCNN, RetinaFace, DFA) simultaneously predicts the landmarks (c) from (b). The least-square minimization algorithm is used to align (b) into the aligned image (d) using keypoints (c) and a reference landmark. This reference landmark is arbitrarily chosen, but the FR community usually adopts one popular setting.

When one trains or evaluates face recognition models, most of the time, it is using aligned images (d), highlighted by the blue path. In our main paper, Tables 1,2, and 4, the aligned dataset and low-quality dataset are evaluated this way. The unaligned dataset in Tables 1 and 2 refers to the orange path. Whenever KPRPE is used, the keypoints are predicted using the inputs (b) or (d) depending on the path.

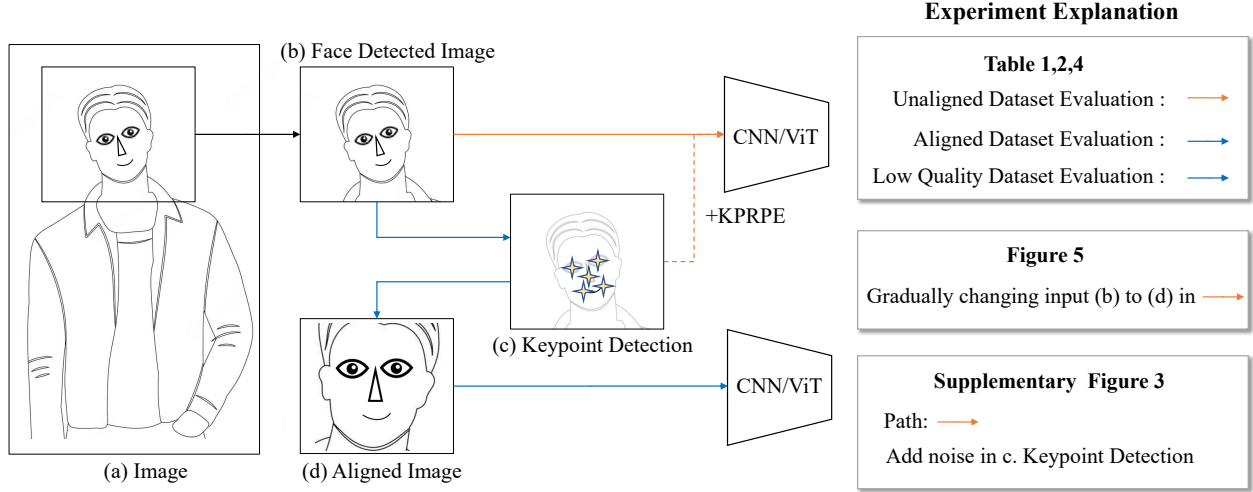


Figure 5.11 An illustration of face recognition pipeline from the raw image (a) to the aligned image (d).

### 5.13 KPRPE Visualization

We show the learned attention offset values in KPRPE. The red star denotes the query location and the blue circles represent the predicted landmarks. We pick head index 0 and plot the Transformer depth 0,1,3,5,7. Figs. 5.12 show different patterns of learned offset based on depth and query locations. Note that the higher values are denoted by a stronger blue color. Some attention offsets are 1) far from the query location, 2) horizontal pattern, etc. But there is an inherent bias toward attending nearby pixels.

Also, we show in Fig. 5.13, an image with different images, therefore different landmark patterns. The changes in attention are not as dramatic as the changes across different head or depth. However, these changes observed in Fig. 5.13 account for the spatial variations in the image once they accumulate over all of the attention modules in the model.

### 5.14 Conclusion

In this work, we introduce Keypoint-based Relative Position Encoding (KP-RPE), a method designed to enhance the robustness of recognition models to alignment errors. Our method uniquely establishes key-query relationships in self-attention based on their distance to the keypoints, improving its performance across a variety of datasets, including those

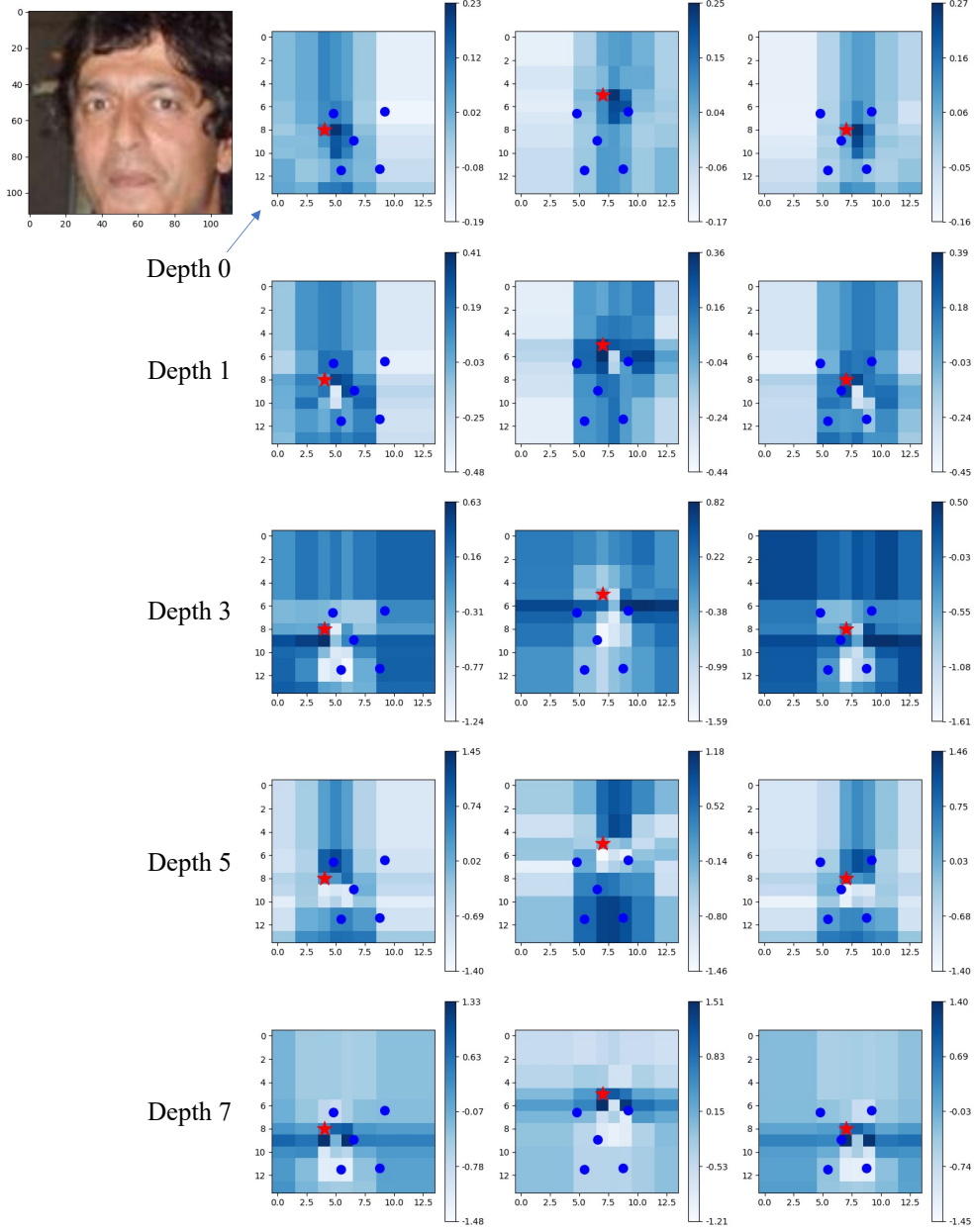


Figure 5.12 KPRPE Learned Offset  $\mathbf{B}_{ij}$  visualization for different Transformer depths.

with low-quality or misaligned images. KP-RPE demonstrates superior efficiency in terms of computational cost, throughput and recognition performance, especially when affine transform robustness is beneficial. We believe that KP-RPE opens a new avenue in recognition research, paving the way for the development of more robust models.

**Limitations** While KP-RPE shows impressive face recognition capabilities, it does require keypoint supervision, which may not always be readily available and can constrain its

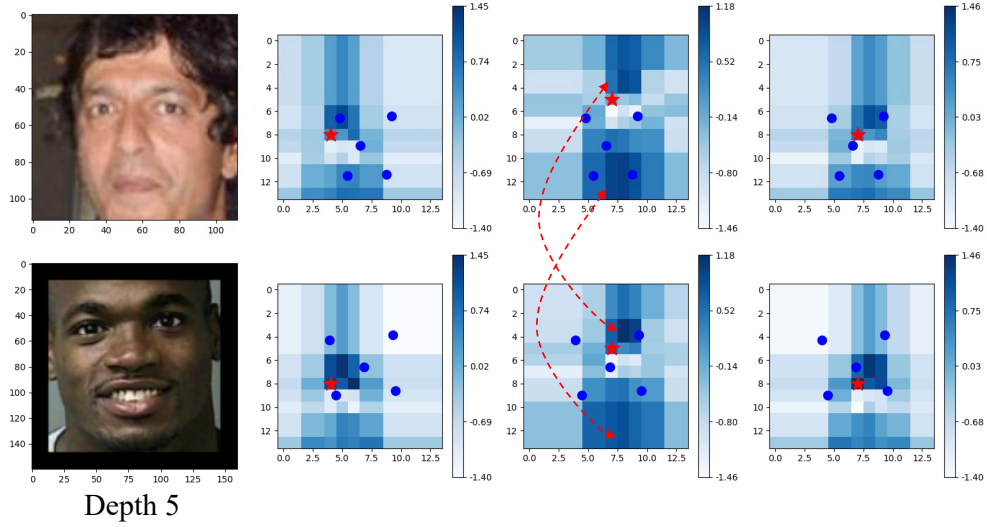


Figure 5.13 Cross image learned KPRPE visualization. We show the depth 5, and head index 0 of the same model.

application, particularly when the dataset is not comprised of images with a consistent topology. Future work should consider the self-discovery of keypoints to lessen this dependence, thereby boosting the model’s flexibility.

**Potential Societal Impacts** Within the CV/ML community, we must strive to mitigate any negative societal impacts. This study uses the MS1MV\* dataset, derived from the discontinued MS-Celeb, to allow a fair comparison with SoTA methods. However, we encourage a shift towards newer datasets, showcasing results using the recent WebFace4M dataset. Data collection ethics are paramount, often requiring IRB approval for human data collection. Most face recognition datasets likely lack IRB approval due to their collection methods. We support the community in gathering large, consent-based datasets or fully synthetic datasets [17, 124], enabling research without societal backlash.

## CHAPTER 6

### SAPIENSID: FOUNDATION MODEL FOR UNIFIED HUMAN RECOGNITION

Existing human recognition systems often rely on separate, specialized models for face and body analysis, limiting their effectiveness in real-world scenarios where pose, visibility, and context vary widely. This paper introduces SapiensID, a unified model that bridges this gap, achieving robust performance across diverse settings. SapiensID introduces (i) Retina Patch (RP), a dynamic patch generation scheme that adapts to subject scale and ensures consistent tokenization of regions of interest, (ii) a masked recognition model (MRM) that learns from variable token length, and (iii) Semantic Attention Head (SAH), an module that learns pose-invariant representations by pooling features around key body parts. To facilitate training, we introduce WebBody4M, a large-scale dataset capturing diverse poses and scale variations. Extensive experiments demonstrate that SapiensID achieves state-of-the-art results on various body ReID benchmarks, outperforming specialized models in both short-term and long-term scenarios while remaining competitive with dedicated face recognition systems. Furthermore, SapiensID establishes a strong baseline for the newly introduced challenge of Cross Pose-Scale ReID, demonstrating its ability to generalize to complex, real-world conditions. The dataset, code and models will be released.

#### 6.1 Introduction

Human recognition has traditionally been approached through domain-specific models focused exclusively on either face [55, 102, 122–124, 128, 154, 239, 240] or body [80, 110, 140, 149, 151, 268] recognition (or ReID). Each of these modalities relies heavily on specific dataset alignments, where face recognition models are optimized for tightly cropped, aligned facial images [1, 54, 82, 300], and body recognition models are designed to process full-body images of standing individuals [212, 250, 268, 295].

Despite the advances in both face and body recognition, no single model has yet effectively managed to handle a diverse range of poses and visible area simultaneously. However,



Figure 6.1 SapiensID is a human recognition model trained on a large-scale dataset of human images featuring varied poses and visible body parts. For the first time, a single model performs effectively across diverse face and body benchmarks [100, 212, 268, 297]. This marks a significant improvement over previous body recognition models, which were often limited to one specific camera setup or image alignments for one model, with worse performance in in-the-wild scenarios. Additionally, we introduce a large-scale, cross-pose and cross-scale training and evaluation set designed to facilitate further research in this area. — The name SapiensID pertains to the ability to recognize humans.

in real-world scenarios, human recognition often requires harnessing the full spectrum of available clues, integrating both face and body information. Typically, individual modality outputs are fused at the feature or score level [87, 147] to mitigate this issue. In other words, no single model can handle both face image and body image at the same time as robustly as the modality-specific model. A unified model would mark a significant advance in human recognition, freeing users from constraints on visible facial or standing-body views and allowing reliable identification across varied poses and scales of different body parts. As shown in Fig. 6.2, current research on body recognition models relies heavily on in-domain datasets, fail to generalize effectively to other datasets.

Addressing this gap is important for several reasons. In real-world applications, human recognition systems should operate across a variety of poses (sitting vs standing) and visible contextual areas (upper torso vs whole body) [271]. Furthermore, a model capable of handling varied inputs simplifies model deployment and usage for downstream tasks by eliminating the need for preprocessing steps such as face alignment [54] or dependency on camera

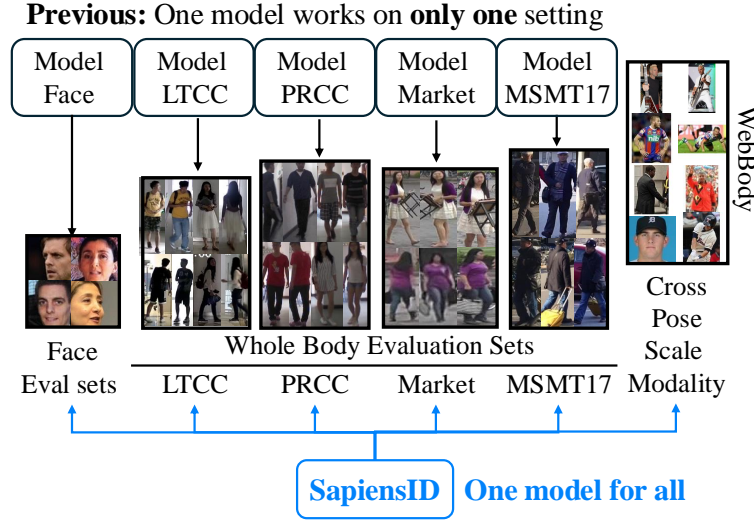


Figure 6.2 Conventionally, face and body recognition were handled independently.

Furthermore, body recognition models were trained on one specific dataset without the ability to generalize to other datasets. SapiensID model for the first time generalizes across modalities and different body poses and camera settings.

setups [212, 268].

However, addressing this problem is not trivial. First, it requires a large-scale labeled human image dataset that captures a wide range of poses and visibility variations. Secondly, even with such a dataset, the model must be capable of managing the substantial variability in scale and pose that human images naturally show. As in Fig. 6.1, close-up portraits show a large face, while full-body shots display it much smaller. Modality-specific models have eliminated the scale inconsistency problem with some form of pre-alignment stage. For instance, body recognition models assume consistent camera setup [212, 268] and face recognition models assume the images are aligned with 5 facial landmarks to a canonical position [54, 285]. Such transformations of input reduce irrelevant variability in recognizing a person, making training easier. However, models fail to generalize when the preprocessing step fails [125].

To this end, we propose **SapiensID**, one model capable of handling the complexities of human recognition in diverse settings. Our contributions are

- **Model Innovations:** We introduce three major improvements over conventional specialized recognition models:

1. **Retina Patch** addresses scale variations often encountered in human images by dynamically allocating more patches to important regions.
2. **Masked Recognition Model** reduces the number of tokens, achieving  $8\times$  speed up in ViT during training.
3. **Semantic Attention Head** addresses pose variations by learning to pool features around keypoints.

- **Data Contribution:** To aid the development and evaluation of **SapiensID**, we release **WebBody4M** (Fig. 6.1), a large-scale dataset specifically designed for comprehensive human recognition across different poses and scales.

Our approach is a paradigm shift human recognition, laying the groundwork for research that bridges the gap between specialized models and holistic recognition systems.

## 6.2 Related Works

### 6.2.1 Face Recognition

Face Recognition (FR) matches query images to an enrolled identity database. State-of-the-art (SoTA) FR models are trained on large-scale datasets [55, 82, 300] with margin-based softmax losses [55, 102, 122, 154, 240]. FR performance is evaluated on a set of benchmarks, *e.g.* LFW [100], CFP-FP [202], CPLFW [296], AgeDB [174], CALFW [297], and IJB-B,C [169, 253]. They are designed to assess the model’s robustness to factors such as pose variations and age differences. Models trained on large datasets, *e.g.* WebFace260M, achieve over 97% verification accuracy on these benchmarks [122]. FR in low-quality imagery is substantially harder and TinyFace [46] and IJB-S [112] are popular benchmarks.

Face recognition is often accompanied by facial landmark prediction [31, 132, 224, 285] so that input faces are aligned and tightly cropped around the facial region. However, when alignment fails, FR models perform poorly [125]. Eliminating alignment would not only

simplify the pipeline but also enhance robustness in conditions where alignments are prone to fail. We propose an *alignment-free* paradigm capable of handling any human image with or without a visible face.

### 6.2.2 Body Recognition

Body recognition, *a.k.a.* Person Re-identification (ReID), seeks to identify individuals across different times, locations, or camera settings. Prior works [71, 72, 137, 138, 146, 241, 278, 284, 295] focus on short-term scenarios where subjects generally end up with the same attire. Removing this assumption has led to long-term, cloth-changing ReID [41, 80, 95, 110, 140, 238, 268, 280], on datasets such as PRCC [268], LTCC [212], CCDA [149] and CelebReID [103, 104].

All of these datasets are composed primarily of whole-body images, where the subjects are fully visible from head to toe, with poses generally limited to walking or standing. While this format has been valuable in the development of person ReID models for controlled environments, it lacks the scale and visibility variety often encountered in real-world applications. To address these limitations, we propose a novel model capable of handling diverse and complex poses and visible areas. Further, to facilitate the training and evaluation of these models, we introduce a new large-scale, labeled dataset that significantly broadens pose-scale diversity.

### 6.2.3 Patch Generation for Vision Transformers

In Vision Transformer (ViT) [61], an image is divided into patches, with each transformed into a token via linear projection. This patch-based approach transforms images to an unordered set of tokens for sequence-to-sequence modeling [236], processing images in a scalable and flexible way in downstream tasks. Typically, patches are created by dividing an image into a grid with a specific number of patches.

Several works explore how the patchifying process helps ViT capture multi-scale objects in images [249]. For instance, [51] predefines patch counts without resizing the input, retaining the image’s aspect ratio and scale. [22] randomizes patch sizes in training for generalization across image scales, enhancing efficiency while sometimes reducing accuracy. Importantly, the representation quality of specific regions, such as face or hand, depends on **the number**

**of tokens** allocated to those areas. A smaller face within a constant patch size, for example, generates fewer tokens and thus captures less detail than a larger face. To address this, we propose to maintain a consistent number of tokens for regions of interest while ensuring full, non-overlapping coverage across the image in line with grid-based tokenization principles.

### 6.3 Proposed Method

A human recognition model is formulated as a metric learning task such that images of the same subject are closer in feature space than those of different subjects, satisfying

$$d(\mathbf{f}_A^i, \mathbf{f}_A^j) < d(\mathbf{f}_A^i, \mathbf{f}_B^k), \quad (6.1)$$

where  $\mathbf{f}_A^i$  and  $\mathbf{f}_A^j$  denote the feature vectors of two different images  $i$  and  $j$  of the same subject  $A$ , while  $\mathbf{f}_B^k$  represents the feature vector of an image of a different subject  $B$ . Notably, the subjects  $A$  and  $B$  are not observed during training. Following established research on margin-based techniques for enhancing intra-class compactness in the feature space [55, 122, 154, 172, 240], we utilize a margin-based softmax loss [122] to train our model on a labeled dataset. We collect a large-scale web-collected human image training dataset which will be discussed in Sec. 6.3.4.

The key challenge that sets this apart from prior work on a separate face [55, 154] or body [140, 268] recognition task is that the input image can be highly varying in 1) scale and 2) body pose. To tackle these challenges, we propose a new architecture, which will be discussed in the subsections.

#### 6.3.1 Retina Patch (RP)

To address the issue of varying scale in human images, we propose a novel **Retina Patch** mechanism inspired by the human eye’s ability to adapt focus dynamically to regions of interest (ROIs) within a scene. In natural images, subjects can appear in diverse poses and with varying visibility of the face and body, leading to substantial differences in scale across regions. For instance, in a full-body image, a face may be a small portion, whereas in a close-up, it dominates. To account for these variations, our Retina Patch dynamically assigns

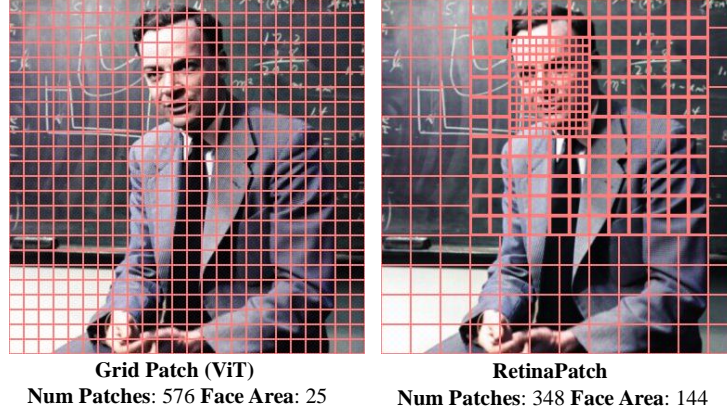


Figure 6.3 Comparison between the standard grid patch scheme of Vision Transformers (ViT) and our Retina Patch. While maintaining the same or lower computational budget (number of tokens), Retina Patch dynamically allocates more patches to critical regions (e.g., face and upper torso) in an image. This allocation enhances the model’s ability to capture fine-grained details in important regions, and to handle varying scales more effectively than fixed grid patch.

more patches to critical regions within the image.

Assume we have an input image  $i$  and a set of image-dependent regions of interest,  $\{\text{ROI}_r^i \mid r = 0, 1, \dots, R\}$ , each defined by a bounding box. There are  $R$  ROIs per image. Details on how ROIs are computed will be discussed later. We also let  $\text{ROI}_0^i$  be the whole image. For each  $\text{ROI}_r^i$ , we set a specific number of patches  $m_r$  and an order  $z_r$ , both controlling how many patches can come from each  $\text{ROI}_r^i$ .

To obtain patches, we may perform a grid patching operation on each ROI independently. However, this would naturally result in overlapping patches with redundant feature extraction. Our aim is to cover the whole image with patches *without any overlap*. To avoid redundancy, overlapping patches between regions with a lower order (e.g., order  $z = 1$ ) and those with a higher order (e.g., order  $z = 2$ ) are excluded from the patch set of the low-order regions. This selective inclusion process ensures that each patch belongs uniquely to the ROI with the

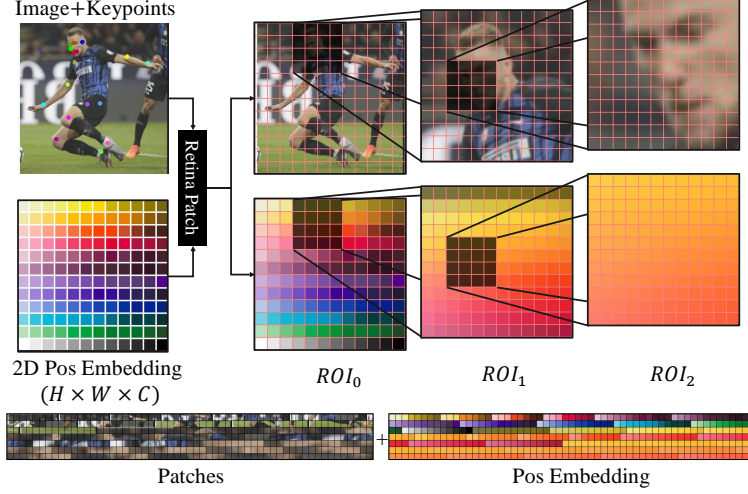


Figure 6.4 Illustration of Retina Patch and Position Encoding computation. **Top:** It shows three different ROIs generating patches at various scales (e.g., full image, upper torso, face). It also shows the corresponding position encodings sampled from the same spatial locations as the patches, allowing ViT to infer spatial context and understand where each patch originated within the image. **Bottom:** patches and position embedding created by Retina Patch.

highest priority, as indicated by the order. Specifically,

$$P^i = \bigcup_{r_1=0}^R \left( P_{\text{ROI}_{r_1}}^i - \bigcup_{r_2=r_1+1}^R P_{\text{ROI}_{r_2}}^i \right), \quad (6.2)$$

where  $P_{\text{ROI}_r}^i$  represents the set of patches for region  $\text{ROI}_r$  of image  $i$ , and  $r$  denotes the index of each ROI, ordered by their respective priorities for patch inclusion.

This approach allows us to dynamically allocate critical regions with more patches while ensuring that the entire image is represented by patches without repetition. Also, the scale inconsistency is mitigated as long as the ROIs are semantically defined (*e.g.*, *face*, *upper torso*). The number of patches within each ROI is kept consistent across images, ensuring that each patch covers a similar scale within its designated ROI. Fig. 6.3 uses an example to compare the vanilla grid patch of ViT with our proposed Retina Patch.

**Computing ROI** Retina Patch is a generic algorithm that can work for any class of images by designing ROIs for the particular domain. In this paper, for recognizing a subject from a

human image, we set the ROIs in 3 parts: 1) whole image, 2) upper torso and 3) face. The upper torso and face ROIs are computed using the off-the-shelf body keypoint detector [34].

**Tokenization** The input to ViT’s transformer block is a set of tokens or feature vectors. Since each patch’s size is dependent on both the ROI size and the number of patches  $m_r$ , the size of each patch may not be the same across ROIs. We simply resize all patches to be the size of patches from the whole image ROI<sub>0</sub><sup>*i*</sup>. We then use a linear layer to map each patch to the desired dimension, as in ViT.

**Position Embedding** Since Transformer operates on sets of tokens without inherent order, Position Embedding (PE) is crucial for informing ViT of the spatial origin of each patch within the original image. For tokens of Retina Patch, we cannot use a traditional PE as the patch’s source location is dynamic. Thus, we propose a Region-Sampled PE.

Let  $PE \in \mathbb{R}^{C \times H \times W}$  be the fixed 2D sin-cosine position embedding [23, 45] for the whole image. Given a normalized region of interest  $ROI_r^i = (x_r^i, y_r^i, h_r^i, w_r^i)$  with values between 0 and 1, we define a sampling grid  $Grid_{ROI_r^i}$  over the region  $[x_r^i, x_r^i + w_r^i]$  and  $[y_r^i, y_r^i + h_r^i]$  within the position embedding PE. Let  $(h'_r, w'_r)$  be the target output shape for  $PE_{ROI_r^i}$ , such that  $h'_r \cdot w'_r = m_r$ , the desired number of patches for  $ROI_r^i$ . The Region Sampled PE,  $PE_{ROI_r^i}$  is then obtained by bilinearly interpolating PE at the points in  $Grid_{ROI_r^i}$  to match the shape  $(h'_r, w'_r)$ :

$$PE_{ROI_r^i} = \text{GridSample}(PE, Grid_{ROI_r^i}, (h'_r, w'_r)). \quad (6.3)$$

By using region-specific position embeddings, Retina Patch enables the model to differentiate between patches from distinct areas of the image while preserving spatial structure similar to the patches. An example is shown in Fig. 6.4.

### 6.3.2 Masked Recognition Model (MRM)

For each image, Retina Patch results in different numbers of tokens because different ROIs create different areas of intersection. For example, the number of patches from ROI<sub>0</sub> in Fig. 6.4 is  $12 \times 12$  but the upper torso ROI<sub>1</sub> subtracts  $4 \times 4$  patches from ROI<sub>0</sub> to avoid overlap. This operation leads to a different number of tokens per image, which prevents

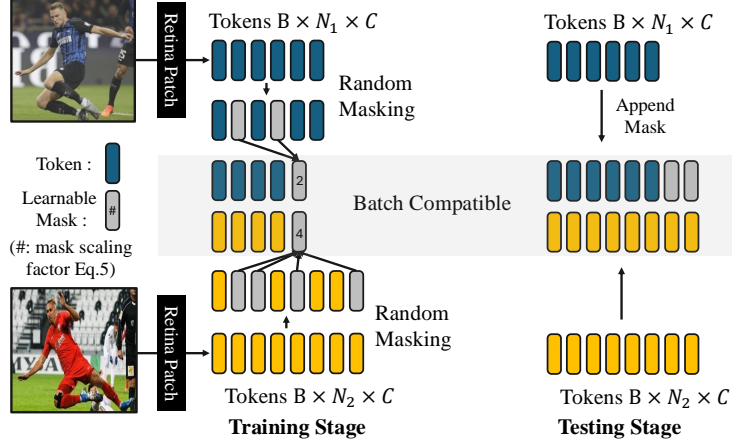


Figure 6.5 Illustration of Masked Recognition Backbone with masking and attention scaling trick for batched input during training. In testing, we pad with mask tokens to make the length the same.

us from training and testing with batched inputs. To address the token inconsistency, we propose the Masked Recognition Model (MRM), introducing two key techniques: (1) masking with attention scaling and (2) a variable masking rate.

**Masking with Attention Scaling** During training, we select tokens to keep. Unlike MAE [85], which discards the masked tokens, we replace them with a learnable mask token. We do this because (i) the mask token will be used during testing for padding the input, and (ii) this allows the model to explicitly know *how many* tokens are masked. Yet, since all masked tokens share the same value, we can reduce computation by applying the Attention Scaling Trick.

Specifically, although there are multiple masked tokens, we can achieve the same effect with a single mask token by adjusting its attention scores to reflect the total number of masked tokens. Let  $n_i$  be the total number of tokens for  $i$ -th image,  $n_k$  be the number of tokens we keep, and  $n_{m,i} = n_i - n_k$  be the number of masked tokens. We modify the attention computation in the Transformer as:

$$A = \text{softmax} \left( \mathbf{QK}^\top / \sqrt{d} + \delta \right), \quad (6.4)$$

where  $\mathbf{Q} \in \mathbb{R}^{(n_k+1) \times d}$  and  $\mathbf{K} \in \mathbb{R}^{(n_k+1) \times d}$  are the query and key matrices with tokens to keep

and one mask token.  $d$  is the embedding dimension. We add a bias matrix  $\delta \in \mathbb{R}^{n \times n}$  so that it is mathematically equivalent to repeating the mask tokens  $n_{m,i}$  times during attention computation.

$$\delta_{ij} = \begin{cases} \log n_{m,i}, & \text{if } j \text{ is the mask token,} \\ 0, & \text{otherwise.} \end{cases} \quad (6.5)$$

In summary, we reduce the number of tokens from  $n_i$  to  $(n_k + 1)$ . Note that  $(n_k + 1)$  is fixed and not image dependent. But we adjust the attention to make it equivalent to using  $n_i$  tokens where  $n_{m,i}$  tokens replaced by learnable mask tokens. By applying the Attention Scaling Trick, we handle varying token counts in training. Also in practice,  $n_k$  is set to be about 1/3 of  $n_i$ , masking 66% of tokens for the speed gain. During testing, we simply find the longest token length and pad the others with the mask token to batchify the inputs. An illustration is in Fig. 6.5.

**Variable Masking Rate** As we view masked training as a form of augmentation, we randomize  $n_k$  during training and adjust the batch size correspondingly. For each batch, let  $\hat{n}_k$  be the sampled number of tokens to keep,

$$\hat{n}_k = n_k + (n_i - n_k) \cdot e^{-\lambda \cdot U(0,1)}. \quad (6.6)$$

$\lambda$  is a scaling factor, and  $U(0,1)$  denotes a random uniform distribution between 0 and 1. In short,  $\hat{n}_k$  is sampled from a distribution that peaks at  $n_k$  and exhibits an exponential decay in probability toward  $n_i$ .

With a randomized token length  $n_k$ , we adjust the batch size  $B$  based on the relationship  $n_k^2 \propto \frac{1}{B}$ , where increasing  $n_k$  would require decreasing  $B$  to maintain the same GPU memory and FLOP. And we adjust the learning rate according to the effective batch size  $\mathcal{L}_{\text{adj}} = \mathcal{L}_{\hat{n}_k} \times B_{\hat{n}_k} / B_{n_k}$  to maintain consistent gradient magnitudes per sample.

The effect of (1) masking with attention scaling and (2) variable masking rate is ablated in Tab 6.5. While (1) and (2) are both helpful, the effect of (2) is more pronounced.

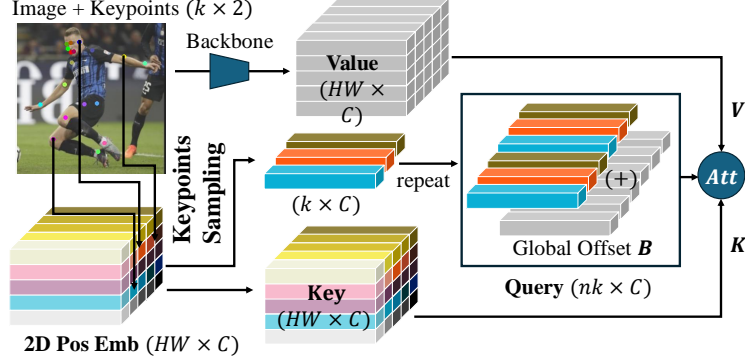


Figure 6.6 Illustration of semantic pooling in Semantic Attention Head. Keypoints (e.g., nose, feet) are used to grid-sample position embeddings (PE), forming queries that repeat  $n$  times and added with a global offset bias  $\mathbb{B}$ . This setup enables attention to slightly varied locations around each keypoint. Value comes from ViT backbone and Key is the PE. Result is a learned pooling mechanism.

### 6.3.3 Semantic Attention Head (SAH)

In biometric recognition, the head module is key for converting the backbone’s output feature map into a compact feature vector for recognition. Face recognition models flatten the feature map and apply linear layers [55, 122], while body recognition models use horizontal pooling [34, 272]. However, these approaches rely on input image alignment (aligned face or standing body) which fails when there are large pose variations. To tackle this, we introduce a Semantic Attention Head (SAH) that extracts semantic part features from key body parts, making the representation less sensitive to pose.

Our method uses keypoints (e.g., nose, hip) for capturing semantic parts. But instead of sampling features only at keypoints, which may miss the surrounding context, SAH *learns* to pool features around each keypoint. We construct a semantic query  $\mathbf{Q}_{kp}^i$  (e.g., nose) using 2D position embeddings (PE) from the backbone, sampled at keypoint locations:

$$\mathbf{Q}_{kp}^i = \text{GridSample}(\text{PE}, \text{kp}^i) + \mathbf{B}, \quad (6.7)$$

where PE is the fixed 2D image position embedding.  $\text{kp}_i \in \mathbb{R}^{n \times 2}$  is the image-specific predicted keypoints [34]. We duplicate keypoints  $n$  times and add shared bias  $\mathbf{B} \in \mathbb{R}^{n \times C}$ .

The purpose of  $\mathbf{B}$  is to learn to offset the center of attention so that it learns to pool from diverse locations around keypoints. Key in attention is the fixed PE. Value is the backbone’s feature map. The attention with  $\mathbf{Q}_{kp}^i$  captures the neighborhood of the backbone feature map around keypoints:

$$\mathbf{O}_{\text{part}}^i = \text{Attention}(\mathbf{Q}_{kp}^i, \text{PE}, \text{backbone}(\mathbf{X}^i)). \quad (6.8)$$

The  $\mathbf{O}_{\text{part}}^i \in \mathbb{R}^{B \times k \times C}$  contains semantic part features corresponding to  $k$  keypoints. Finally, applying a multi-layer perceptron (MLP) to the flattened  $\mathbf{O}_{\text{part}}^i$  produces a feature,

$$f^i = \text{MLP}(\text{flatten}(\mathbf{O}_{\text{part}}^i)). \quad (6.9)$$

By learning to pool features adaptively around each keypoint, this attention mechanism enables pose-invariant recognition that goes beyond conventional alignment-dependent methods. Fig. 6.6 illustrates the attention pooling.

**Training with Mixed Datasets** While SAH effectively handles pose variations, we hypothesize that key cues for recognition differ between short-term and long-term training datasets. Clothing and hairstyle, for example, are useful in short-term datasets but less reliable in long-term due to possible appearance changes.

To aid learning with mixed datasets which combines short-term and long-term datasets, we introduce one more measure during training. We introduce a learnable scale that controls the importance of individual part features in  $(\mathbf{O}_{\text{part}}^i)$  for each dataset. It is to allow the model to emphasize features that are most discriminative for each dataset. During testing, however, we can use the average scale because we do not want utilize the knowledge about the test dataset a priori.

Specifically, let  $\mathbf{W}_t \in \mathbb{R}^k$  be a weight for the  $t$ -th dataset. For each sample, we choose the weight and apply

$$f^i = \text{MLP}(\text{flatten}(\mathbf{O}_{\text{part}}^i \cdot \sigma(\mathbf{W}_t))), \quad (6.10)$$

where  $\sigma$  is the Sigmoid function, ensuring weights are between 0 and 1, controlling the influence of each of the  $k$  semantic parts. We observe that after training, short-term datasets

Method	Arch	Train Data	Avg	LTCC (General) top1	PRCC (SC) [268] mAP	CCVID (General) top1	Market1501 top1	MSMT17 [250] mAP					
CAL [80]	R50	LTCC	48.64	74.04	40.84	99.51	95.64	75.63	28.08	35.60	16.11	15.92	5.06
CAL [80]	R50	PRCC	35.07	20.69	6.19	100.00	99.76	74.48	20.86	18.97	6.47	2.56	0.69
CAL [80]	R50	LTCC+PRCC	49.69	72.41	38.12	99.54	99.01	74.83	29.43	43.65	21.03	14.48	4.44
CLIP3DReID [151]	R50	LTCC	50.89	<b>75.66</b>	<b>45.15</b>	99.43	96.43	77.28	30.01	41.66	20.33	17.45	5.50
CLIP3DReID [151]	R50	PRCC	35.14	21.30	6.19	100.00	<b>99.84</b>	71.73	19.81	20.93	7.49	3.28	0.85
SOLDIER [44]	Swin-Base	LU4M+Market1501	64.85	73.83	36.28	99.51	99.53	40.27	36.56	<b>97.03</b>	<b>94.04</b>	48.64	22.77
SOLDIER [44]	Swin-Base	LU4M+MSMT17	70.19	74.44	36.74	99.30	98.71	32.73	27.76	89.85	73.20	<b>91.12</b>	<b>78.01</b>
HAP [281]	ViT-Base	LU4M+LTCC	45.71	65.11	29.02	95.53	86.44	44.16	30.43	51.63	27.29	20.89	6.56
HAP [281]	ViT-Base	LU4M+PRCC	54.09	63.29	29.36	98.84	98.38	49.15	37.73	73.49	50.11	29.61	10.99
HAP [281]	ViT-Base	LU4M+Market1501	66.61	73.02	35.97	99.30	98.45	54.74	45.14	96.23	92.20	48.01	23.02
HAP [281]	ViT-Base	LU4M+MSMT17	66.64	67.95	32.07	99.15	96.50	37.81	30.52	80.37	57.07	89.13	75.85
HAP [281]	ViT-Base	WebBody4M (Ours)	61.49	56.80	25.88	99.72	98.26	89.00	71.65	66.18	42.41	43.61	21.42
SapiensID (Ours)	ViT-Base	WebBody4M (Ours)	<b>73.05</b>	72.01	34.56	<b>100.00</b>	98.79	<b>92.57</b>	<b>77.82</b>	88.18	68.26	67.25	31.02

Method	Arch	Train Data	Avg	LTCC (CC) [212] top1	PRCC (CC) mAP	CCVID (CC) [80] top1	CCDA [149] mAP	Celeb-ReID [103] top1					
CAL [80]	R50	LTCC	28.40	38.01	18.84	37.00	35.20	74.97	25.08	3.91	9.67	37.42	3.92
CAL [80]	R50	PRCC	24.71	6.38	3.14	55.69	55.64	71.61	17.40	2.85	8.61	23.59	2.20
CAL [80]	R50	LTCC+PRCC	29.46	33.16	16.27	45.39	45.42	73.89	26.65	3.74	9.14	37.11	3.81
CLIP3DReID [151]	R50	LTCC	30.24	41.84	<b>22.58</b>	40.81	38.38	76.28	26.69	4.31	10.18	37.31	4.02
CLIP3DReID [151]	R50	PRCC	25.79	6.63	3.17	62.40	61.97	69.32	16.38	3.17	8.89	23.82	2.17
SOLDIER [44]	Swin-Base	LU4M+Market1501	24.84	25.00	12.18	26.87	32.12	39.61	35.48	8.62	16.48	46.37	5.66
SOLDIER [44]	Swin-Base	LU4M+MSMT17	22.17	26.02	11.33	22.27	25.36	31.85	26.48	8.79	15.54	47.95	6.14
HAP [281]	ViT-Base	LU4M+LTCC	20.21	25.00	11.63	26.14	22.34	41.64	25.77	4.56	11.18	30.28	3.54
HAP [281]	ViT-Base	LU4M+PRCC	26.12	29.08	12.52	38.05	41.94	45.73	33.12	5.13	13.40	37.79	4.48
HAP [281]	ViT-Base	LU4M+Market1501	27.49	24.74	11.71	33.90	37.00	52.37	41.33	8.30	16.02	44.38	5.20
HAP [281]	ViT-Base	LU4M+MSMT17	21.61	23.47	10.74	23.82	25.00	34.54	26.81	6.27	13.33	46.37	5.77
HAP [281]	ViT-Base	WebBody4M (Ours)	44.90	22.70	9.96	54.93	49.38	88.34	68.66	28.80	41.49	65.78	18.93
SapiensID (Ours)	ViT-Base	WebBody4M (Ours)	<b>66.30</b>	<b>42.35</b>	17.79	<b>78.75</b>	<b>72.60</b>	<b>88.72</b>	<b>72.22</b>	<b>61.84</b>	<b>69.08</b>	<b>92.80</b>	<b>66.92</b>

Table 6.1 Generalization comparison with SoTA ReID models on two settings. "Long-term" refers to clothing change (CC) protocol of LTCC, PRCC, and CCVID datasets, while "short-term" the same clothing (SC) protocol. For other datasets, the data capture characteristics define short or long-term conditions. SapiensID demonstrates superior generalization in both settings. Our WebBody4M dataset shows higher performance in long-term ReID, but not with the dataset alone, as shown in the comparison of HAP vs SapiensID with the same training set. The proposed Retina-Patch and Semantic Attention Head are essential for learning under large pose and scale variations.

tend to focus on the clothing and long-term datasets focus on the upper torso. The weight is for learning discriminative parts during training but we do not use dataset-specific weights in testing.

### 6.3.4 WebBody Dataset

To facilitate the training, we collect a large-scale, labeled human dataset from the web. Specifically, we gather 94 million images with 3.8 million celebrity names. Given the inherent noise in web-sourced name queries, we perform extensive label cleaning. First, we use YOLOv8 [111] to crop the dominant person in each image to a size of  $384 \times 384$ , adding padding to maintain aspect ratio. We then extract facial features using RetinaFace [54] and KP-RPE [125]. Following the approach in [300], we apply DBSCAN [65] clustering to identify

Method	Arch	Train Data	Avg	WebBody Testset top1 mAP
CAL [80]	R50	PRCC	2.47	4.29 0.64
CAL [80]	R50	LTCC	3.79	6.57 1.02
SOLDIER [44]	Swin-Base	Market1501	3.22	5.42 1.02
SOLDIER [44]	Swin-Base	MSMT17	5.96	9.95 1.98
HAP [281]	ViT-Base	LTCC	1.74	2.89 0.58
HAP [281]	ViT-Base	PRCC	2.61	4.37 0.85
HAP [281]	ViT-Base	Market1501	4.31	7.22 1.39
HAP [281]	ViT-Base	MSMT17	4.87	8.22 1.52
HAP [281]	ViT-Base	WebBody4M	47.12	64.36 29.89
SapiensID (Ours)	ViT-Base	WebBody4M	<b>64.41</b>	<b>76.82 52.00</b>

Table 6.2 ReID Performance on variable pose and scale settings.

the most consistent group of images for each name. By assuming all images stem from a single name query, we relax the similarity threshold beyond conventional face recognition standards. We also exclude any images with face features matching those in validation sets [100, 174, 202, 296, 297].

This process yields a labeled dataset with 4.4 million images across 217, 722 unique subjects. However, because the dataset is labeled based on facial similarity, it lacks images where the face is obscured (e.g., back-facing images). To address this, we incorporate additional body ReID training datasets [70, 80, 103, 212, 222, 264, 268, 295], which account for approximately 10% of the final dataset. After merging, the resulting dataset—named WebBody4M—comprises 4.9 million images and 263, 920 subjects in total. WebBody4M is the largest labeled dataset to date with high pose and scale variation. The keypoint visibility distribution of different body parts shows a predominance of visible upper body, with visibility decreasing gradually down the body (around 17% visible ankles). An example of the WebBody4M dataset can be seen in Fig. 6.1.

The dataset collection and label cleaning procedure is similar to WebFace4M dataset [300]. We compare the face-cropped version of WebBody4M with WebFace4M and observe that an FR model trained on WebBody4M-FaceCrop is similar in performance to WebFace4M. Separate from the WebBody4M, we also prepare a test set called WebBody-test to evaluate the cross pose-scale ReID performance. It comprises 96, 624 images of 4, 000 gallery and probe subjects. Examples are shown in Fig. 6.2.

## 6.4 Experiments

**Implementation Details** To train SapiensID on Webbody4M, we use AdaFace [122] loss and ViT-Base with KP-RPE as the main backbone [125], following the convention of face recognition model training pipeline. We do not include additional losses such as Triplet Loss [200] since there are a sufficient number of subjects in the training set. Input image size is  $384 \times 384$  with white padding if the aspect ratio is not 1. We use 3 ROIs (whole image, upper torso, and head) and the grid size per ROI is  $12 \times 12$  leading to a maximum  $144 \times 3$  number of patches. With masked recognition training, we replace at most 66% of tokens with mask (Sec. 6.3.2), leading to  $\sim 9$  times speed up in training. The masking probability and batch size rule are discussed later. We use 7 H100 GPUs to train the whole model in 2 days, starting from scratch.

**Whole Body ReID** The task identifies individuals walking or standing in distant camera views, categorized into short or long-term scenarios based on the time gap between captures and the likelihood of clothing changes. Tab. 6.1 shows our results on the ReID benchmarks. A significant departure from prior works is the use of a single SapiensID model across all evaluation settings, whereas previous methods employ fine-tuned models for each evaluation dataset (one model per dataset). This distinction highlights SapiensID’s potential for deployment in diverse, unseen, real-world environments.

SapiensID achieves the highest average mAP of 73.05% across short-term ReID benchmarks. Furthermore, we attain SoTA results on all evaluated long-term ReID datasets. This strong performance underscores the value of the WebBody4M dataset in training a generalizable model. However, this achievement would not have been possible without our SapiensID architecture, which effectively handles variations in pose and visible body areas. A strong baseline (HAP [281]) trained on WebBody4M alone does not achieve comparable results, highlighting the importance of our architectural innovations to leverage the dataset. SapiensID marks a significant advance by being the first single model capable of strong performance across short and long-term ReID tasks.

Method	Training Data	OccludedReID	
		top1	mAP
KPR [214] + SOLDIER	LU4M +OccludedReID	84.80	<b>82.60</b>
SapiensID	WebBody4M	<b>87.30</b>	75.57

Table 6.3 Performance in occluded ReID. SapiensID achieves a higher top-1 accuracy, while KPR [214] shows a higher mAP. SapiensID is trained without OccludedReID training data.

Method	AdaFace-ViT [122]	SapiensID (Ours)
Train Data	WebBody4M-FaceCrop	WebBody4M
LFW [100]	99.82	99.82
CPLFW [296]	95.12	94.85
CFPFP [202]	99.19	98.74
CALFW [297]	96.07	95.78
AGEDB [174]	97.97	97.33
Face Avg	<b>97.63</b>	97.31
LTCC [212]	21.70	72.01
Market1501 [295]	7.81	88.18
Body Avg	14.76	<b>80.10</b>
Combined Avg	56.19	<b>89.80</b>

Table 6.4 Performance on cross-modality setting. Face recognition is evaluated on aligned face recognition datasets and body recognition is evaluated on short-term ReID datasets. LTCC and Market1501 measure top1 of short-term setting.

**Cross Pose-Scale ReID** Real-world human recognition can present scenarios where subjects are captured across varying camera viewpoints and exhibit diverse poses, such as sitting, bending, or engaging in activities. For example, a security camera might capture a person standing upright, while a social media photo shows the same individual sitting in a cafe. This poses a challenge for conventional ReID systems. We refer to this setting as Cross Pose-Scale ReID.

To evaluate this setting, we introduce the WebBody-Test dataset, specifically designed to encompass such pose and scale variations. Tab. 6.2 details the performance comparison on this dataset. Conventional ReID models struggle to generalize to this scenario due to the significant shift in visual appearance caused by pose and scale changes. SapiensID with the highest performance establishes a strong baseline for this research area. Since the task itself is challenging, there is still room for improvement. WebBody dataset demonstrates the potential of SapiensID to address the complexities of Cross Pose-Scale ReID, while offering a valuable starting point for future research in this area.

	All	Face	Whole Body ReID	
			Short	Long
(1) ViT	59.54	90.63	56.17	31.81
(2) ViT+RP	66.35	92.93	59.16	46.95
(3) ViT+SAH	71.67	95.84	72.63	46.55
(4) ViT+RP+SAH (SapiensID)	<b>78.67</b>	<b>96.66</b>	<b>73.05</b>	<b>66.30</b>
(4) – Learned Mask	76.99	96.08	70.44	64.46
(4) – Variable $n_k$	74.39	95.95	69.58	57.64

Table 6.5 Ablation study of SapiensID. Face is the average accuracy of CPLFW, CFPFP, CALFW, and AGEDB. Short and Long Term use the average of the datasets in Tab 6.1. Results show the necessity and strong complementarity of both RP and SAH in SapiensID.

		LTCC CC		PRCC CC	
		Top1	mAP	Top1	mAP
1	None	0.00	3.56	1.47	4.28
2	1+Nose	25.77	5.78	27.21	21.04
3	2+Eye	30.61	8.87	63.87	55.17
4	3+Mouth	38.01	11.81	73.36	65.05
5	4+Ear	39.80	14.05	77.65	70.45
6	5+Shoulder	41.84	15.82	79.73	73.14
7	6+Elbow	41.07	16.64	<b>80.55</b>	<b>73.54</b>
8	7+Wrist	41.07	17.16	79.34	73.16
9	8+Hip	40.56	17.50	79.99	73.38
10	9+Knee	42.35	17.73	79.00	72.88
11	10+Ankle (Full)	<b>42.35</b>	<b>17.79</b>	78.75	72.6

Table 6.6 Impact of adding body parts on ReID. None means all features are zeroed out. Each row adds features to the previous row.

**Occluded ReID** Occlusions, whether due to obstacles in the scene or self-occlusion from the subject’s pose, present a further challenge for robust human recognition. We evaluate SapiensID in occluded scenarios on the OccludedReID dataset [301], comparing with KPR [214], a SoTA method designed for occlusion handling. As shown in Tab. 6.3, SapiensID achieves a competitive performance of top-1 87.30%, demonstrating its strong ability to handle occlusions even without being explicitly trained on the OccludedReID dataset. This result further underscores the value of our architecture and training dataset in learning representations that are resilient to real-world challenges like occlusions.

**Face Recognition** We evaluate on traditional aligned face recognition benchmarks to assess the ability to handle FR tasks. Tab. 6.4 compares SapiensID with a SoTA FR model, AdaFace [122], both with a ViT-Base backbone. AdaFace is trained on faces aligned and cropped to  $112 \times 112$  by [54]. AdaFace achieves a slightly higher average accuracy of 97.63% across five benchmarks. This marginal difference is expected, given AdaFace’s training on

tightly cropped, aligned faces. However, SapiensID’s performance remains highly competitive, bridging the gap between specialized face recognition and general human recognition tasks.

While AdaFace excels in FR datasets, its performance degrades when applied to ReID datasets which contain images without visible face region (*e.g.* back of the face). AdaFace is evaluated by cropping faces using [54]. In contrast, SapiensID maintains strong performance across both modalities.

**Ablation of Components** Tab. 6.5 ablates SapiensID’s key components: Retina Patch (RP) and Semantic Attention Head (SAH). Starting from a simple ViT backbone with AvgMax pooling [80] as a baseline, we progressively incorporate RP and SAH to analyze their individual and combined contributions. Performance is evaluated across face recognition and both short-term and long-term ReID. The results show that both RP and SAH are essential.

We also show the importance of MRM. (4) - Learned Mask means using MAE [85] to simply drop tokens. (4) - Variable  $n_k$  is fixing  $n_k$  without sampling. The result shows that learned mask is of some benefit while changing the masking rate during training is of larger benefit.

**Analysis of Part Contribution** To see the impact of body parts in recognition, we erase part features by making them zero. Tab. 6.6 shows a trend of performance gain as more parts are added. For LTCC dataset accuracy increases from 25.77% to 42.35% as body parts from the nose to ankle are incorporated. This suggests that including the full range of body parts aids recognition. In contrast, PRCC achieves high performance by using upper body cues, reaching a top-1 accuracy of 80.55% with parts up to the shoulder and elbow. Lower body features add minimal or even negative value. This analysis implies the benefit of scenario-specific adjustments where relevant body regions can optimize recognition performance. We also visualize the part features similarity with sample images from the test set of WebBody4M in Fig 6.7. Samples of different scales and poses are visualized.



Figure 6.7 Part Similarity Visualization. Top shows the same subject pairs. Bottom shows different subject pairs. Part features provide some indication of where the similar parts are, but the final similarity is generated through a nonlinear mapping of the part features.

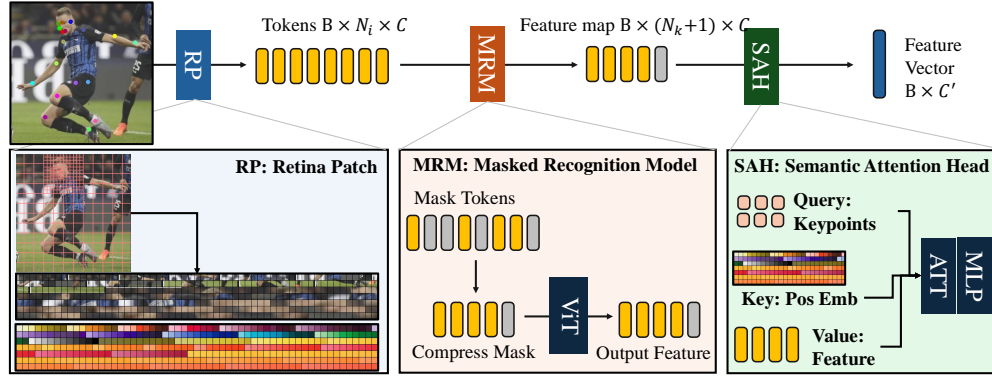


Figure 6.8 Illustration of the feature vector generation in SapiensID. First, Retina Patch (RP) generates image patches. Then, Masked Recognition Model (MRM) modifies the number of tokens. Finally, Semantic Attention Head (SAH) produces the feature vector from the set of tokens.

## 6.5 Method Details

### 6.5.1 Training Details

The training pipeline of SapiensID is largely similar to the setting of training a ViT model in face recognition [125]. This is possible because WebBody4M is a labeled dataset with a sufficient number of subjects, just as face recognition datasets. We use the AdaFace [122] loss

and optimize the model with the AdamW [164] optimizer for 33 epochs. The learning rate is scheduled by the Cosine Annealing Learning Rate Scheduler [163] with an additional warm-up period of 3 epochs. The maximum learning rate is set to 0.0001. We use 7 A100 GPUs with a batch size of 128. We also change the classifier to PartialFC [11] with a sampling ratio of 0.1 to save GPU memory and gain computation efficiency. Overview of the model is shown in Fig. 6.8.

For data augmentation, we find that it is important to use a moderate amount of geometric augmentation (zoom in-out:  $0.9 \sim 1.1$ , translation:  $\pm 0.05$ ) and aspect ratio adjustments ( $0.95 \sim 1.05$ ). We also find it effective for improving aligned face recognition performance to include face-zoomed-in images frequently (40%). We also oversample images that contain more visible keypoints because those images are relatively scarce (note Tab. 6.12).

### 6.5.2 Notation Clarification in the Main Paper

In Semantic Attention Pooling’s SAH, the equation presented as Eq. 6.8:

$$\mathbf{O}_{\text{part}}^i = \text{Attention}(\mathbf{Q}_{kp}^i, \text{PE}, \text{backbone}(\mathbf{X}^i)), \quad (6.11)$$

$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V})$  is specifically defined as:

$$\mathbf{O}_{\text{part}}^i = \text{softmax}\left(\frac{\mathbf{W}_q \mathbf{Q} \mathbf{W}_k \mathbf{K}^\top}{\sqrt{d}}\right) \mathbf{W}_v \mathbf{V}, \quad (6.12)$$

where  $\mathbf{Q}$ ,  $\mathbf{K}$ , and  $\mathbf{V}$  represent the query, key, and value matrices, respectively, and  $\mathbf{W}_q$ ,  $\mathbf{W}_k$ , and  $\mathbf{W}_v$  are their associated projection weights. This is how the size of the attention is modulated during learning.

Also notice that without the learnable projections  $\mathbf{W}_{q,k,v}$  and a small  $d$ , the attention simply focuses on the position with the highest proximity to the keypoint. To make sure that we have this feature from the sharp peak at the keypoint location, we additionally use

$$\mathbf{O}_{\text{peak}}^i = \text{softmax}\left(\frac{\mathbf{Q} \mathbf{K}^\top}{\sqrt{d}}\right) \mathbf{V}. \quad (6.13)$$

The final feature vector is computed by concatenating the two sets of semantic features  $\mathbf{O}_{\text{part}}^i$  and  $\mathbf{O}_{\text{peak}}^i$  and flattening them for MLP projection. Specifically, it is

$$f^i = \text{MLP}(\text{flatten}([\mathbf{O}_{\text{part}}^i, \mathbf{O}_{\text{peak}}^i])). \quad (6.14)$$

The addition of  $\mathbf{O}_{\text{peak}}^i$  is simply to ensure that the model always has the feature from the keypoint location. We have not tested how much performance gap is created by removing this inductive bias in SAH. The final number of part features is 152 (19 keypoints  $\times$  4 offset repeats  $\times$  2 from concatenating  $\mathbf{O}_{\text{part}}^i$  and  $\mathbf{O}_{\text{peak}}^i$ ). We realize that the readers could be confused about the formulation of SAH attention, so we will make it clearer in the main paper.

### 6.5.3 Things We Tried That Did Not Make it into the Main Algorithm

- We tried to initialize the model with the Sapiens [118] pretrained backbone, thinking it would be a good starting point that leads to better generalization. However, it did not lead to better performance. We believe this is because: 1) our patch scheme is dramatically different from the original patch scheme, and 2) Sapiens is trained with the MAE [85] objective, which is suitable for dense prediction tasks. However, SapiensID is a classification (or metric learning) task. Dense prediction tasks prioritize spatial consistency and detailed reconstruction, whereas classification tasks focus on extracting discriminative features, which may require different feature representations.
- We tried using the differential layerwise learning rate [270], but it did not help and the learning was only slower.
- We tried not learning the size and offset for the Semantic Attention Head (SAH) by simply taking the feature from the keypoint locations. This led to worse performance in general.

### 6.5.4 Transforming Keypoints to ROIs

SapiensID relies on predicted keypoints to define Regions of Interest (ROIs). Assuming we have an input image roughly cropped around the visible body area (typically using a person detector’s bounding box), we start with a set of predicted keypoints  $\mathbf{K} = \{(x_k, y_k)\}_{k=1}^N$ , where  $N$  is the number of keypoints. Our goal is to generate bounding boxes for each ROI. Specifically, we generate two bounding boxes—for the face and the upper torso—in the format  $(x_1, y_1, x_2, y_2)$ , representing the top-left and bottom-right corners.

## 1. Valid Keypoint Selection:

Let  $\mathcal{K} = \{1, 2, \dots, N\}$  be the set of keypoint indices. For each keypoint  $k \in \mathcal{K}$ , the coordinates are  $(x_k, y_k) \in \mathbb{R}^2$ . We define a visibility indicator  $v_k$  for each keypoint:

$$v_k = \begin{cases} 1, & \text{if } x_k \neq -1 \text{ and } y_k \neq -1, \\ 0, & \text{otherwise.} \end{cases} \quad (6.15)$$

Define the sets of keypoint indices relevant to each ROI:

$K_1$ : Left Eye      $K_6$ : Left Mouth Corner  
 $K_2$ : Right Eye     $K_7$ : Right Mouth Corner  
 $K_3$ : Left Ear      $K_8$ : Left Shoulder  
 $K_4$ : Right Ear     $K_9$ : Right Shoulder  
 $K_5$ : Nose

Then Face Keypoints are

$$\mathcal{M}_f = \{K_1, K_2, K_3, K_4, K_5, K_6, K_7\}.$$

And Upper Torso Keypoints are

$$\mathcal{M}_u = \mathcal{M}_f \cup \{K_8, K_9, K_{10}, K_{11}\}.$$

The valid keypoints for each ROI are those that are both visible and relevant:

$$\mathcal{V}^{\text{face}} = \{k \in \mathcal{M}_f \mid v_k = 1\}, \quad (6.16)$$

$$\mathcal{V}^{\text{torso}} = \{k \in \mathcal{M}_u \mid v_k = 1\}. \quad (6.17)$$

## 2. Bounding Box Center and Size Calculation:

For each ROI (face or upper torso), we compute the center using the set  $\mathcal{V}$ , which is either  $\mathcal{V}^{\text{face}}$  or  $\mathcal{V}^{\text{torso}}$ :

First compute the minimum and maximum coordinates among valid keypoints:

$$x_{\min} = \min_{k \in \mathcal{V}} x_k, \quad y_{\min} = \min_{k \in \mathcal{V}} y_k, \quad (6.18)$$

$$x_{\max} = \max_{k \in \mathcal{V}} x_k, \quad y_{\max} = \max_{k \in \mathcal{V}} y_k. \quad (6.19)$$

Then calculate the center of the bounding box:

$$c_x = \frac{x_{\min} + x_{\max}}{2}, \quad c_y = \frac{y_{\min} + y_{\max}}{2}. \quad (6.20)$$

Then determine the maximum distance  $d$  from the center to the valid keypoints:

$$d = \max_{k \in \mathcal{V}} \sqrt{(x_k - c_x)^2 + (y_k - c_y)^2}. \quad (6.21)$$

### 3. Bounding Box with Padding:

First define the bounding box size  $s$  with a padding factor  $p$  (e.g.,  $p = 0.3$ ):

$$s = d \times (1 + p). \quad (6.22)$$

Then calculate the coordinates of the bounding box:

$$x_1 = c_x - s, \quad y_1 = c_y - s, \quad (6.23)$$

$$x_2 = c_x + s, \quad y_2 = c_y + s. \quad (6.24)$$

**4. Making Bounding Box Divisible:** To ensure that the patches cover the image without any overlap, the boundaries of the bounding box must *snap* onto the patch grid. In other words, the bounding box coordinate should be divisible by the patch size  $(p_w, p_h)$  of the enclosing ROI. Let  $n_r$  and  $n_c$  be the desired number of rows and columns for patches within the ROI. We modify the bounding box size  $s$  to ensure divisibility.

$$x'_1 = \lfloor \frac{x_1}{p_w} \rfloor \times p_w, \quad y'_1 = \lfloor \frac{y_1}{p_h} \rfloor \times p_h \quad (6.25)$$

$$x'_2 = \lceil \frac{x_2}{p_w} \rceil \times p_w, \quad y'_2 = \lceil \frac{y_2}{p_h} \rceil \times p_h \quad (6.26)$$

The final, grid-aligned bounding box is then:

$$\mathbf{b} = (x'_1, y'_1, x'_2, y'_2) \in \mathbb{R}^4. \quad (6.27)$$

This snapping process ensures that the bounding box boundaries coincide with patch boundaries, resulting in clean, non-overlapping patch extraction. We compute two bounding boxes,  $\mathbf{b}^{\text{face}}$  and  $\mathbf{b}^{\text{torso}}$ , using this process. All these steps can be conducted in GPU for efficient computation.

### 6.5.5 Proof of Scaled Attention Equivalence

Let the scaled dot-product attention mechanism for self attention is defined as:

$$A = \text{softmax} \left( \frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d}} \right) \mathbf{V},$$

We aim to prove that when a scaling factor  $\boldsymbol{\delta} \in \mathbb{R}^{1 \times M}$  is added to the logits:

$$A = \text{softmax} \left( \frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d}} + \boldsymbol{\delta} \right) \mathbf{V},$$

this is equivalent to repeating each key  $\mathbf{K}_j$  and value  $\mathbf{V}_j$  exactly  $m_j$  times, where  $\delta_j = \log m_j$ .

**Proof:** Consider the following term:

$$\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d}} + \boldsymbol{\delta}.$$

For a query  $i$  and key  $j$ , the element of this matrix is:

$$\left( \frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d}} + \boldsymbol{\delta} \right)_{ij} = \frac{\mathbf{Q}_i \cdot \mathbf{K}_j^\top}{\sqrt{d}} + \log m_j,$$

where  $\mathbf{Q}_i$  is the  $i$ -th query and  $\mathbf{K}_j$  is the  $j$ -th key. Applying the softmax function, we get:

$$A_{ij} = \frac{\exp \left( \frac{\mathbf{Q}_i \cdot \mathbf{K}_j^\top}{\sqrt{d}} + \log m_j \right)}{\sum_k \exp \left( \frac{\mathbf{Q}_i \cdot \mathbf{K}_k^\top}{\sqrt{d}} + \log m_k \right)}.$$

Using the property  $\exp(a + b) = \exp(a) \exp(b)$ , this simplifies to:

$$A_{ij} = \frac{\exp \left( \frac{\mathbf{Q}_i \cdot \mathbf{K}_j^\top}{\sqrt{d}} \right) m_j}{\sum_k \exp \left( \frac{\mathbf{Q}_i \cdot \mathbf{K}_k^\top}{\sqrt{d}} \right) m_k}.$$

Dataset	Avg	LFW	CPLFW	CFPFP	CALFW	AGEDB
WF4M	97.44	99.80	94.97	98.94	96.03	97.48
WB4M-Facecrop	<b>97.63</b>	<b>99.82</b>	<b>95.12</b>	<b>99.19</b>	<b>96.07</b>	<b>97.97</b>

Table 6.7 Performance Comparison between WebFace4M and WebBody4M in the Face Recognition Task.

This is equivalent to each key  $\mathbf{K}_j$  and corresponding value  $\mathbf{V}_j$  are duplicated  $m_j$  times. We discard the values corresponding to the mask, so the result of the attention mechanism is the same. Thus, the attention mechanism with  $\delta$  scaling is mathematically equivalent to duplicating the keys and values proportionally to the number of times the mask appears.

## 6.6 Performance

### 6.6.1 WebBody4M vs WebFace4M Comparison

To assess the quality of the face image data within WebBody4M, we create WebBody-Facecrop by cropping face from the WebBody dataset. And we compare its face recognition performance against WebFace4M [300], a dedicated large-scale face recognition dataset. We train the same ViT-based model with AdaFace loss on both datasets. Tab. 6.7 presents the results on standard face recognition benchmarks (LFW, CPLFW, CFPFP, CALFW, and AGEDB). The model trained on WebBody4M achieves a slightly higher average accuracy (97.63%) compared to that of WebFace4M (97.44%). This indicates WebBody4M label is of comparable quality, even slightly exceeding WebFace4M label.

### 6.6.2 Fusion Performance

While SapiensID inherently handles both face and body information within a single model, a common alternative approach involves training separate face and body recognition models and fusing their outputs. We compare SapiensID’s performance with such multi-modal fusion methods. We consider a baseline where a body model (CAL [80]) is trained on either PRCC or LTCC, and a face model (ViT-Base [122]) is trained on WebFace4M. We then fuse the similarity scores of these two dedicated face and body models using three common fusion strategies: Max Fusion, Min-Max Normalization Fusion, and Mean Fusion. Tab. 6.8 presents the performance.

	AVG	LTCC CC Top1 mAP	PRCC CC Top1 mAP
Body	42.04	38.01	55.69
Face	36.56	17.60	72.62
Fused-Max	42.93	39.80	61.22
Fused Min-Max	49.92	39.80	79.00
Fused-Mean	49.99	39.80	<b>79.48</b>
SapiensID	<b>52.87</b>	<b>42.35</b>	<b>78.75</b>

Table 6.8 Performance table of score fusion (Body and Face).

Method Training Data	KPR [214] + SOLDIER		SapiensID
	LUPerson4M + OccludedReID		WebBody4M
OccludedReID	top1	84.80	<b>87.30</b>
	mAP	<b>82.60</b>	75.57
LTCC General	top1	68.15	<b>74.24</b>
	mAP	32.42	<b>36.88</b>
LTCC CC	top1	21.17	<b>42.60</b>
	mAP	10.19	<b>17.39</b>

Table 6.9 Generalization performance comparison under occlusion. SapiensID demonstrates superior generalization to unseen datasets (LTCC) compared to KPR+SOLDIER.

As shown in the table, even the best fusion strategy (Mean Fusion) achieves an average mAP of 49.99%, lower than SapiensID’s 52.87%. Fusion is more helpful in PRCC but not much in LTCC with an increase in Top1 and a decrease in mAP. This result highlights the advantage of SapiensID’s unified architecture, which learns to integrate face and body information more effectively than post-hoc fusion methods. Fusion methods treat each modality independently, potentially missing valuable contextual information that arises from their combined analysis.

### 6.6.3 Occluded ReID

Occlusions pose a significant challenge for robust human recognition. While specialized methods can be effective within their training domain, generalization to unseen scenarios is crucial for real-world deployment. We compare SapiensID’s performance with KPR [214] combined with SOLDIER, a state-of-the-art occlusion handling method, to evaluate their respective generalization capabilities. KPR+SOLDIER is trained on a combination of LUPerson4M and the OccludedReID [301] dataset, while SapiensID is trained on our WebBody4M dataset without any OccludedReID data.

Tab. 6.9 presents the results on OccludedReID and the LTCC dataset (both General

		LTCC CC		PRCC CC				LTCC CC		PRCC CC	
		Top1	mAP	Top1	mAP			Top1	mAP	Top1	mAP
1	None	0.00	3.56	1.47	4.28	1	None	0.00	3.56	1.47	4.28
2	1+Nose	25.77	5.78	27.21	21.04	2	1+Ankle	27.04	7.37	45.05	35.32
3	2+Eye	30.61	8.87	63.87	55.17	3	2+Knee	32.14	9.55	55.12	44.97
4	3+Mouth	38.01	11.81	73.36	65.05	4	3+Hip	35.71	12.34	66.07	55.04
5	4+Ear	39.80	14.05	77.65	70.45	5	4+Wrist	37.24	13.83	67.63	58.43
6	5+Shoulder	41.84	15.82	79.73	73.14	6	5+Elbow	40.05	15.72	69.57	62.61
7	6+Elbow	41.07	16.64	<b>80.55</b>	<b>73.54</b>	7	6+Shoulder	41.33	16.87	73.84	67.80
8	7+Wrist	41.07	17.16	79.34	73.16	8	7+Ear	41.58	17.61	76.21	70.62
9	8+Hip	40.56	17.50	79.99	73.38	9	8+Mouth	41.58	17.95	78.18	72.63
10	9+Knee	42.35	17.73	79.00	72.88	10	9+Eye	41.58	<b>17.80</b>	<b>79.23</b>	<b>72.92</b>
11	10+Ankle (full)	<b>42.35</b>	<b>17.79</b>	78.75	72.60	11	10+Nose (Full)	<b>42.35</b>	17.79	78.75	72.60

(a) top-down

(b) bottom-up

Table 6.10 Comparison of feature erasing performance. (a) shows the performance as we progressively introduce features from Nose to Ankle (top-down approach). (b) demonstrates the performance when adding features from Ankle to Nose (bottom-up approach). Results are evaluated on LTCC and PRCC Cloth Changing (CC) protocol.

		LTCC CC		PRCC CC				LTCC CC		PRCC CC	
		Top1	mAP	Top1	mAP			Top1	mAP	Top1	mAP
1	None	2.30	1.89	12.67	4.78	1	None	2.30	1.87	12.50	4.78
2	1+Top1	5.10	2.61	78.04	67.29	2	1+Bottom1	2.81	2.26	24.56	10.89
3	2+Top2	27.04	11.88	79.25	70.53	3	2+Bottom2	6.12	3.08	31.22	16.94
4	3+Top3	29.34	13.20	78.35	69.85	4	3+Bottom3	5.87	3.62	33.78	20.65
5	4+Top4	33.67	13.88	77.82	69.55	5	4+Bottom4	10.20	4.26	33.08	24.59
6	5+Top5	37.24	14.65	76.97	69.28	6	5+Bottom5	12.50	5.33	22.10	21.31
7	6+Top6	36.48	15.49	78.55	70.39	7	6+Bottom6	16.07	6.48	24.47	24.80
8	7+Top7	41.07	16.63	<b>80.07</b>	<b>71.52</b>	8	7+Bottom7	35.46	13.20	29.07	28.63
9	Full	<b>42.35</b>	<b>17.79</b>	78.75	72.60	9	Full	<b>42.35</b>	<b>17.79</b>	<b>78.75</b>	<b>72.60</b>

(a) top-add

(b) bottom-add

Table 6.11 Impact of progressively adding visible parts from the (a) top and from the (b) bottom. In contrast to Tab. 6.10 which measures the performance with the intermediate features zeroed out, here the actual input image is masked out.

and Clothing Change protocols). KPR+SOLDIER and SapiensID similar performance on OccludedReID, SapiensID demonstrates significantly better generalization performance. On LTCC, SapiensID substantially outperforms KPR+SOLDIER across both protocols, highlighting the limitations of specialized training. This underscores the importance of training on diverse datasets like WebBody4M to achieve robust generalization in real-world human recognition. SapiensID, by learning from a wide range of poses, viewpoints, and clothing styles, is more adaptable and effective in unseen scenarios.

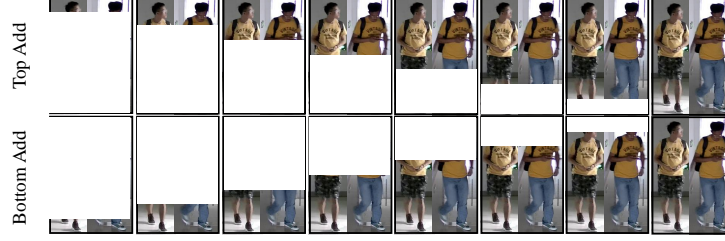


Figure 6.9 Illustration of how Images are erased from top to bottom or bottom to top.

#### 6.6.4 Impact of Body Part Features

We investigate the relative importance of different body parts in human recognition by conducting an ablation study on the Semantic Attention Head (SAH). Starting from part features ( $\mathbf{O}_{part}^i$  in Eq. 6.8) multiplied by zero, we progressively undo masking, either from nose-to-ankles (top-down) or ankles-to-nose (bottom-up). We evaluate performance on LTCC (Clothing Change protocol) and PRCC (Clothing Change protocol). Results are presented side-by-side in Tab. 6.10. The top-down approach generally yields faster performance gains than bottom-up, suggesting that upper-body features contribute more significantly to recognition.

Interestingly, ankle features alone appear more discriminative than nose features alone. However, this counter-intuitive finding does not imply that ankles are inherently more informative than noses for person identification. We hypothesize that this observation arises because each part feature within SAH is not solely derived from the corresponding body part. Due to the preceding ViT backbone’s attention mechanism, each part feature incorporates information from other body regions. Therefore, the presented results reflect the discriminative power of a part plus peripheral information from other parts, rather than the isolated contribution of each part.

A more accurate assessment of a part’s individual discriminative ability would involve manipulating the input image directly, such as by occluding specific body parts. This approach, which isolates the impact of each part, is explored in the following section.

### 6.6.5 Impact of Actual Image Erased

To isolate the contribution of each body region, we conduct a second ablation study where we progressively erase sections of the input image, either top-down or bottom-up, as illustrated in Fig. 6.9. We erase equal-sized horizontal strips, starting with a single strip and progressively adding more until the whole image is erased (represented as "None" in the tables). The "Full" row represents the baseline performance with the complete image. Results are presented in Tab. 6.11.

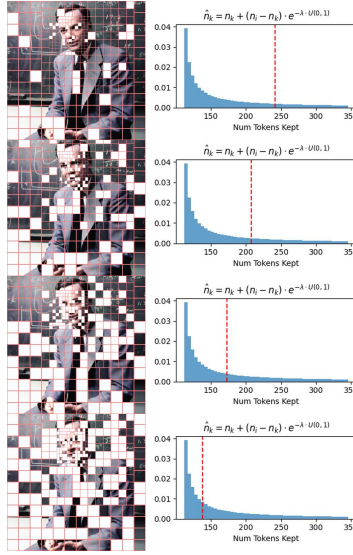


Figure 6.10 Illustration of the masked image and the sampling distribution of the number of tokens to keep  $\hat{n}_k$ . The red vertical line shows where the sampling took place for the right image. From top to bottom, less samples are kept (more masking).

The direct manipulation of the image confirms the importance of upper body regions. On both datasets, removing the top portion of the image drastically reduces performance. It comes as a surprise that PRCC can achieve a very good performance with only 1 top strip of image. But for LTCC, the lower parts are necessary to obtain a good performance. This indicates that different datasets exhibit different characteristics that can be exploited for conducting ReID.

## 6.7 Visualization

### 6.7.1 Token Length Sampling Distribution

In Masked Recognition Model (MRM), we propose an adaptive token sampling strategy during training to enhance the robustness and generalization of our masked recognition model. Fig. 6.10 illustrates the sampling distribution and its effect on the input image. The number of tokens to keep,  $\hat{n}_k$ , is determined by Eqn. 6.6:

$$\hat{n}_k = n_k + (n_i - n_k) \cdot e^{-\lambda \cdot U(0,1)},$$

where  $n_i$  is the maximum possible number of tokens (432 in our case, with 3 ROIs of 12x12 patches each),  $n_k$  is the minimum number of tokens to keep,  $U(0,1)$  is a uniform random variable, and  $\lambda$  controls the decay rate (set to 4).

This sampling strategy allows us to retain between 26% and 80% of the tokens (112 to 345 tokens), with an average of 166 tokens per batch. As depicted in Fig. 6.10, heavy masking can significantly distort the input image. Fixing the masking rate to such high levels could introduce a distribution shift between training and testing (where all tokens are used), causing a performance drop. Our adaptive sampling mitigates this issue by exposing the model to a variety of masking ratios, encouraging it to learn robust representations that generalize well to full token input during inference.

One thing to note is that the sampling of  $\hat{n}_k$  happens per batch. And when a larger  $\hat{n}_k$  is sampled per batch, we reduce the batch size accordingly for the given GPU memory (See Sec. 6.3.2 for more details).

### 6.7.2 WebBody4M Dataset Body Parts Visibility

WebBody4M dataset encompasses a wide range of human poses and viewpoints, resulting in varying visibility of body keypoints. Tab. 6.12 presents the percentage of images in which each keypoint (left and right sides) is visible. As expected, keypoints in the upper body, such as eyes and shoulders, exhibit high visibility rates (over 74% and 88% respectively). Visibility decreases progressively down the body, with elbows and wrists around 50%, hips around

Visibility	Left (%)	Right (%)
Eye	93.49	93.59
Ear	76.87	74.48
Shoulder	88.15	90.04
Elbow	53.76	53.80
Wrist	49.98	50.35
Hip	45.68	45.70
Knee	23.92	23.95
Ankle	16.98	17.00

Table 6.12 Keypoint Visibility in WebBody Dataset.

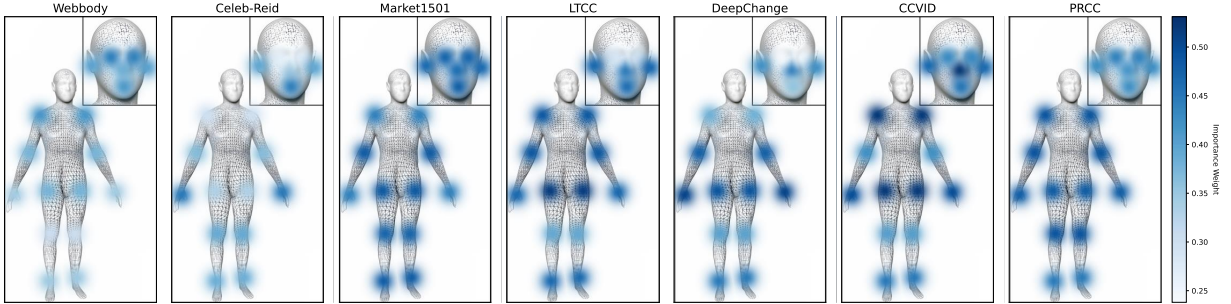


Figure 6.11 Comparison of learned part weights across seven datasets. Left and right sides are averaged together before visualization.

45%, and knees and ankles below 24% and 17% respectively. This distribution reflects the natural tendency for upper body parts to be more frequently visible in unconstrained images, as lower body parts are often occluded by clothing, objects, or the image frame itself. This distribution also helps explain why upper body parts provide greater discriminative power for person ReID in our earlier analysis (Supp 6.6.4).

### 6.7.3 Visualization of Part Weights

To facilitate effective learning from a mixture of short-term and long-term ReID datasets, we hypothesize that it would be helpful to add learnable weights that modulate the importance of individual part features within the Semantic Attention Head (SAH). Our conjecture is the discriminative characteristics of body parts can vary significantly depending on whether clothing remains constant or varying in the training dataset.

Fig. 6.11 visualizes the learned weights (Eqn. 6.14) for WebBody4M and several additional whole-body ReID datasets. WebBody4M, primarily composed of web-collected images, exhibits a higher emphasis on facial features compared to lower body parts. This is expected,

as the WebBody4M was collected largely based on facial similarity.

In contrast to WebBody4M, auxiliary datasets like Market1501, LTCC, and PRCC, which feature many images with consistent clothing (e.g., 1-3 outfits across 20-30 images per person), show increased emphasis on body features for recognition. This highlights the importance of body shape, pose, and clothing appearance as discriminative cues when attire remains relatively constant. However, Celeb-ReID, similar to WebBody4M, primarily contains images with clothing changes across captures. Consequently, Celeb-ReID exhibits a similar weighting pattern, with less emphasis on body features and a relatively higher focus on other cues, likely emphasizing facial features.

To validate the hypothesis, we conducted an ablation study to evaluate the impact of training with learnable weights. Tab. 6.13 presents a comparison between SapiensID and SapiensID without the learnable weights. In the latter, all aspects remain the same except that the learnable weights are removed during training.

From the results, it is evident that the inclusion of learnable weights does not yield a significant overall improvement. Instead, it shows a specific enhancement in long-term ReID performance, possibly because WebBody4M’s learning was not hindered by the influence of short-term datasets with same clothings. However, for short-term datasets, the addition of weights does not result in performance gains. This suggests that while the weighting mechanism provides insights into dataset-specific learning behaviors, it is not a definitive factor for achieving better ReID performance.

In conclusion, while the introduction of learnable weights is interesting for analytical purposes, we want to let the readers clearly know that it is not a deciding factor for learning universal representation that works for both short-term and long-term ReID. Future research could explore alternative methods that better balance the learning from diverse dataset characteristics without negatively impacting specific subsets.

	All	Face	Whole Body ReID	Short Long
SapiensID	<b>78.67</b>	<b>96.66</b>	73.05	<b>66.30</b>
SapiensID-Weight	78.59	<b>96.66</b>	<b>75.72</b>	63.39

Table 6.13 Performance comparison of SapiensID and SapiensID without weight masking during training across different metrics.

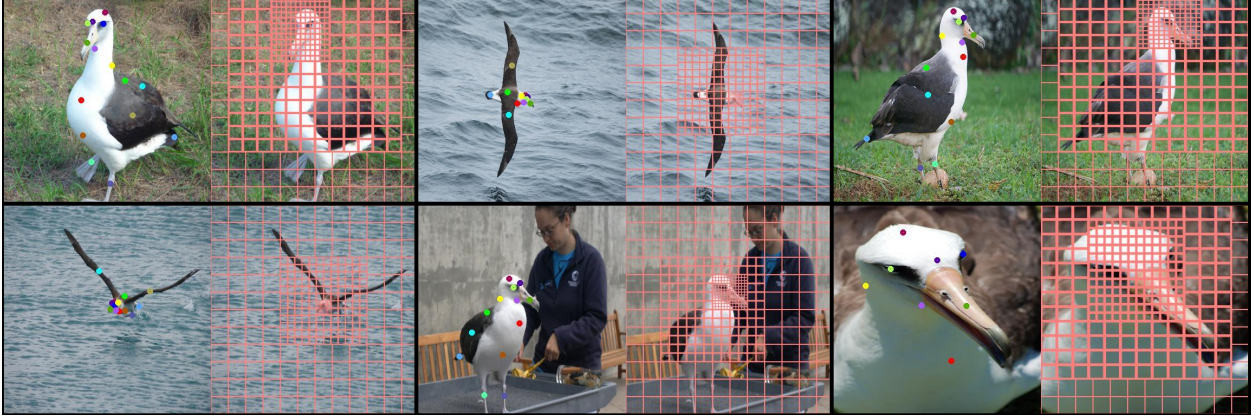


Figure 6.12 Keypoint visualization (left) and corresponding Retina Patch results (right) for images from the CUB dataset.

#### 6.7.4 SAH Visualization

The Semantic Attention Head (SAH) plays a crucial role in SapiensID by generating pose-invariant features. To understand how SAH behaves after training, we visualize its attention maps in Fig. 6.13. To be specific, we visualize the following. Let  $\mathbf{Q}_{kp}^i = \text{GridSample}(\text{PE}, \text{kp}^i) + \mathbf{B}$  be the semantic query embedding for  $i$ -th image created by sampling from the fixed 2D position embeddings (PE) at the 19 keypoint locations. The dimension is  $\mathbf{Q}_{kp}^i \in \mathbb{R}^{nk \times C}$ , where  $k = 19$  and  $n = 4$  because it is repeated 4 times to learn 4 different offsets. In SAH, we perform attention with  $\mathbf{Q}_{kp}^i$  and PE by

$$\mathbf{O}_{\text{part}}^i = \text{softmax} \left( \frac{\mathbf{W}_q \mathbf{Q} \mathbf{W}_k \mathbf{K}^\top}{\sqrt{d}} \right) \mathbf{W}_v \mathbf{V}. \quad (6.28)$$

In our visualization, we are showing

$$\text{softmax} \left( \frac{\mathbf{W}_q \mathbf{Q} \mathbf{W}_k \mathbf{K}^\top}{\sqrt{d}} \right),$$

for each keypoint and each offset. We have  $nk$  attention maps as shown by the visualization.

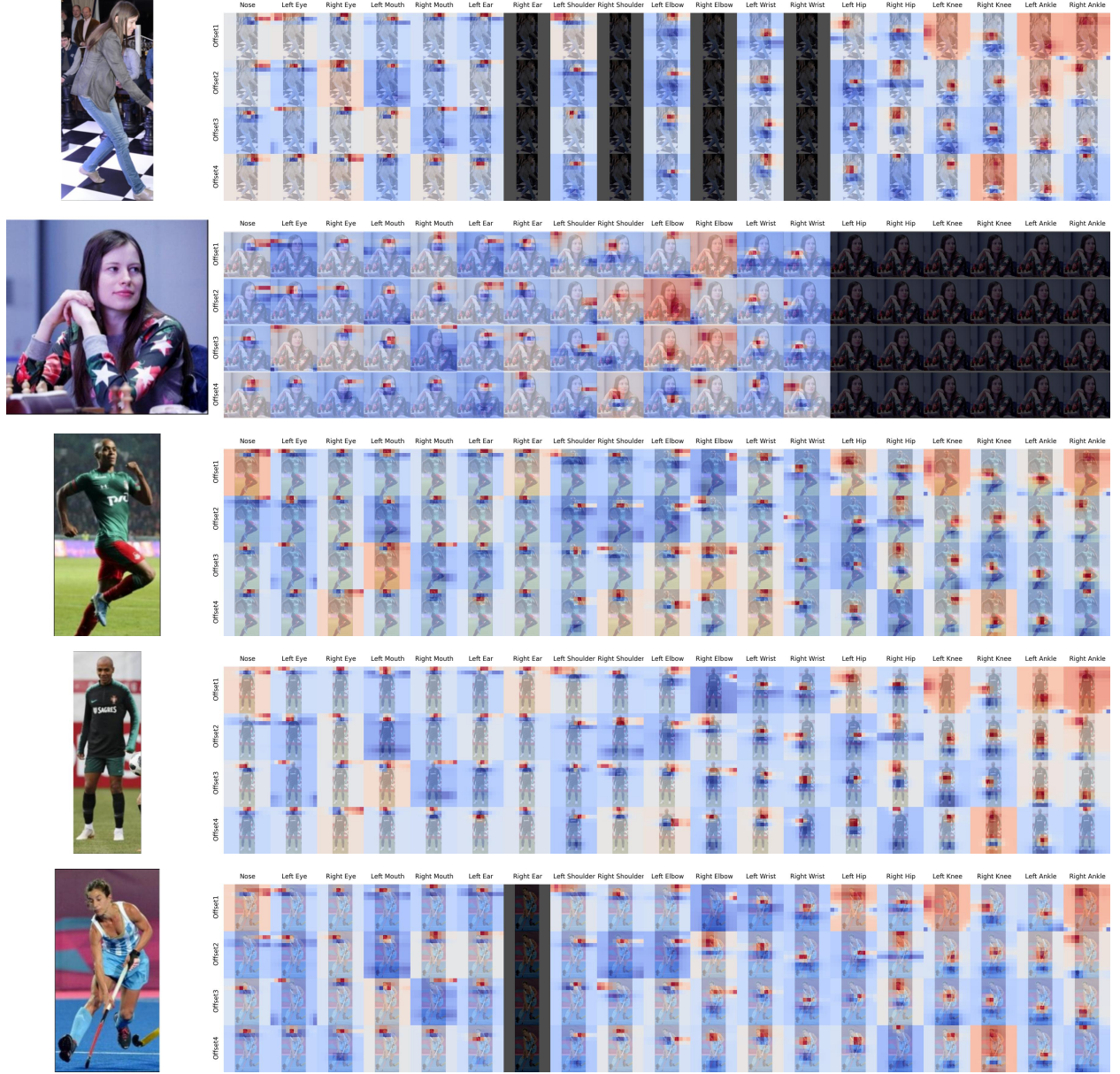


Figure 6.13 Visualization of attention maps in the Semantic Attention Head (SAH). Regions with higher attention values are highlighted in red, while regions with lower attention values are shown in blue. Blacked-out areas represent parts of the images without visible keypoints. The visualizations provides how SAH allows learning both varied size and offsets based on a set of keypoints.

For each input image, we show each row corresponds to a different offset. There are 4 rows because we learn  $n = 4$  offsets for each of 19 keypoints. Offset refers to  $\mathbf{B} \in \mathbb{R}^{nk \times C}$  in Eqn. 6.7. Offset bias allows the keypoints to move slightly from its original position. Each

column correspond to different keypoints used by SAH (e.g., nose, left right shoulder, *etc*). As the visualization shows, the learned attention maps are not limited to the keypoint location but also move around the keypoints and vary in size.

## 6.8 Potential Application of Retina Patch

While SapiensID focuses on human recognition, the Retina Patch (RP) mechanism has broader applicability to other domains. Figure 6.12 demonstrates its potential for fine-grained visual recognition, using the CUB birds dataset as an example. This dataset provides semantic keypoints, enabling the definition of meaningful regions of interest (ROIs) for RP. We define two ROIs: "head" (beak, forehead, crown, left eye, right eye, throat) and "body" (back, belly, breast, nape, left wing, right wing) excluding tail, left leg and right leg.

The figure showcases multiple bird images processed with RP, illustrating its ability to handle variations in bird size and head size. By dynamically allocating more patches to these regions, RP ensures consistent representation of crucial features, regardless of their scale within the image. Though we do not know whether the performance of CUB bird classification will be improved with RP, we want to suggest that RP could be beneficial for general recognition tasks where image naturally contains large pose and scale variation. Future work could explore the integration of RP into models for more broad set of datasets to quantitatively evaluate its benefits.

## 6.9 Limitations

While SapiensID demonstrates promising results for human recognition, its reliance on predefined Regions of Interest (ROIs) introduces certain limitations. The effectiveness of the Retina Patch mechanism hinges on the ability to define meaningful ROIs that capture discriminative features. This approach works well for humans, who share a consistent body topology and where keypoints like the face, torso, and limbs provide valuable cues for recognition.

However, this reliance on ROIs poses challenges when dealing with objects or entities that lack a consistent or well-defined structure. For instance, applying SapiensID to amorphous

objects, scenes with highly variable elements, or categories with significant intra-class topological differences would require alternative strategies. In such cases, predefined ROIs might not adequately capture the relevant information, or might even be detrimental by focusing on irrelevant or inconsistent features. Future research could explore more flexible or adaptive mechanisms for defining regions of interest, enabling the application of similar principles to a wider range of object recognition tasks.

### **6.10 Ethical Concerns**

Our goal is to facilitate research in human recognition while operating strictly within the bounds of copyright law, privacy regulations, and ethical considerations. For large-scale image datasets, it is a common practice to release datasets in URL format [18, 201] because researchers do not hold the rights to redistribute the data directly. By providing permanent link URLs, labels and a one step code to download and prepare dataset, researchers can have access and utilize the data responsibly, while respecting the rights of copyright holders and individuals. We believe this approach balances the need for large-scale datasets to advance research with the imperative to protect intellectual property and privacy.

### **6.11 Conclusion**

SapiensID presents a paradigm shift in human recognition, moving beyond modality-specific models to a unified architecture capable of identification across diverse poses and body-part scales. Retina Patch, Semantic Attention Head, and Masked Recognition Model combined with WebBody4M dataset, enable SapiensID to achieve SoTA performance across various ReID benchmarks and establish a strong baseline for Cross Pose-Scale ReID. This work marks a step towards holistic human recognition systems.

## CHAPTER 7

### EFFICIENT HUMAN RECOGNITION FRAMEWORK

While unified face and body recognition models offer enhanced robustness across diverse poses and scales, their reliance on Vision Transformers (ViTs) processing numerous tokens often leads to prohibitive computational costs, hindering practical deployment in real-time applications. This paper introduces a novel approach to significantly improve the efficiency of unified biometric recognition ViTs without compromising accuracy. We propose Keypoint-based Token Fusion (KP-ToFu), a heuristic token reduction strategy specifically designed for biometrics, which merges less informative tokens while strategically preserving those corresponding to crucial human keypoints essential for identification. To maintain spatial reasoning capabilities after the token structure is altered by fusion, we develop Keypoint Absolute Position Encoding (KP-APE). Additionally, we introduce Reasoning Tokens, progressively added learnable tokens that compensate for the reduced input token count and enhance the model’s representational capacity for complex identity reasoning. Our synergistic approach, combining KP-ToFu, KP-APE, and Reasoning Tokens, achieves state-of-the-art performance on challenging joint face and body recognition benchmarks while providing substantial computational speed-ups. We further demonstrate the versatility of our efficient backbone by successfully adapting it to gait recognition. This work paves the way for fast, accurate, and deployable unified human recognition systems.

#### 7.1 Introduction

Human recognition remains a fundamental challenge in computer vision, crucial for applications ranging from security surveillance to personalized user experiences. Historically, this task has been tackled using disparate approaches: highly specialized models for face recognition [55, 101, 102, 122–124, 128, 154, 239, 240, 252, 276] and separate models for body-based person re-identification (ReID) [80, 110, 140, 149, 151, 268]. While successful in constrained environments relying on specific alignments [1, 54] or consistent camera views [212, 268], this fragmented strategy falls short in real-world settings. Practical scenarios often present

humans in diverse poses (sitting, standing, partial views) and scales, requiring systems to leverage both face and body cues opportunistically [112, 271]. The conventional solution involves fusing outputs from multiple models [87, 147], adding system complexity and potential failure points. A unified model, capable of processing the full spectrum of human appearance variations, promises greater robustness and simpler deployment.

Recent advancements, such as the SapiensID framework, have moved towards this unification using Vision Transformer (ViT) architectures. By processing multiple input resolutions (e.g., whole body, upper torso, face), SapiensID aims to achieve scale invariance. However, this approach comes at a steep price: computational cost. Feeding multiple high-resolution views into a ViT drastically increases the number of input tokens (e.g., 432), leading to significant computational overhead and slow inference speeds. It forms a critical bottleneck, rendering such powerful unified models impractical for many real-world biometric applications, including real-time video analysis, large-scale identity searches, and deployment on resource-constrained edge devices, where low latency and high throughput are paramount. Addressing this efficiency gap is therefore essential to unlock the potential of unified human recognition.

A promising direction for enhancing ViT efficiency is token fusion [27, 121], which reduces the computational load by dropping or averaging redundant or less informative tokens within the network. However, applying standard token fusion techniques directly to biometric recognition tasks presents a major challenge. Biometric identification relies heavily on preserving fine-grained details and the precise spatial arrangement of keypoints (e.g., facial landmarks, body joints). Naive token fusion, which merges tokens without considering their semantic importance, can inadvertently collapse these critical keypoint representations, severely degrading the model’s discriminative ability and undermining the core purpose of recognition.

To overcome this limitation while still reaping the benefits of token reduction, we first propose Keypoint-based Token Fusion (KP-ToFu). The core motivation is to achieve computa-

tional efficiency without sacrificing the crucial fine-grained information needed for biometrics. KP-ToFu intelligently identifies tokens corresponding to essential human keypoints and explicitly prevents them from being merged. Similar tokens are fused, significantly reducing the token count while ensuring that the structural integrity of the human form, vital for part-based matching, is preserved. This allows for substantial speed-ups while safeguarding recognition accuracy.

Secondly, the act of merging tokens fundamentally disrupts the regular grid structure inherent in the initial tokenization of the image. This disruption poses a significant problem because standard methods for incorporating spatial awareness in ViTs, such as 2D Relative Position Encoding (RPE) [89] or KP-RPE [125], rely on this grid structure to calculate positional relationships. Without effective positional encoding, the model loses vital information about where features are located relative to each other, further hindering recognition. To address this, we introduce efficient keypoint Absolute Position Encoding. KP-APE is specifically designed to calculate meaningful positional biases to keypoints even after tokens have been fused and their original grid coordinates are lost, thereby allowing the model to maintain spatial reasoning capabilities within the reduced and irregular token set.

Finally, while KP-ToFu preserves keypoints and KP-APE maintains spatial awareness, the overall reduction in token count via fusion might lead to a potential loss in the model’s representational capacity. The concern is that fewer tokens might limit the network’s ability to perform complex reasoning and integrate information across different parts of the input effectively. To counteract this, we introduce Reasoning Tokens. These are a small number of randomly initialized, learnable tokens that are progressively added into the ViT blocks alongside the image tokens. Similar to the [CLS] token, these reasoning tokens are not tied to specific spatial locations. They serve as adaptable computational resources, providing the network with additional capacity to synthesize features, model complex relationships, and perform higher-level reasoning about identity, compensating for the information density increase caused by token fusion.

By combining KP-ToFu, KP-APE, and Reasoning Tokens, we construct an efficient yet powerful backbone for unified human recognition. Our approach achieves state-of-the-art results on challenging benchmarks requiring joint face and body identification, demonstrating significant improvements in both computational efficiency and recognition accuracy. Furthermore, we showcase the versatility of our optimized backbone by successfully adapting it for efficient gait recognition through the addition of temporal attention mechanisms. This work paves the way for deploying robust, unified human recognition systems in demanding real-world applications where both accuracy and speed are critical.

## 7.2 Related Work

### 7.2.1 Biometric Recognition

The field of biometric recognition has traditionally operated with distinct silos for face recognition (FR) [55, 101, 102, 122–124, 128, 154, 239, 240, 252, 276] and body recognition (Person ReID) [80, 110, 140, 149, 151, 268]. While achieving high performance, these specialized models often depend on constrained inputs, such as aligned faces [1, 54] or canonical full-body poses [212, 268], limiting their utility in unconstrained real-world scenarios [112, 271]. Recent advancements, exemplified by models like SapiensID, have pioneered a unified approach, developing single models capable of jointly processing face and body information across varying scales and poses, often eliminating the need for strict pre-alignment [125]. This unification offers enhanced robustness and simplifies deployment pipelines.

However, this progress towards unification introduced a new, critical challenge: computational efficiency. Architectures like SapiensID, which employ Vision Transformers (ViTs) and process multiple input resolutions (e.g., face, upper-torso, full-body) to handle scale variance, inherently generate a very large number of tokens. This large token count leads to substantial computational demands and slow inference speeds, creating a significant barrier to deploying these powerful unified models in practical, real-time biometric systems where low latency is often crucial. The prohibitive cost associated with these initial unified models underscores the urgent need for methods that can preserve their recognition capabilities while drastically

improving their computational efficiency.

### 7.2.2 Token Reduction

The quadratic complexity of self-attention makes Vision Transformers (ViTs) computationally expensive, hindering their use in efficiency-sensitive applications like real-time biometrics and motivating token reduction strategies. Existing methods include learned approaches [8, 68, 142, 171, 182, 192, 275, 277] that prune tokens using auxiliary modules often requiring complex training, and simpler, training-free heuristic alternatives. Heuristic techniques like pooling [168], sampling [68], and Token Merging (ToMe) [28, 29]—which pioneered similarity-based training-free merging. Token Fusion [121] further explored strategies blending pruning and merging concepts.

However, a critical limitation of these heuristic methods for biometrics is their lack of semantic awareness. Merging tokens based solely on general feature similarity risks destroying the fine-grained details and spatial relationships of key anatomical points (e.g., eyes, joints) crucial for identification. Our work addresses this via Keypoint-based Token Fusion (KP-ToFu), an approach that explicitly preserves important keypoint tokens during fusion. We also propose KP-APE which is a modified version of KP-RPE [125] that can be applied to reduced token sets.

## 7.3 Proposed Work

Our goal is to develop an efficient backbone for unified human recognition, capable of processing diverse inputs containing faces and bodies across various scales and poses, while remaining computationally tractable for practical applications. We formulate the task as metric learning, aiming to produce discriminative embeddings where images of the same identity are closer than images of different identities, trained using a margin-based softmax loss [122]. The overall pipeline is shown in Fig 7.1.

### 7.3.1 Overview and Baseline Input Processing

Inspired by multi-region processing but simplified for efficiency, we handle scale variance by extracting the whole image, upper torso, and face regions (derived via keypoints [34]),

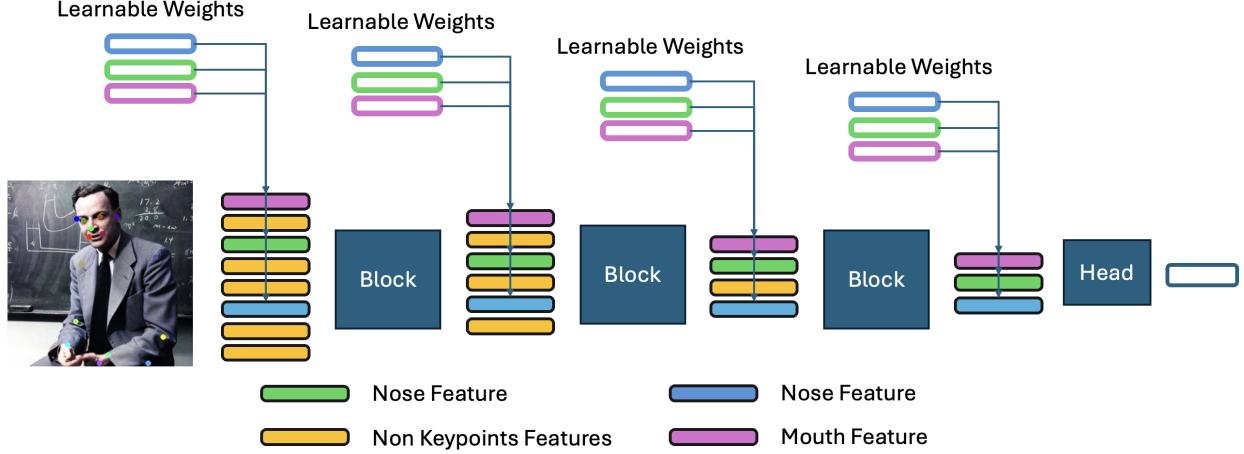


Figure 7.1 Overview of the Proposed Pipeline. Tokens are merge each layer while the keypoint tokens remain intact.

resizing each to a standard resolution (e.g., 384x384), and concatenating them horizontally into a single wide image (e.g., 384x1152). This wide image is processed by a standard Vision Transformer (ViT) backbone [60]. Using 32x32 patches results in a large number of initial tokens ( $N=432$  for a 384x1152 input). These tokens, augmented with standard position embeddings.

The primary challenge is the computational cost associated with processing this large initial token count ( $N = 432$ ) through self-attention layers, which have ( $O(N^2)$ ) complexity. This significantly hinders practical deployment. To address this efficiency bottleneck, we introduce the following techniques designed to reduce the effective number of processed tokens while preserving critical biometric information: Keypoint-based Token Fusion (KP-ToFu), Keypoint Absolute Position Encoding (KP-APE), and Reasoning Tokens, detailed in the subsequent sections.

### 7.3.2 Keypoint-based Token Fusion (KP-ToFu)

Our approach to improving the efficiency of the ViT backbone hinges on reducing the number of tokens  $N$  processed in its layers. We adapt recent training-free token reduction techniques, specifically focusing on token merging, but tailor it for the biometric recognition.

**1. Standard Token Merging** Standard Token Merging employs Bipartite Soft Matching (BSM) [28] to identify the  $r$  most similar pairs of tokens ( $\text{idx}_{\text{src}}, \text{idx}_{\text{dst}}$ ) within an input sequence  $\mathbf{X} \in \mathbb{R}^{N \times C}$ . Similarity is typically based on token features from the preceding Multihead Self Attention (MSA) layer (e.g., using Key  $\mathbf{K}$  vectors averaged across heads). The  $r$  source tokens (indexed by  $\text{idx}_{\text{src}}$ ) are then merged into their corresponding destination tokens (indexed by  $\text{idx}_{\text{dst}}$ ) using Average Merging via `scatter_reduce` [121]:

$$\mathbf{X}_{\text{src\_selected}} \leftarrow \mathbf{X}[\text{idx}_{\text{src}}] \quad (7.1)$$

$$\mathbf{X}' \leftarrow \mathbf{X}.\text{scatter\_reduce}(\mathbf{X}_{\text{src\_selected}}, \text{idx}_{\text{dst}}, \text{mode}=\text{'mean'}) \quad (7.2)$$

After merging, the  $r$  source tokens are removed, yielding  $N - r$  tokens. However, this standard approach is unaware of token semantics and can inadvertently merge tokens representing anatomical keypoints, degrading biometric performance.

**2. Keypoint Identification and Index Partitioning** For robust biometric recognition, preserving keypoint information is crucial. We first identify the indices of tokens corresponding to  $K$  anatomical keypoints (detected via [34]), denoted as the set  $I_{\text{kp}} \subset \{1, \dots, N\}$ . The remaining token indices form the non-keypoint set  $I_{\text{nkp}} = \{1, \dots, N\} \setminus I_{\text{kp}}$ . This distinction is fundamental to our modified fusion strategy.

**3. KP-ToFu: Keypoint-Preserving Merging** Our Keypoint-based Token Fusion (KP-ToFu) method modifies the BSM matching process to explicitly protect keypoint tokens. We achieve this by carefully defining the source and destination pools for BSM:

- The non-keypoint indices  $I_{\text{nkp}}$  is partitioned into two subsets,  $I_{\text{nkp}}^{(1)}$  and  $I_{\text{nkp}}^{(2)}$ , based on alternating index.
- The **Source (SRC)** pool for BSM is restricted to tokens indexed by  $I_{\text{nkp}}^{(1)}$ .
- The **Destination (DST)** pool for BSM includes tokens indexed by the other non-keypoint partition *plus* all keypoint tokens, i.e.,  $I_{\text{nkp}}^{(2)} \cup I_{\text{kp}}$ .

Then we find the  $r$  most similar pairs  $(\text{idx}_{\text{src}}, \text{idx}_{\text{dst}})$  between the SRC and DST pools, using the same similarity metric (averaged Key vectors). Crucially, this construction guarantees that the source indices  $\text{idx}_{\text{src}}$  are always a subset of  $I_{\text{nkp}}^{(1)}$ , ensuring no keypoint token is ever selected for removal ( $\forall i \in \text{idx}_{\text{src}}, i \notin I_{\text{nkp}}$ ).

The merging (Eq. 7.2) and subsequent removal of the  $r$  source tokens proceed as in standard merging, producing a sequence of  $N - r$  tokens. KP-ToFu thus efficiently reduces token count while guaranteeing the preservation of all  $K$  keypoint tokens, maintaining the structural fidelity essential for accurate biometric recognition.

### 7.3.3 Keypoint Absolute Position Encoding (KP-APE)

Token fusion via KP-ToFu (Sec 7.3.2) disrupts the token grid, making standard positional encoding methods like KP-RPE [125] impractical due to the excessive overhead of merging relative positional biases at each layer. To efficiently provide spatial awareness in the reduced token set, we introduce Keypoint Absolute Position Encoding (KP-APE).

KP-APE leverages the importance of anatomical landmarks in biometrics. We define a set of learnable absolute position embeddings,  $\mathbf{P}_{kp} \in \mathbb{R}^{K \times C}$ , one  $\mathbf{p}_k \in \mathbb{R}^C$  for each of the  $K$  keypoint types. At each layer  $l$ , KP-APE updates every token  $\mathbf{x}_i^{(l)}$  in the current sequence  $\mathbf{X}^{(l)}$  by adding a distance-weighted sum of the keypoint embeddings. Using learnable non-negative decay parameters  $\lambda_k$ , the update is:

$$\mathbf{x}_i'^{(l)} = \mathbf{x}_i^{(l)} + \sum_{k=1}^K e^{-\lambda_k d_{i,k}} \mathbf{p}_k \quad (7.3)$$

Here,  $\mathbf{x}_i'^{(l)}$  denotes the updated token feature vector, and  $d_{i,k}$  is the distance between one token and a keypoint token.

This distance  $d_{i,k}$  is determined by tracking token locations. We initialize  $(x, y)$  coordinates for each token and update them at each layer  $l$  in parallel with KP-ToFu. As tokens merge (using indices  $\text{idx}_{\text{src}}, \text{idx}_{\text{dst}}$ ), their coordinates are also merged. In other words, the average of coordinate represents the location of the new token.  $d_{i,k}$  is then the Euclidean distance between the resulting tracked coordinate of token  $i$  and the coordinate of keypoint  $k$ .

This approach efficiently encodes spatial information. The benefit is that keypoint tokens receive a strong signal from their corresponding embedding ( $d_{i,k} \approx 0 \implies e^{-\lambda_k d_{i,k}} \approx 1$ , assuming keypoints don't merge into non-keypoints). Furthermore, all tokens gain awareness of their position relative to keypoints via distance-modulated contributions based on their dynamically updated effective locations. It adapts seamlessly to the fused token sequence at each layer using tracked coordinates and fixed embeddings. KP-APE thus maintains crucial spatial reasoning capabilities focused on keypoints within an efficient, fusion-compatible framework.

### 7.3.4 Reasoning Tokens

To enhance pose/shape invariance and compensate for increased information density after token fusion (Sec 7.3.2), we introduce Reasoning Tokens (RTs). These are learnable, randomly initialized tokens, not tied to specific spatial image locations, functioning as adaptable computational resources within the network.

RTs are progressively added based on a schedule specifying  $r_l \geq 0$  new tokens for each transformer block  $l$ . Let  $\mathbf{X}^{(l)} \in \mathbb{R}^{N^{(l)} \times C}$  be the image tokens entering block  $l$ , and  $\mathbf{R}^{(l)} \in \mathbb{R}^{M^{(l)} \times C}$  be the RTs propagated from the previous block ( $M^{(1)} = 0$ ). We initialize  $r_l$  new RTs,  $\mathbf{R}_{\text{new}}^{(l)} \in \mathbb{R}^{r_l \times C}$ .

The full input sequence to block  $l$ 's self-attention is the concatenation:

$$\mathbf{Z}^{(l)} = \text{Concat}(\mathbf{X}^{(l)}, \mathbf{R}^{(l)}, \mathbf{R}_{\text{new}}^{(l)}) \in \mathbb{R}^{(N^{(l)} + M^{(l)} + r_l) \times C} \quad (7.4)$$

All tokens in  $\mathbf{Z}^{(l)}$  interact. The output tokens corresponding to  $\mathbf{R}^{(l)}$  and  $\mathbf{R}_{\text{new}}^{(l)}$  form the propagated set  $\mathbf{R}^{(l+1)}$  for the next block. These RTs provide additional capacity for the model to synthesize features, integrate information across image tokens, and potentially distill more abstract, invariant semantic information crucial for robust identity recognition, mitigating potential representation loss from token fusion.

Method	LFW [100]	CPLFW [296]	CFPFP [202]	CALFW [297]	FLOPs (G)
SapiensID [125]	99.82	94.85	98.74	95.78	31.77
Proposed Work	99.82	<b>94.92</b>	<b>98.80</b>	<b>95.88</b>	<b>20.87</b>

Table 7.1 Face Recognition Performance Comparison. Accuracy (%) is reported. FLOPs (G) are measured for a single forward pass with the standard input (384x1152).

## 7.4 Experiments

**Implementation Details** We evaluate our proposed efficient unified recognition backbone against the SapiensID [126] baseline. Both models use a ViT-Base architecture and are trained on the WebBody4M dataset using the AdaFace loss [122], following established training practices. The input image size is  $384 \times 384$  (with padding), and 3 ROIs (whole image, upper torso, head) are extracted, initially leading to  $12 \times 12 \times 3 = 432$  patches for the ViT backbone.

**Face Recognition Performance** We evaluate performance on standard aligned face recognition benchmarks. Table 7.1 compares our Proposed Work against the SapiensID baseline. The results show that our method achieves highly comparable accuracy across all datasets (LFW, CPLFW, CFPFP, CALFW). This demonstrates that the proposed efficiency enhancements, including token fusion and the KP-APE positional encoding, successfully preserve the fine-grained details necessary for face identification.

**Body Recognition Performance** We further evaluate on body recognition benchmarks, including long-term ReID (LTCC, PRCC, focusing on clothing changes) and short-term ReID (Market1501). Table 7.2 shows the top-1 accuracy comparison. Our Proposed Work demonstrates strong performance, achieving higher accuracy on PRCC long-term ReID compared to the baseline. While slightly lower on LTCC and Market1501, the results remain highly competitive, showcasing the effectiveness of the proposed architecture in handling body recognition tasks under challenging conditions even after significant token reduction. The performance trade-offs might reflect the different balance struck between spatial detail and

Method	Long-Term ReID		Short-Term ReID
	LTCC [212]	PRCC [268]	Market1501 [295]
SapiensID [125]	<b>42.60</b>	66.69	<b>90.53</b>
Proposed Work	29.59	<b>69.94</b>	87.98

Table 7.2 Body Recognition Performance Comparison. Top-1 Accuracy (%) is reported.

Long-term datasets use clothing-change protocols.

semantic abstraction due to token fusion and the KP-APE vs KP-RPE encoding strategies.

**Efficiency Analysis** A primary contribution of our work is the enhancement of computational efficiency. As indicated in Table 7.1, our Proposed Work reduces the computational cost from 31.77 GFLOPs (SapiensID baseline) to 20.87 GFLOPs. This constitutes a significant reduction of approximately 34.3%, offering a theoretical speedup of about 1.52x. This efficiency gain is primarily achieved through Keypoint-based Token Fusion (KP-ToFu), which substantially decreases the number of tokens processed by the computationally intensive self-attention layers, while the addition of Reasoning Tokens incurs minimal overhead. This makes our proposed backbone much more suitable for deployment in resource-constrained environments or real-time applications.

**Visualization of Token Retention** Figure 7.2 visualizes the effectiveness of KP-ToFu in preserving keypoint-related tokens during the fusion process. For two example inputs—one seated and one standing—the visualizations compare the standard ToFu (top row) and our proposed KP-ToFu (bottom row) at deeper layers (e.g., depth 23). Each red dot indicates a remaining token after fusion. The facial and body keypoints are marked on top of the input images. We observe that KP-ToFu explicitly retains tokens near critical landmarks (e.g., facial features, joints), whereas standard ToFu can aggressively merge away fine-grained regions. This highlights how KP-ToFu maintains semantic structure crucial for accurate biometric recognition while achieving substantial token reduction.

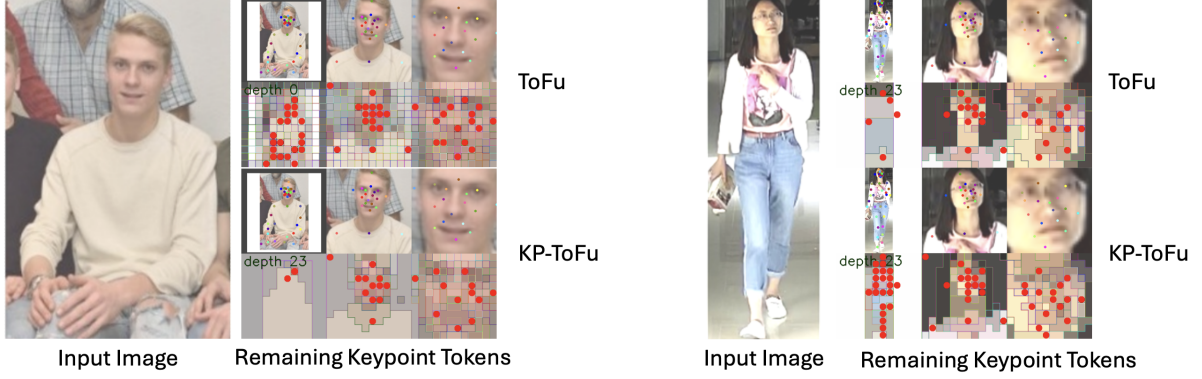


Figure 7.2 Visualization of token retention with ToFu vs. KP-ToFu. Each red dot represents a retained token at a deeper transformer layer. KP-ToFu better preserves keypoint-related tokens (e.g., facial landmarks, joints), improving structural integrity for recognition tasks.

## 7.5 Conclusion

In this work, we present a new backbone for unified human recognition that addresses a critical challenge in modern biometric systems: how to maintain high recognition accuracy across diverse human appearances while significantly reducing computational cost. We introduce three key innovations—Keypoint-based Token Fusion (KP-ToFu), Keypoint Absolute Position Encoding (KP-APE), and Reasoning Tokens—that together enable effective token reduction in Vision Transformers without compromising the fine-grained spatial information vital for face and body recognition.

Through extensive experiments, we demonstrate that our proposed method achieves recognition performance on par with or exceeding the state-of-the-art across multiple benchmarks, while reducing FLOPs by over 34%. Our approach maintains strong discriminative power for both aligned face recognition and unconstrained person re-identification, showcasing its robustness across pose, scale, and appearance variations. Furthermore, visualizations confirm that KP-ToFu preserves semantic structure by protecting keypoint-relevant tokens, a critical feature that standard token fusion methods lack. Our efficient and unified backbone paves the way for scalable, real-time biometric systems deployable on edge devices, enabling practical applications in security, forensics, and personalized user interfaces.

## CHAPTER 8

### DISCUSSION AND CONCLUSION

#### 8.1 Historical Context and Research Trajectory

As shown in Fig 8.1, face recognition has undergone several paradigm shifts over the past decades. Early systems relied on hand-crafted features such as Eigenfaces [233] and Local Binary Patterns (LBP) [7], later evolving into more structured feature engineering approaches like SIFT [165] and Haar features [237]. The major turning point arrived with the advent of deep learning around 2014. Pioneering models like DeepFace [225] and FaceNet [200] introduced end-to-end learning pipelines that significantly outperformed traditional techniques. This deep learning wave was quickly followed by a flurry of innovations in loss functions [55, 102, 122, 154, 239, 240], architecture design [86], and large-scale dataset development [300], each contributing to improved robustness and scalability.

This dissertation contributes to this ongoing evolution by addressing key limitations of modern face recognition. It comprises five core works that span the three foundational pillars of progress in this domain: loss functions, dataset design, and architectural innovations. These include:

- **Loss function innovation:** *AdaFace*, which adapts margin constraints based on image quality for robust training.
- **Architectural design:** *CAFace* and *KPRPE*, which address challenges in large-scale video recognition and pose misalignment, respectively.
- **Dataset and generative modeling:** *DCFace*, a dual-conditioned diffusion framework for generating high-quality synthetic training identities.
- **Multi-modal Biometrics** *SapiensID*: A large-scale multimodal model developed to support a new biometric paradigm that unifies face and body recognition.

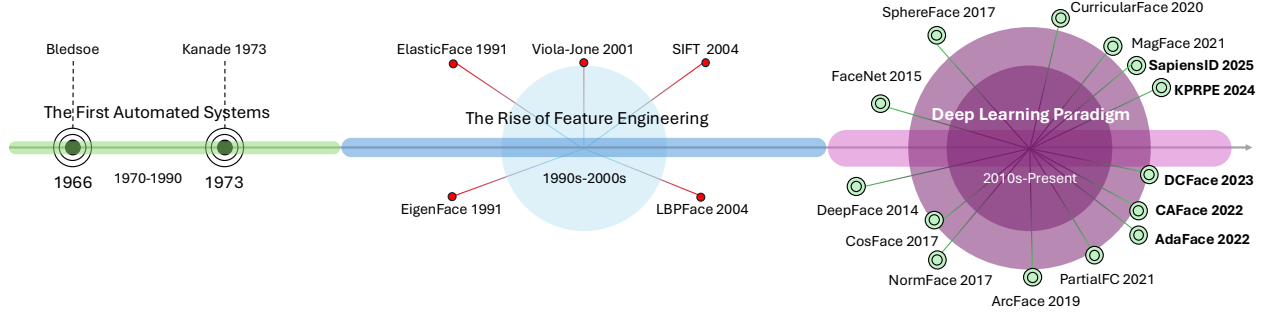


Figure 8.1 Timeline of face recognition evolution, tracing the transition from hand-crafted features to deep learning-based approaches. The contribution of this thesis is in bolded text.

## 8.2 Limitations and Open Challenges

### 8.2.1 Precision of Facial Features

As face recognition systems are increasingly deployed in unconstrained, real-world environments, the precision of facial features becomes a critical factor. In particular, systems must handle low-quality imagery, varied poses, occlusions, and other challenging conditions. Under such circumstances, the quality and reliability of the underlying keypoint detections can substantially influence recognition performance. Architectures like KP-RPE [125] aim to incorporate structural priors by explicitly modeling spatial relationships between facial landmarks. However, such approaches inherently rely on the success of the keypoint detector itself [54, 165]. When landmarks are mislocalized due to motion blur, extreme pose, or occlusion, the downstream recognition model can suffer from incorrect relational encodings.

This points to a broader challenge: the current decoupling between keypoint detection and identity recognition creates a potential vulnerability. For future systems, it may be beneficial to explore joint optimization frameworks where the keypoint estimation is co-trained or tightly coupled with the recognition objective, improving resilience in low-quality scenarios. Moreover, advancements in keypoint detection, especially under degraded conditions such as surveillance video—will be essential for enabling robust structural encodings in the wild. As applications move beyond controlled datasets, these front-end challenges are likely to become bottlenecks, underscoring the need for more integrated, end-to-end solutions.

### 8.2.2 Identical Twins and Similarity Challenges

Despite significant advances, deep face recognition systems still struggle with edge cases such as identical twins. Even under high-quality imaging conditions, the facial similarities between twins can be so high that image-based models fail to reliably distinguish them [130]. Consumer devices have demonstrated this limitation. Basic facial recognition methods, which rely on matching key facial features, can sometimes be fooled by twins. To remedy this, advanced systems use depth sensors to reduce the likelihood of errors.

This challenge highlights a fundamental limitation of appearance-based recognition: when inter-subject variance is extremely low, visual information alone may be insufficient. Recent work has extended this insight to the broader problem of lookalike disambiguation, where non-twin individuals exhibit extremely similar facial appearances. Swearingen and Ross [223] proposed a reranking strategy that augments traditional face matchers with a dedicated disambiguator, specifically tuned to distinguish between such lookalikes. Their method improved closed-set identification accuracy on the challenging TinyFace dataset, suggesting that hybrid architectures may offer a viable path forward in these hard scenarios.

To further improve robustness, future systems will likely need to adopt multi-modal inputs, integrating cues such as depth or voice. These additional modalities can provide independent, discriminative signals that are more effective when appearance alone fails. As face recognition moves into more diverse and security-critical applications, handling these hard cases will be essential for practical reliability.

### 8.2.3 Interpretability and Trustworthiness

Modern recognition systems function as black boxes, often outputting similarity scores without rationale. Despite efforts in visualizing attention or embedding distances, interpretability remains minimal. In high-stakes scenarios such as border control or forensic analysis, models must provide calibrated confidence, clear failure modes, and possibly human-interpretable explanations.

### 8.2.4 Identity Capacity of Generative Models

An emerging question in synthetic dataset design is not just whether generated faces look realistic, but how many truly distinct and usable identities a generative model can produce. This is fundamentally a question of *identity capacity*: given a fixed number of real training images, how many well-separated subjects can a model generate?

DCFace [124], trained on 52k real face images, generates 20k new synthetic identities. In contrast, Vec2Face [255], trained on a much larger dataset (360k images), achieves up to 200k well-separated identities. This scaling behavior demonstrates that generative identity capacity is closely related to the diversity and richness of the real training data.

Recent work by Boddeti et al. [26] offers a principled statistical framework for estimating the upper bound of this capacity, framing it as a hyperspherical packing problem in the feature space of a face recognition model. They define capacity as the maximum number of identities that can be placed in this space without exceeding a predefined similarity threshold (related to a false acceptance rate). Their empirical estimates show that StyleGAN3 have a practical upper bound—approximately 1.43 million identities at a 0.1% FAR, which decreases sharply with stricter thresholds. For class-conditional models like DCFace, the capacity was significantly lower, due to its greater intra-class variation.

These results underscore an important insight: while generative models can amplify identity diversity, their capacity is not unlimited. The sampling distribution remains bounded by the identity entropy encoded during training. Thus, future research can aim to formalize these constraints, explore the theoretical upper bounds of novel identity generation, and propose methods for synthetic identities to be meaningfully distinct and diverse.

### 8.2.5 Recognition at Scale: The Challenge of Large Galleries

Beyond academic benchmarks, real-world biometric systems often operate at an entirely different scale. For example, India’s national identification system, Aadhaar, maintains biometric records—including face, fingerprint, and iris data—for over 1.4 billion individuals [180]. In such large-scale deployments, the gallery size is not in the thousands but in the hundreds

Gallery Setting	Gallery Size	Rank-1 Acc.	Rank-5 Acc.	TPIR @ FPIR=0.01
Baseline Gallery	202	62.0%	68.2%	46.1%
+1K External Imposters	1,202	56.1%	61.6%	43.7%
+5K External Imposters	5,202	51.1%	57.3%	40.8%
+10K External Imposters	10,202	48.4%	55.1%	38.0%

Table 8.1 Performance degradation on IJB-S (small/surv2single protocol) as gallery size increases with imposters sampled from an external dataset. The addition of external distractors reveals challenges not captured in standard closed-set benchmarks.

of millions or more. At this scale, even a small drop in recognition accuracy can lead to a significant number of false matches or missed identifications, potentially affecting millions of people.

While academic benchmarks like IJB-S offer a valuable setting to evaluate face recognition systems, they often fall short in simulating the true scale and complexity of operational deployments. In real-world applications, systems must search against vast galleries filled with distractors, occlusions, and varying quality levels.

To better approximate such conditions, we conducted an experiment where the baseline IJB-S gallery (containing approximately 202 identities in this setup) was augmented by sampling additional imposter identities from an external dataset. We added between 1,000 and 10,000 such imposters and measured the impact on recognition performance using the ‘surv2single-small’ protocol. As detailed in Table 8.1, the results show a marked deterioration in accuracy as the number of external distractors increases. This decline highlights a fundamental truth: face recognition, especially when dealing with large galleries containing unknown imposters, remains a significant challenge.

This experiment serves as a reminder that deploying face recognition systems at scale introduces complexities not yet fully captured in many academic settings. As we move forward, bridging the gap between benchmark success and real-world reliability, especially under large-scale, open-set conditions, remains a central challenge for the field.

## 8.3 Looking Ahead: Potential Future Directions

### 8.3.1 From Recognition Towards Reasoning

The next phase in biometrics appears poised to move beyond simple matching. Future recognition systems could extend beyond identification to also incorporate reasoning, explanation, and interaction capabilities. This emerging paradigm might involve multimodal inputs, contextual memory, and probabilistic inference—potentially leading to agents that could *ask follow-up questions*, *simulate identities*, or *defer to humans* in ambiguous situations.

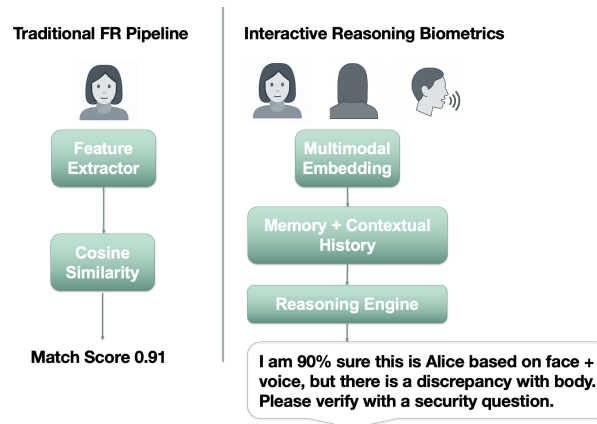


Figure 8.2 Future biometric agents may need to go beyond raw accuracy, providing calibrated confidence, interpretability, and the ability to call for intervention.

### 8.3.2 Evaluation Beyond Accuracy

Traditional metrics such as TAR or rank-1 may prove insufficient for evaluating future biometric deployments. There appears to be a growing need for new benchmarks that quantify aspects like trust, explanation clarity, and user experience. These could include metrics for: 1) Intervention accuracy (when the model seeks assistance) 2) Hypothesis quality in low-confidence settings 3) Interpretability and decision traceability.

### 8.3.3 Multimodal and Personalized Recognition

Multimodal fusion—combining face, body, gait, or even voice or language—continues to hold promise for boosting robustness. Similarly, personalized models that adapt to specific users or deployment contexts may improve usability. Pursuing these directions will

likely involve considerations around *continual learning*, *fusion architectures*, and *cross-modal representation learning*.

## 8.4 Closing Remarks

The journey of face recognition, significantly accelerated by deep learning, has reached impressive milestones, yet it is crucial to acknowledge that the core challenge of reliable biometric identification in diverse, real-world conditions is far from solved. This dissertation, therefore, concludes not at an end-point, but at what appears to be an inflection point for the community. Also the focus may increasingly pivot from pure accuracy maximization towards the creation of systems that embody trustworthiness, interpretability, and effective human interaction. Looking ahead, the goal seems less about marginal gains on leaderboards and more about engineering resilient systems designed to coexist meaningfully with humans, adapt appropriately to context, and navigate ambiguity with grace.

## BIBLIOGRAPHY

- [1] InsightFace. <https://github.com/deepinsight/insightface.git>. Accessed: 2021-09-01.
- [2] InsightFacePytorch. <https://github.com/TreB1eN/InsightFacePytorch.git>. Accessed: 2021-09-01.
- [3] TFace. <https://github.com/Tencent/TFace.git>. Accessed: 2021-10-03.
- [4] Face detection vs facial recognition – what’s the difference?, 06 2022. Accessed: 2025-03-21.
- [5] Pros and cons of facial recognition, 08 2023. Accessed: 2025-03-21.
- [6] Why facial recognition is the best biometric, 06 2023. Accessed: 2025-03-21.
- [7] Timo Ahonen, Abdenour Hadid, and Matti Pietikäinen. Face recognition with local binary patterns. In *ECCV, Prague, Czech Republic, May 11-14, 2004. Proceedings, Part I 8*, pages 469–481. Springer, 2004.
- [8] Joshua Ainslie, James Lee-Thorp, Michiel de Jong, Yury Zemlyanskiy, Federico Lebrón, and Sumit Sanghai. GQA: Training generalized Multi-Query transformer models from Multi-Head checkpoints. In *arXiv*, May 2023.
- [9] Vítor Albiero. Face analysis pytorch. <https://github.com/vitoralbiero/faceanalysis>, 2022.
- [10] Xiang An, Jiankang Deng, Jia Guo, Ziyong Feng, XuHan Zhu, Jing Yang, and Tongliang Liu. Killing two birds with one stone: Efficient and robust training of face recognition cnns by partial fc. In *CVPR*, pages 4042–4051, 2022.
- [11] Xiang An, Xuhan Zhu, Yuan Gao, Yang Xiao, Yongle Zhao, Ziyong Feng, Lan Wu, Bin Qin, Ming Zhang, Debing Zhang, et al. Partial fc: Training 10 million identities on a single machine. In *ICCV*, pages 1445–1449, 2021.
- [12] Ognjen Arandjelovic, Gregory Shakhnarovich, John Fisher, Roberto Cipolla, and Trevor Darrell. Face recognition with image sets using manifold density divergence. In *CVPR*, 2005.
- [13] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. Vivit: A video vision transformer. In *ICCV*, 2021.
- [14] Vishal Asnani, Xi Yin, Tal Hassner, Sijia Liu, and Xiaoming Liu. Proactive image manipulation detection. In *CVPR*, 2022.
- [15] Shekoofeh Azizi, Simon Kornblith, Chitwan Saharia, Mohammad Norouzi, and David J Fleet. Synthetic data from diffusion models improves imagenet classification. *arXiv preprint*, 2023.
- [16] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint*, 2016.

- [17] Gwangbin Bae, Martin de La Gorce, Tadas Baltrusaitis, Charlie Hewitt, Dong Chen, Julien Valentin, Roberto Cipolla, and Jingjing Shen. Digiface-1m: 1 million digital face images for face recognition. In *WACV*, 2023.
- [18] Romain Beaumont. img2dataset: Easily turn large sets of image urls to an image dataset. <https://github.com/rom1504/img2dataset>, 2021.
- [19] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *ICML*, pages 41–48, 2009.
- [20] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding. In *ICML*, 2021.
- [21] Lacey Best-Rowden, Hu Han, Charles Otto, Brendan F Klare, and Anil K Jain. Unconstrained face recognition: Identifying a person of interest from a media collection. *TIFS*, 2014.
- [22] Lucas Beyer, Pavel Izmailov, Alexander Kolesnikov, Mathilde Caron, Simon Kornblith, Xiaohua Zhai, Matthias Minderer, Michael Tschannen, Ibrahim Alabdulmohsin, and Filip Pavetic. Flexivit: One model for all patch sizes. In *CVPR*, 2023.
- [23] Lucas Beyer, Xiaohua Zhai, and Alexander Kolesnikov. Better plain vit baselines for imagenet-1k. *arXiv preprint*, 2022.
- [24] Volker Blanz and Thomas Vetter. A morphable model for the synthesis of 3D faces. In *SIGGRAPH*, 1999.
- [25] Andreas Blattmann, Robin Rombach, Kaan Oktay, and Björn Ommer. Retrieval-augmented diffusion models. *arXiv preprint*, 2022.
- [26] Vishnu Naresh Boddeti, Gautam Sreeksumar, and Arun Ross. On the biometric capacity of generative face models. In *2023 IEEE International Joint Conference on Biometrics (IJCB)*, pages 1–10. IEEE, 2023.
- [27] Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. Token merging: Your vit but faster. *arXiv preprint*, 2022.
- [28] Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. Token merging: Your ViT but faster. In *ICLR*, 2023.
- [29] Daniel Bolya and Judy Hoffman. Token merging for fast stable diffusion. In *CVPR Workshop*, pages 4598–4602, Mar. 2023.
- [30] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint*, 2018.
- [31] Adrian Bulat and Georgios Tzimiropoulos. How far are we from solving the 2D & 3D face alignment problem? (and a dataset of 230,000 3d facial landmarks). In *ICCV*, 2017.

- [32] Mikhail S Burtsev, Yuri Kuratov, Anton Peganov, and Grigory V Sapunov. Memory transformer. *arXiv preprint*, 2020.
- [33] Qiong Cao, Li Shen, Weidi Xie, Omkar M Parkhi, and Andrew Zisserman. VGGFace2: A dataset for recognising faces across pose and age. In *FG*, 2018.
- [34] Z. Cao, G. Hidalgo Martinez, T. Simon, S. Wei, and Y. A. Sheikh. Openpose: Realtime multi-person 2d pose estimation using part affinity fields. *PAMI*, 2019.
- [35] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *CVPR*, pages 7291–7299, 2017.
- [36] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, 2020.
- [37] Hakan Cevikalp and Bill Triggs. Face recognition based on image sets. In *CVPR*, 2010.
- [38] Jie Chang, Zhonghao Lan, Changmao Cheng, and Yichen Wei. Data uncertainty learning in face recognition. In *CVPR*, pages 5710–5719, 2020.
- [39] Jie Chang, Zhonghao Lan, Changmao Cheng, and Yichen Wei. Data uncertainty learning in face recognition. In *CVPR*, 2020.
- [40] Hanqing Chao, Yiwei He, Junping Zhang, and Jianfeng Feng. Gaitset: Regarding gait as a set for cross-view gait recognition. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 8126–8133, 2019.
- [41] Jiaying Chen, Xinyang Jiang, Fudong Wang, Jun Zhang, Feng Zheng, Xing Sun, and Wei-Shi Zheng. Learning 3d shape feature for texture-insensitive person re-identification. In *CVPR*, 2021.
- [42] Jun-Cheng Chen, Rajeev Ranjan, Amit Kumar, Ching-Hui Chen, Vishal M Patel, and Rama Chellappa. An end-to-end system for unconstrained face verification with deep convolutional neural networks. In *ICCVW*, 2015.
- [43] Minshuo Chen, Song Mei, Jianqing Fan, and Mengdi Wang. An overview of diffusion models: Applications, guided generation, statistical rates and optimization. *arXiv preprint*, 2024.
- [44] Weihua Chen, Xianzhe Xu, Jian Jia, Hao Luo, Yaohua Wang, Fan Wang, Rong Jin, and Xiuyu Sun. Beyond appearance: a semantic controllable self-supervised learning framework for human-centric visual tasks. In *CVPR*, 2023.
- [45] Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised vision transformers. In *ICCV*, 2021.
- [46] Zhiyi Cheng, Xiatian Zhu, and Shaogang Gong. Low-resolution face recognition. In *Asian Conference on Computer Vision*, pages 605–621, 2018.

- [47] Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. StarGAN: Unified generative adversarial networks for multi-domain image-to-image translation. In *CVPR*, 2018.
- [48] Xiangxiang Chu, Zhi Tian, Bo Zhang, Xinlong Wang, Xiaolin Wei, Huaxia Xia, and Chunhua Shen. Conditional positional encodings for vision transformers. *arXiv preprint*, 2021.
- [49] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *CVPR workshops*, pages 702–703, 2020.
- [50] Zihang Dai, Zhilin Yang, Yiming Yang, Jaime G Carbonell, Quoc Le, and Ruslan Salakhutdinov. Transformer-xl: Attentive language models beyond a fixed-length context. In *ACL*, pages 2978–2988, 2019.
- [51] Mostafa Dehghani, Basil Mustafa, Josip Djolonga, Jonathan Heek, Matthias Minderer, Mathilde Caron, Andreas Steiner, Joan Puigcerver, Robert Geirhos, Ibrahim M Alabdulmohsin, et al. Patch n’pack: Navit, a vision transformer for any aspect ratio and resolution. In *NeurIPS*, 2024.
- [52] Jiankang Deng, Shiyang Cheng, Niannan Xue, Yuxiang Zhou, and Stefanos Zafeiriou. UV-GAN: Adversarial facial uv map completion for pose-invariant face recognition. In *CVPR*, 2018.
- [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. Ieee*, 2009.
- [54] Jiankang Deng, Jia Guo, Evangelos Ververas, Irene Kotsia, and Stefanos Zafeiriou. Retinaface: Single-shot multi-level face localisation in the wild. In *CVPR*, pages 5203–5212, 2020.
- [55] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. ArcFace: Additive angular margin loss for deep face recognition. In *CVPR*, pages 4690–4699, 2019.
- [56] Jiankang Deng, Jia Guo, Jing Yang, Alexandros Lattas, and Stefanos Zafeiriou. Variational prototype learning for deep face recognition. In *CVPR*, pages 11906–11915, 2021.
- [57] Jiankang Deng, Jia Guo, Debing Zhang, Yafeng Deng, Xiangju Lu, and Song Shi. Lightweight face recognition challenge. In *ICCV Workshops*, pages 0–0, 2019.
- [58] Li Deng. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012.
- [59] Yu Deng, Jiaolong Yang, Dong Chen, Fang Wen, and Xin Tong. Disentangled and controllable face image generation via 3D imitative-contrastive learning. In *CVPR*, 2020.

- [60] 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. *arXiv preprint*, 2020.
- [61] 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.
- [62] Huanzhang Dou, Pengyi Zhang, Wei Su, Yunlong Yu, Yining Lin, and Xi Li. Gaitgci: Generative counterfactual intervention for gait recognition. In *CVPR*, 2023.
- [63] Stefan Elfving, Eiji Uchibe, and Kenji Doya. Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. *Neural Networks*, 107, 2018.
- [64] Joshua J Engelsma, Steven A Grosz, and Anil K Jain. Printsgan: synthetic fingerprint generator. *TPAMI*, 2022.
- [65] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *kdd*, 1996.
- [66] Chao Fan, Saihui Hou, Yongzhen Huang, and Shiqi Yu. Exploring deep models for practical gait recognition. *arXiv preprint*, 2023.
- [67] Chao Fan, Junhao Liang, Chuanfu Shen, Saihui Hou, Yongzhen Huang, and Shiqi Yu. Opengait: Revisiting gait recognition toward better practicality. *arXiv preprint*, 2022.
- [68] Mohsen Fayyaz, Soroush Abbasi Koohpayegani, Farnoush Rezaei Jafari, Sunando Sengupta, Hamid Reza Vaezi Joze, Eric Sommerlade, Hamed Pirsiavash, and Jürgen Gall. Adaptive token sampling for efficient vision transformers. In *ECCV*, pages 396–414, 2022.
- [69] Yao Feng, Fan Wu, Xiaohu Shao, Yanfeng Wang, and Xi Zhou. Joint 3d face reconstruction and dense alignment with position map regression network. In *ECCV*, pages 534–551, 2018.
- [70] Dengpan Fu, Dongdong Chen, Hao Yang, Jianmin Bao, Lu Yuan, Lei Zhang, Houqiang Li, Fang Wen, and Dong Chen. Large-scale pre-training for person re-identification with noisy labels. *arXiv preprint*, 2022.
- [71] Yixiao Ge, Dapeng Chen, and Hongsheng Li. Mutual mean-teaching: Pseudo label refinery for unsupervised domain adaptation on person re-identification. In *ICLR*, 2020.
- [72] Yixiao Ge, Feng Zhu, Dapeng Chen, Rui Zhao, et al. Self-paced contrastive learning with hybrid memory for domain adaptive object re-id. In *NeurIPS*, 2020.
- [73] Baris Gecer, Binod Bhattarai, Josef Kittler, and Tae-Kyun Kim. Semi-supervised adversarial learning to generate photorealistic face images of new identities from 3D morphable model. In *ECCV*, 2018.

- [74] Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. Convolutional sequence to sequence learning. In *ICML*, pages 1243–1252. PMLR, 2017.
- [75] Zhenglin Geng, Chen Cao, and Sergey Tulyakov. 3D guided fine-grained face manipulation. In *CVPR*, 2019.
- [76] Sharath Girish, Saksham Suri, Sai Saketh Rambhatla, and Abhinav Shrivastava. Towards discovery and attribution of open-world gan generated images. In *ICCV*, 2021.
- [77] Sixue Gong, Yichun Shi, and Anil Jain. Low quality video face recognition: Multi-mode aggregation recurrent network (MARN). In *ICCVW*, 2019.
- [78] Sixue Gong, Yichu Shi, Nathan D Kalka, and Anil K Jain. Video face recognition: Component-wise feature aggregation network (C-Fan). In *ICB*, 2019.
- [79] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11), 2020.
- [80] Xinqian Gu, Hong Chang, Bingpeng Ma, Shutao Bai, Shiguang Shan, and Xilin Chen. Clothes-changing person re-identification with RGB modality only. In *CVPR*, 2022.
- [81] J Gui, Z Sun, Y Wen, D Tao, and J Ye. A review on generative adversarial networks: Algorithms, theory, and applications. arxiv preprint arxiv: 200106937. 2020.
- [82] Yandong Guo, Lei Zhang, Yuxiao Hu, Xiaodong He, and Jianfeng Gao. MS-Celeb-1M: A dataset and benchmark for large-scale face recognition. In *ECCV*, pages 87–102, 2016.
- [83] Ryo Hachiuma, Fumiaki Sato, and Taiki Sekii. Unified keypoint-based action recognition framework via structured keypoint pooling. In *CVPR*, pages 22962–22971, 2023.
- [84] Mehrtash T Harandi, Conrad Sanderson, Sareh Shirazi, and Brian C Lovell. Graph embedding discriminant analysis on grassmannian manifolds for improved image set matching. In *CVPR*, 2011.
- [85] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *CVPR*, 2022.
- [86] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.
- [87] Mingxing He, Shi-Jinn Horng, Pingzhi Fan, Ray-Shine Run, Rong-Jian Chen, Jui-Lin Lai, Muhammad Khurram Khan, and Kevin Octavius Sentosa. Performance evaluation of score level fusion in multimodal biometric systems. *Pattern Recognition*, 43(5):1789–1800, 2010.
- [88] Tong He, Zhi Zhang, Hang Zhang, Zhongyue Zhang, Junyuan Xie, and Mu Li. Bag of tricks for image classification with convolutional neural networks. In *CVPR*, pages 558–567, 2019.

- [89] Byeongho Heo, Song Park, Dongyoon Han, and Sangdoo Yun. Rotary position embedding for vision transformer. In *ECCV (ECCV)*, 2024.
- [90] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *NeurIPS*, 30, 2017.
- [91] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *NeurIPS*, 33, 2020.
- [92] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint*, 2022.
- [93] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 1997.
- [94] Elad Hoffer, Tal Ben-Nun, Itay Hubara, Niv Giladi, Torsten Hoeffler, and Daniel Soudry. Augment your batch: Improving generalization through instance repetition. In *CVPR*, pages 8129–8138, 2020.
- [95] Peixian Hong, Tao Wu, Ancong Wu, Xintong Han, and Wei-Shi Zheng. Fine-grained shape-appearance mutual learning for cloth-changing person re-identification. In *CVPR*, 2021.
- [96] Qiyang Hu, Attila Szabó, Tiziano Portenier, Paolo Favaro, and Matthias Zwicker. Disentangling factors of variation by mixing them. In *CVPR*, 2018.
- [97] Yiqun Hu, Ajmal S Mian, and Robyn Owens. Sparse approximated nearest points for image set classification. In *CVPR*, 2011.
- [98] Yuan-Ting Hu, Jiahong Wang, Raymond A Yeh, and Alexander G Schwing. Sail-vos 3d: A synthetic dataset and baselines for object detection and 3d mesh reconstruction from video data. In *CVPR*, 2021.
- [99] Gary Huang, Marwan Mattar, Honglak Lee, and Erik Learned-Miller. Learning to align from scratch. *NeurIPS*, 25, 2012.
- [100] Gary B Huang, Marwan Mattar, Tamara Berg, and Eric Learned-Miller. Labeled Faces in the Wild: A database for studying face recognition in unconstrained environments. In *Workshop on Faces in ‘Real-Life’ Images: Detection, Alignment, and Recognition*, 2008.
- [101] Yuge Huang, Pengcheng Shen, Ying Tai, Shaoxin Li, Xiaoming Liu, Jilin Li, Feiyue Huang, and Rongrong Ji. Improving face recognition from hard samples via distribution distillation loss. In *ECCV*, pages 138–154, 2020.
- [102] Yuge Huang, Yuhan Wang, Ying Tai, Xiaoming Liu, Pengcheng Shen, Shaoxin Li, Jilin Li, and Feiyue Huang. CurricularFace: adaptive curriculum learning loss for deep face recognition. In *CVPR*, pages 5901–5910, 2020.
- [103] Yan Huang, Qiang Wu, Jingsong Xu, and Yi Zhong. Celebrities-ReID: A benchmark for clothes variation in long-term person re-identification. In *IJCNN*, 2019.

- [104] Yan Huang, Jingsong Xu, Qiang Wu, Yi Zhong, Peng Zhang, and Zhaoxiang Zhang. Beyond scalar neuron: Adopting vector-neuron capsules for long-term person re-identification. *TCSVT*, 2019.
- [105] Zhiheng Huang, Davis Liang, Peng Xu, and Bing Xiang. Improve transformer models with better relative position embeddings. In *EMNLP*, pages 3327–3335, Online, Nov. 2020.
- [106] Zhiwu Huang, Ruiping Wang, Shiguang Shan, Xianqiu Li, and Xilin Chen. Log-euclidean metric learning on symmetric positive definite manifold with application to image set classification. In *ICML*, 2015.
- [107] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, 2015.
- [108] Anil K Jain, Karthik Nandakumar, and Arun Ross. 50 years of biometric research: Accomplishments, challenges, and opportunities. *Pattern recognition letters*, 79:80–105, 2016.
- [109] Xiaoyi Jiang, Michael Binkert, Bernard Achermann, and Horst Bunke. Towards detection of glasses in facial images. *Pattern Analysis & Applications*, 3(1), 2000.
- [110] Xin Jin, Tianyu He, Kecheng Zheng, Zhiheng Yin, Xu Shen, Zhen Huang, Ruoyu Feng, Jianqiang Huang, Zhibo Chen, and Xian-Sheng Hua. Cloth-changing person re-identification from a single image with gait prediction and regularization. In *CVPR*, 2022.
- [111] Glenn Jocher, Ayush Chaurasia, and Jing Qiu. Ultralytics yolov8, 2023.
- [112] Nathan D Kalka, Brianna Maze, James A Duncan, Kevin O’Connor, Stephen Elliott, Kaleb Hebert, Julia Bryan, and Anil K Jain. IJB–S: IARPA Janus Surveillance Video Benchmark. In *2018 IEEE 9th International Conference on Biometrics Theory, Applications and Systems (BTAS)*, pages 1–9, 2018.
- [113] Takeo Kanade. Picture processing system by computer complex and recognition of human faces. 1974.
- [114] Bong-Nam Kang, Yonghyun Kim, Bongjin Jun, and Daijin Kim. Attentional feature-pair relation networks for accurate face recognition. In *ICCV*, pages 5472–5481, 2019.
- [115] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. In *ICLR*, 2018.
- [116] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *CVPR*, 2019.
- [117] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *CVPR*, 2020.

- [118] Rawal Khierodkar, Timur Bagautdinov, Julieta Martinez, Su Zhaoen, Austin James, Peter Selednik, Stuart Anderson, and Shunsuke Saito. Sapiens: Foundation for human vision models. In *ECCV*, pages 206–228. Springer, 2025.
- [119] Hyeonwoo Kim, Pablo Garrido, Ayush Tewari, Weipeng Xu, Justus Thies, Matthias Niessner, Patrick Pérez, Christian Richardt, Michael Zollhöfer, and Christian Theobalt. Deep video portraits. *TOG*, 2018.
- [120] Insoo Kim, Seungju Han, Ji-won Baek, Seong-Jin Park, Jae-Joon Han, and Jinwoo Shin. Quality-agnostic image recognition via invertible decoder. In *CVPR*, pages 12257–12266, 2021.
- [121] Minchul Kim, Shangqian Gao, Yen-Chang Hsu, Yilin Shen, and Hongxia Jin. Token fusion: Bridging the gap between token pruning and token merging. In *WACV*, pages 1383–1392, 2024.
- [122] Minchul Kim, Anil K Jain, and Xiaoming Liu. AdaFace: Quality adaptive margin for face recognition. In *CVPR*, 2022.
- [123] Minchul Kim, Feng Liu, Anil Jain, and Xiaoming Liu. Cluster and aggregate: Face recognition with large probe set. *NeurIPS*, 2022.
- [124] Minchul Kim, Feng Liu, Anil Jain, and Xiaoming Liu. DCFace: Synthetic face generation with dual condition diffusion model. 2023.
- [125] Minchul Kim, Yiyang Su, Feng Liu, Anil Jain, and Xiaoming Liu. Keypoint relative position encoding for face recognition. In *CVPR*, 2024.
- [126] Minchul Kim, Dingqiang Ye, Yiyang Su, Feng Liu, and Xiaoming Liu. Sapiensid: Foundation for human recognition. In *CVPR*, 2025.
- [127] Tae Hyun Kim, Mehdi SM Sajjadi, Michael Hirsch, and Bernhard Scholkopf. Spatio-temporal transformer network for video restoration. In *ECCV*, 2018.
- [128] Yonghyun Kim, Wonpyo Park, and Jongju Shin. BroadFace: Looking at tens of thousands of people at once for face recognition. In *ECCV*, pages 536–552, 2020.
- [129] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.
- [130] Brendan Klare, Alessandra A Paulino, and Anil K Jain. Analysis of facial features in identical twins. In *2011 International Joint Conference on Biometrics (IJCB)*, pages 1–8. IEEE, 2011.
- [131] Brendan F Klare, Ben Klein, Emma Taborsky, Austin Blanton, Jordan Cheney, Kristen Allen, Patrick Grother, Alan Mah, and Anil K Jain. Pushing the frontiers of unconstrained face detection and recognition: IARPA Janus Benchmark-A. In *CVPR*, 2015.

- [132] Abhinav Kumar, Tim K. Marks, Wenxuan Mou, Ye Wang, Michael Jones, Anoop Cherian, Toshiaki Koike-Akino, Xiaoming Liu, and Chen Feng. Luvli face alignment: Estimating landmarks' location, uncertainty, and visibility likelihood. In *CVPR*, 2020.
- [133] David Kupas and Balazs Harangi. Solving the problem of imbalanced dataset with synthetic image generation for cell classification using deep learning. In *EMBC*, 2021.
- [134] Tuomas Kynkäänniemi, Tero Karras, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Improved precision and recall metric for assessing generative models. *NeurIPS*, 32, 2019.
- [135] Kenneth Lai, Leonardo Queiroz, Vlad Shmerko, Kelly Sundberg, and Svetlana Yanushkevich. Post-pandemic follow-up audit of security checkpoints. *IEEE Access*, 11:7599–7616, 2023.
- [136] HyunJae Lee, Hyo-Eun Kim, and Hyeonseob Nam. Srm: A style-based recalibration module for convolutional neural networks. In *ICCV*, 2019.
- [137] Minxian Li, Xiatian Zhu, and Shaogang Gong. Unsupervised person re-identification by deep learning tracklet association. In *ECCV*, 2018.
- [138] Minxian Li, Xiatian Zhu, and Shaogang Gong. Unsupervised tracklet person re-identification. *PAMI*, 2019.
- [139] Shen Li, Jianqing Xu, Xiaqing Xu, Pengcheng Shen, Shaoxin Li, and Bryan Hooi. Spherical confidence learning for face recognition. In *CVPR*, pages 15629–15637, 2021.
- [140] Yu-Jhe Li, Xinshuo Weng, and Kris M Kitani. Learning shape representations for person re-identification under clothing change. In *WACV*, 2021.
- [141] Jingyun Liang, Jiezhong Cao, Guolei Sun, Kai Zhang, Luc Van Gool, and Radu Timofte. Swinir: Image restoration using swin transformer. In *ICCV*, 2021.
- [142] Youwei Liang, Chongjian Ge, Zhan Tong, Yibing Song, Jue Wang, and Pengtao Xie. Not all patches are what you need: Expediting vision transformers via token reorganizations. Feb. 2022.
- [143] Jianxin Lin, Yingce Xia, Tao Qin, Zhibo Chen, and Tie-Yan Liu. Conditional image-to-image translation. In *CVPR*, 2018.
- [144] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *CVPR*, pages 2117–2125, 2017.
- [145] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, pages 2980–2988, 2017.
- [146] Yutian Lin, Xuanyi Dong, Liang Zheng, Yan Yan, and Yi Yang. A bottom-up clustering approach to unsupervised person re-identification. In *AAAI*, 2019.

- [147] Feng Liu, Ryan Ashbaugh, Nicholas Chimitt, Najmul Hassan, Ali Hassani, Ajay Jaiswal, Minchul Kim, Zhiyuan Mao, Christopher Perry, Zhiyuan Ren, et al. Farsight: A physics-driven whole-body biometric system at large distance and altitude. In *WACV*, 2024.
- [148] Feng Liu, Ryan Ashbaugh, Nicholas Chimitt, Najmul Hassan, Ali Hassani, Ajay Jaiswal, Minchul Kim, Zhiyuan Mao, Christopher Perry, Zhiyuan Ren, Yiyang Su, Pegah Varghaei, Kai Wang, Xingguang Zhang, Stanley Chan, Arun Ross, Humphrey Shi, Zhangyang Wang, Anil Jain, and Xiaoming Liu. Farsight: A physics-driven whole-body biometric system at large distance and altitude. In *WACV*, 2024.
- [149] Feng Liu, Minchul Kim, ZiAng Gu, Anil Jain, and Xiaoming Liu. Learning clothing and pose invariant 3d shape representation for long-term person re-identification. In *ICCV*, 2023.
- [150] Feng Liu, Minchul Kim, Anil Jain, and Xiaoming Liu. Controllable and guided face synthesis for unconstrained face recognition. In *ECCV*, 2022.
- [151] Feng Liu, Minchul Kim, Zhiyuan Ren, and Xiaoming Liu. Distilling clip with dual guidance for learning discriminative human body shape representation. In *CVPR*, 2024.
- [152] Jiaheng Liu, Yudong Wu, Yichao Wu, Chuming Li, Xiaolin Hu, Ding Liang, and Mengyu Wang. DAM: Discrepancy alignment metric for face recognition. In *ICCV*, pages 3814–3823, 2021.
- [153] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *ECCV*, pages 21–37. Springer, 2016.
- [154] Weiyang Liu, Yandong Wen, Zhiding Yu, Ming Li, Bhiksha Raj, and Le Song. SphereFace: Deep hypersphere embedding for face recognition. In *CVPR*, pages 212–220, 2017.
- [155] Xiaoming Liu and Tsuhan Chen. Video-based face recognition using adaptive hidden markov models. In *CVPR*, 2003.
- [156] Xiaofeng Liu, Zhenhua Guo, Site Li, Lingsheng Kong, Ping Jia, Jane You, and BVK Kumar. Permutation-invariant feature restructuring for correlation-aware image set-based recognition. In *ICCV*, 2019.
- [157] Xiaofeng Liu, BVK Kumar, Chao Yang, Qingming Tang, and Jane You. Dependency-aware attention control for unconstrained face recognition with image sets. In *ECCV*, 2018.
- [158] Yaojie Liu and Xiaoming Liu. Spoof trace disentanglement for generic face anti-spoofing. *TPAMI*, 45(3), 2023.
- [159] Yu Liu, Junjie Yan, and Wanli Ouyang. Quality aware network for set to set recognition. In *CVPR*, 2017.

- [160] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, 2021.
- [161] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *ICCV*, pages 3730–3738, 2015.
- [162] Ze Liu, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, and Han Hu. Video swin transformer. *arXiv preprint*, 2021.
- [163] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint*, 2016.
- [164] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint*, 2017.
- [165] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60:91–110, 2004.
- [166] Jiwen Lu, Gang Wang, and Pierre Moulin. Image set classification using holistic multiple order statistics features and localized multi-kernel metric learning. In *ICCV*, 2013.
- [167] Kang Ma, Ying Fu, Dezhi Zheng, Chunshui Cao, Xuecai Hu, and Yongzhen Huang. Dynamic aggregated network for gait recognition. In *CVPR*, 2023.
- [168] Dmitrii Marin, Jen-Hao Rick Chang, Anurag Ranjan, Anish Prabhu, Mohammad Rastegari, and Oncel Tuzel. Token pooling in vision transformers. In *arxiv*, Oct. 2021.
- [169] Brianna Maze, Jocelyn Adams, James A Duncan, Nathan Kalka, Tim Miller, Charles Otto, Anil K Jain, W Tyler Niggel, Janet Anderson, Jordan Cheney, and Patrick Grother. IARPA Janus Benchmark-C: Face dataset and protocol. In *2018 International Conference on Biometrics (ICB)*, pages 158–165, 2018.
- [170] Safa C. Medin, Bernhard Egger, Anoop Cherian, Ye Wang, Joshua B. Tenenbaum, Xiaoming Liu, and Tim K. Marks. MOST-GAN: 3d morphable stylegan for disentangled face image manipulation. In *AAAI*, 2022.
- [171] Lingchen Meng, Hengduo Li, Bor-Chun Chen, Shiyi Lan, Zuxuan Wu, Yu-Gang Jiang, and Ser-Nam Lim. AdaViT: Adaptive vision transformers for efficient image recognition. In *CVPR*, pages 12309–12318, June 2022.
- [172] Qiang Meng, Shichao Zhao, Zhida Huang, and Feng Zhou. MagFace: A universal representation for face recognition and quality assessment. In *CVPR*, pages 14225–14234, 2021.
- [173] Anish Mittal, Anush Krishna Moorthy, and Alan Conrad Bovik. No-reference image quality assessment in the spatial domain. *IEEE Transactions on Image Processing*, 21(12):4695–4708, 2012.

- [174] Stylianos Moschoglou, Athanasios Papaioannou, Christos Sagonas, Jiankang Deng, Irene Kotsia, and Stefanos Zafeiriou. AGEDB: the first manually collected, in-the-wild age database. In *CVPR Workshops*, pages 51–59, 2017.
- [175] Hyeonseob Nam, HyunJae Lee, Jongchan Park, Wonjun Yoon, and Donggeun Yoo. Reducing domain gap by reducing style bias. In *CVPR*, 2021.
- [176] Daniel Neimark, Omri Bar, Maya Zohar, and Dotan Asselmann. Video transformer network. In *ICCV*, 2021.
- [177] Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In *ECCV*, pages 483–499. Springer, 2016.
- [178] Thu Nguyen-Phuoc, Chuan Li, Lucas Theis, Christian Richardt, and Yong-Liang Yang. HoloGAN: Unsupervised learning of 3d representations from natural images. In *ICCV*, 2019.
- [179] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *ICML*, pages 8162–8171. PMLR, 2021.
- [180] Unique Identification Authority of India (UIDAI). Aadhaar dashboard. <https://uidai.gov.in/aadhaardashboard/>, 2024. Accessed: 2024-04-01.
- [181] Necmiye Ozay, Yan Tong, Frederick W. Wheeler, and Xiaoming Liu. Improving face recognition with a quality-based probabilistic framework. In *Proceeding of IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 134–141, 2009.
- [182] Bowen Pan, Rameswar Panda, Yifan Jiang, Zhangyang Wang, Rogerio Feris, and Aude Oliva. IA-RED: Interpretability-Aware redundancy reduction for vision transformers. In *NeurIPS*, volume 34, pages 24898–24911, 2021.
- [183] George Papandreou, Tyler Zhu, Nori Kanazawa, Alexander Toshev, Jonathan Tompson, Chris Bregler, and Kevin Murphy. Towards accurate multi-person pose estimation in the wild. In *CVPR*, pages 4903–4911, 2017.
- [184] Omkar M Parkhi, Andrea Vedaldi, and Andrew Zisserman. Deep face recognition. In *BMVC*, 2015.
- [185] Jingtian Piao, Chen Qian, and Hongsheng Li. Semi-supervised monocular 3D face reconstruction with end-to-end shape-preserved domain transfer. In *ICCV*, 2019.
- [186] Konpat Preechakul, Nattanat Chatthee, Suttisak Wizadwongsa, and Supasorn Suwanakorn. Diffusion autoencoders: Toward a meaningful and decodable representation. In *CVPR*, 2022.
- [187] Albert Pumarola, Antonio Agudo, Aleix M Martinez, Alberto Sanfeliu, and Francesc Moreno-Noguer. Ganimation: Anatomically-aware facial animation from a single image. In *ECCV*, 2018.

- [188] Haibo Qiu, Baosheng Yu, Dihong Gong, Zhifeng Li, Wei Liu, and Dacheng Tao. SynFace: Face recognition with synthetic data. In *ICCV*, pages 10880–10890, 2021.
- [189] Prajit Ramachandran, Niki Parmar, Ashish Vaswani, Irwan Bello, Anselm Levskaya, and Jon Shlens. Stand-alone self-attention in vision models. *NeurIPS*, 32, 2019.
- [190] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint*, 2022.
- [191] Rajeev Ranjan, Carlos D Castillo, and Rama Chellappa. L2-constrained softmax loss for discriminative face verification. *arXiv preprint*, 2017.
- [192] Yongming Rao, Wenliang Zhao, Benlin Liu, Jiwen Lu, Jie Zhou, and Cho-Jui Hsieh. DynamicViT: Efficient vision transformers with dynamic token sparsification. In *NeurIPS*, Nov. 2021.
- [193] Jeff Rasley, Samyam Rajbhandari, Olatunji Ruwase, and Yuxiong He. Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 3505–3506, 2020.
- [194] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015.
- [195] Syed A Rizvi, P Jonathon Phillips, and Hyeonjoon Moon. The FERET verification testing protocol for face recognition algorithms. In *FG*, 1998.
- [196] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022.
- [197] Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. Dynamic routing between capsules. *Advances in neural information processing systems*, 30, 2017.
- [198] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *NeurIPS*, 29, 2016.
- [199] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *CVPR*, pages 4510–4520, 2018.
- [200] Florian Schroff, Dmitry Kalenichenko, and James Philbin. FaceNet: A unified embedding for face recognition and clustering. In *CVPR*, pages 815–823, 2015.
- [201] Christoph Schuhmann, Richard Vencu, Romain Beaumont, Robert Kaczmarczyk, Clayton Mullis, Aarush Katta, Theo Coombes, Jenia Jitsev, and Aran Komatsuzaki. Laion-400m: Open dataset of clip-filtered 400 million image-text pairs. *arXiv preprint*, 2021.

- [202] Soumyadip Sengupta, Jun-Cheng Chen, Carlos Castillo, Vishal M Patel, Rama Chelappa, and David W Jacobs. Frontal to profile face verification in the wild. In *WACV*, pages 1–9, 2016.
- [203] Zeyang Sha, Zheng Li, Ning Yu, and Yang Zhang. De-fake: Detection and attribution of fake images generated by text-to-image diffusion models. *arXiv preprint*, 2022.
- [204] Gregory Shakhnarovich, John W Fisher, and Trevor Darrell. Face recognition from long-term observations. In *ECCV*, 2002.
- [205] Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. Self-attention with relative position representations. *arXiv preprint*, 2018.
- [206] Hamid R Sheikh and Alan C Bovik. Image information and visual quality. *IEEE Transactions on Image Processing*, 15(2):430–444, 2006.
- [207] Yujun Shen, Bolei Zhou, Ping Luo, and Xiaoou Tang. Facefeat-GAN: a two-stage approach for identity-preserving face synthesis. *arXiv preprint*, 2018.
- [208] Yichun Shi and Anil K Jain. Probabilistic face embeddings. In *ICCV*, pages 6902–6911, 2019.
- [209] Yichun Shi and Anil K Jain. Probabilistic face embeddings. In *ICCV*, 2019.
- [210] Yichun Shi, Xiang Yu, Kihyuk Sohn, Manmohan Chandraker, and Anil K Jain. Towards universal representation learning for deep face recognition. In *CVPR*, pages 6817–6826, 2020.
- [211] Abhinav Shrivastava, Abhinav Gupta, and Ross Girshick. Training region-based object detectors with online hard example mining. In *CVPR*, pages 761–769, 2016.
- [212] Xiujun Shu, Xiao Wang, Xianghao Zang, Shiliang Zhang, Yuanqi Chen, Ge Li, and Qi Tian. Large-scale spatio-temporal person re-identification: Algorithms and benchmark. *TCSVT*, 2021.
- [213] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *ICML*, 2015.
- [214] Vladimir Somers, Alexandre Alahi, and Christophe De Vleeschouwer. Keypoint promptable re-identification. In *ECCV*, 2025.
- [215] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *ICLR*, 2021.
- [216] Yang Song, Conor Durkan, Iain Murray, and Stefano Ermon. Maximum likelihood training of score-based diffusion models. *NeurIPS*, 34, 2021.
- [217] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *NeurIPS*, 32, 2019.

- [218] Yang Song and Stefano Ermon. Improved techniques for training score-based generative models. *NeurIPS*, 33:12438–12448, 2020.
- [219] Joel Stehouwer, Amin Jourabloo, Yaojie Liu, and Xiaoming Liu. Noise modeling, synthesis and classification for generic object anti-spoofing. In *CVPR*, 2020.
- [220] Yukun Su, Guosheng Lin, Jinhui Zhu, and Qingyao Wu. Human interaction learning on 3d skeleton point clouds for video violence recognition. In *ECCV*, pages 74–90. Springer, 2020.
- [221] Tiancheng Sun, Jonathan T Barron, Yun-Ta Tsai, Zexiang Xu, Xueming Yu, Graham Fyffe, Christoph Rhemann, Jay Busch, Paul E Debevec, and Ravi Ramamoorthi. Single image portrait relighting. *TOG*, 2019.
- [222] Yifan Sun, Liang Zheng, Yi Yang, Qi Tian, and Shengjin Wang. Beyond part models: Person retrieval with refined part pooling (and a strong convolutional baseline). In *ECCV*, 2018.
- [223] Thomas Swearingen and Arun Ross. Lookalike disambiguation: Improving face identification performance at top ranks. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 10508–10515. IEEE, 2021.
- [224] Ying Tai, Yicong Liang, Xiaoming Liu, Lei Duan, Jilin Li, Chengjie Wang, Feiyue Huang, and Yu Chen. Towards highly accurate and stable face alignment for high-resolution videos. In *AAAI*, 2019.
- [225] Yaniv Taigman, Ming Yang, Marc’Aurelio Ranzato, and Lior Wolf. Deepface: Closing the gap to human-level performance in face verification. In *CVPR*, pages 1701–1708, 2014.
- [226] Yi Tay, Vinh Q Tran, Mostafa Dehghani, Jianmo Ni, Dara Bahri, Harsh Mehta, Zhen Qin, Kai Hui, Zhe Zhao, Jai Gupta, et al. Transformer memory as a differentiable search index. In *NeurIPS*, 2022.
- [227] Philipp Terhörst, Jan Niklas Kolf, Naser Damer, Florian Kirchbuchner, and Arjan Kuijper. Ser-fiq: unsupervised estimation of face image quality based on stochastic embedding robustness. in 2020 ieee. In *CVF Conference on Computer Vision and Pattern Recognition, CVPR*, pages 13–19, 2020.
- [228] Yonglong Tian, Lijie Fan, Kaifeng Chen, Dina Katabi, Dilip Krishnan, and Phillip Isola. Learning vision from models rivals learning vision from data. In *CVPR*, pages 15887–15898, 2024.
- [229] 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*, 2021.
- [230] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *ICML*, 2021.

- [231] Luan Tran, Xi Yin, and Xiaoming Liu. Disentangled representation learning GAN for pose-invariant face recognition. In *Proceeding of IEEE Computer Vision and Pattern Recognition*, pages 1415–1424, 2017.
- [232] Jonathan Tremblay, Aayush Prakash, David Acuna, Mark Brophy, Varun Jampani, Cem Anil, Thang To, Eric Cameracci, Shaad Boochoon, and Stan Birchfield. Training deep networks with synthetic data: Bridging the reality gap by domain randomization. In *CVPRW*, 2018.
- [233] Matthew A Turk, Alex Pentland, et al. Face recognition using eigenfaces. In *CVPR*, volume 91, pages 586–591, 1991.
- [234] Boris van Breugel, Trent Kyono, Jeroen Berrevoets, and Mihaela van der Schaar. Decaf: Generating fair synthetic data using causally-aware generative networks. *NeurIPS*, 34:22221–22233, 2021.
- [235] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9(11), 2008.
- [236] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017.
- [237] Paul Viola and Michael Jones. Rapid object detection using a boosted cascade of simple features. In *CVPR*, volume 1, pages I–I. Ieee, 2001.
- [238] Fangbin Wan, Yang Wu, Xuelin Qian, Yixiong Chen, and Yanwei Fu. When person re-identification meets changing clothes. In *CVPRW*, 2020.
- [239] Feng Wang, Xiang Xiang, Jian Cheng, and Alan Loddon Yuille. NormFace: L2 hypersphere embedding for face verification. In *Proceedings of the 25th ACM International Conference on Multimedia*, pages 1041–1049, 2017.
- [240] Hao Wang, Yitong Wang, Zheng Zhou, Xing Ji, Dihong Gong, Jingchao Zhou, Zhifeng Li, and Wei Liu. CosFace: Large margin cosine loss for deep face recognition. In *CVPR*, pages 5265–5274, 2018.
- [241] Jingya Wang, Xiatian Zhu, Shaogang Gong, and Wei Li. Transferable joint attribute-identity deep learning for unsupervised person re-identification. In *CVPR*, 2018.
- [242] Lei Wang, Bo Liu, Fangfang Liang, and Bincheng Wang. Hierarchical spatio-temporal representation learning for gait recognition. In *ICCV*, 2023.
- [243] Ming Wang, Xianda Guo, Beibei Lin, Tian Yang, Zheng Zhu, Lincheng Li, Shunli Zhang, and Xin Yu. Dygait: Exploiting dynamic representations for high-performance gait recognition. In *ICCV*, 2023.
- [244] Ruiping Wang, Shiguang Shan, Xilin Chen, and Wen Gao. Manifold-manifold distance with application to face recognition based on image set. In *CVPR*, 2008.

- [245] Sheng-Yu Wang, Oliver Wang, Richard Zhang, Andrew Owens, and Alexei A Efros. Cnn-generated images are surprisingly easy to spot... for now. In *CVPR*, 2020.
- [246] Tengfei Wang, Ting Zhang, Bo Zhang, Hao Ouyang, Dong Chen, Qifeng Chen, and Fang Wen. Pretraining is all you need for image-to-image translation. *arXiv preprint*, 2022.
- [247] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *CVPR*, 2018.
- [248] Xiaobo Wang, Shifeng Zhang, Shuo Wang, Tianyu Fu, Hailin Shi, and Tao Mei. Mis-classified vector guided softmax loss for face recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 12241–12248, 2020.
- [249] Yulin Wang, Rui Huang, Shiji Song, Zeyi Huang, and Gao Huang. Not all images are worth 16x16 words: Dynamic transformers for efficient image recognition. In *NeurIPS*, 2021.
- [250] Longhui Wei, Shiliang Zhang, Wen Gao, and Qi Tian. Person transfer gan to bridge domain gap for person re-identification. In *CVPR*, 2018.
- [251] Yandong Wen, Kaipeng Zhang, Zhifeng Li, and Yu Qiao. A discriminative feature learning approach for deep face recognition. In *ECCV*, 2016.
- [252] Frederick W Wheeler, Xiaoming Liu, and Peter H Tu. Multi-frame super-resolution for face recognition. In *2007 First IEEE International Conference on Biometrics: Theory, Applications, and Systems*, 2007.
- [253] Cameron Whitelam, Emma Taborsky, Austin Blanton, Brianna Maze, Jocelyn Adams, Tim Miller, Nathan Kalka, Anil K Jain, James A Duncan, Kristen Allen, et al. IARPA Janus Benchmark-B face dataset. In *CVPR Workshops*, pages 90–98, 2017.
- [254] Ross Wightman. Pytorch image models. <https://github.com/rwightman/pytorch-image-models>, 2019.
- [255] Haiyu Wu, Jaskirat Singh, Sicong Tian, Liang Zheng, and Kevin W Bowyer. Vec2face: Scaling face dataset generation with loosely constrained vectors. *arXiv preprint*, 2024.
- [256] Kan Wu, Houwen Peng, Minghao Chen, Jianlong Fu, and Hongyang Chao. Rethinking and improving relative position encoding for vision transformer. In *ICCV*, pages 10033–10041, 2021.
- [257] Tianxing Wu. Realtime glasses detection. <https://github.com/TianxingWu/realtime-glasses-detection>, 2022.
- [258] Wayne Wu, Chen Qian, Shuo Yang, Quan Wang, Yici Cai, and Qiang Zhou. Look at boundary: A boundary-aware face alignment algorithm. In *CVPR*, pages 2129–2138, 2018.
- [259] Yuxin Wu and Kaiming He. Group normalization. In *ECCV*, 2018.

- [260] Andre Brasil Vieira Wyzkowski and Anil K Jain. Synthetic latent fingerprint generator. In *WACV*, 2023.
- [261] Jiahao Xia, Weiwei Qu, Wenjian Huang, Jianguo Zhang, Xi Wang, and Min Xu. Sparse local patch transformer for robust face alignment and landmarks inherent relation learning. In *CVPR*, pages 4052–4061, 2022.
- [262] Taihong Xiao, Jiapeng Hong, and Jinwen Ma. Elegant: Exchanging latent encodings with GAN for transferring multiple face attributes. In *ECCV*, 2018.
- [263] Jiarui Xu, Shalini De Mello, Sifei Liu, Wonmin Byeon, Thomas Breuel, Jan Kautz, and Xiaolong Wang. GroupViT: Semantic segmentation emerges from text supervision. In *CVPR*, 2022.
- [264] Peng Xu and Xiatian Zhu. DeepChange: A large long-term person re-identification benchmark with clothes change. 2021.
- [265] Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.
- [266] Jiaolong Yang, Peiran Ren, Dongqing Zhang, Dong Chen, Fang Wen, Hongdong Li, and Gang Hua. Neural aggregation network for video face recognition. In *CVPR*, 2017.
- [267] Meng Yang, Pengfei Zhu, Luc Van Gool, and Lei Zhang. Face recognition based on regularized nearest points between image sets. In *FG*, 2013.
- [268] Qize Yang, Ancong Wu, and Wei-Shi Zheng. Person re-identification by contour sketch under moderate clothing change. *PAMI*, 2019.
- [269] Shuo Yang, Ping Luo, Chen-Change Loy, and Xiaoou Tang. Wider face: A face detection benchmark. In *CVPR*, pages 5525–5533, 2016.
- [270] Zhilin Yang. Xlnet: Generalized autoregressive pretraining for language understanding. *arXiv preprint*, 2019.
- [271] Bangpeng Yao and Li Fei-Fei. Modeling mutual context of object and human pose in human-object interaction activities. In *CVPR*, 2010.
- [272] Dingqiang Ye, Chao Fan, Jingzhe Ma, Xiaoming Liu, and Shiqi Yu. Biggait: Learning gait representation you want by large vision models. In *CVPR*, 2024.
- [273] Mang Ye, Jianbing Shen, Gaojie Lin, Tao Xiang, Ling Shao, and Steven CH Hoi. Deep learning for person re-identification: A survey and outlook. *IEEE transactions on pattern analysis and machine intelligence*, 44(6):2872–2893, 2021.
- [274] Dong Yi, Zhen Lei, Shengcai Liao, and Stan Z Li. Learning face representation from scratch. *arXiv preprint*, 2014.

- [275] Hongxu Yin, Arash Vahdat, Jose M Alvarez, Arun Mallya, Jan Kautz, and Pavlo Molchanov. A-ViT: Adaptive tokens for efficient vision transformer. In *CVPR*, pages 10809–10818, June 2022.
- [276] Xi Yin, Ying Tai, Yuge Huang, and Xiaoming Liu. FAN: Feature adaptation network for surveillance face recognition and normalization. In *Proceedings of the Asian Conference on Computer Vision*, pages 301–319, 2020.
- [277] Hao Yu and Jianxin Wu. A unified pruning framework for vision transformers. volume 66, pages 1–2, Apr. 2023.
- [278] Hong-Xing Yu, Wei-Shi Zheng, Ancong Wu, Xiaowei Guo, Shaogang Gong, and Jian-Huang Lai. Unsupervised person re-identification by soft multilabel learning. 2019.
- [279] Ning Yu, Larry S Davis, and Mario Fritz. Attributing fake images to gans: Learning and analyzing gan fingerprints. In *ICCV*, 2019.
- [280] Shijie Yu, Shihua Li, Dapeng Chen, Rui Zhao, Junjie Yan, and Yu Qiao. COCAS: A large-scale clothes changing person dataset for re-identification. In *CVPR*, 2020.
- [281] Junkun Yuan, Xinyu Zhang, Hao Zhou, Jian Wang, Zhongwei Qiu, Zhiyin Shao, Shaofeng Zhang, Sifan Long, Kun Kuang, Kun Yao, et al. Hap: Structure-aware masked image modeling for human-centric perception. In *NeurIPS*, 2023.
- [282] Li Yuan, Yunpeng Chen, Tao Wang, Weihao Yu, Yujun Shi, Zi-Hang Jiang, Francis EH Tay, Jiashi Feng, and Shuicheng Yan. Tokens-to-token vit: Training vision transformers from scratch on imagenet. In *ICCV*, 2021.
- [283] Guangtao Zhai and Xiongkuo Min. Perceptual image quality assessment: a survey. *Science China Information Sciences*, 63(11):211301, 2020.
- [284] Yunpeng Zhai, Shijian Lu, Qixiang Ye, Xuebo Shan, Jie Chen, Rongrong Ji, and Yonghong Tian. Ad-cluster: Augmented discriminative clustering for domain adaptive person re-identification. In *CVPR*, 2020.
- [285] Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li, and Yu Qiao. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Processing Letters*, 23(10):1499–1503, 2016.
- [286] Xiao Zhang, Rui Zhao, Yu Qiao, Xiaogang Wang, and Hongsheng Li. Adacos: Adaptively scaling cosine logits for effectively learning deep face representations. In *CVPR*, pages 10823–10832, 2019.
- [287] Xiao Zhang, Rui Zhao, Junjie Yan, Mengya Gao, Yu Qiao, Xiaogang Wang, and Hongsheng Li. P2sGrad: Refined gradients for optimizing deep face models. In *CVPR*, pages 9906–9914, 2019.
- [288] Yanyi Zhang, Xinyu Li, Chunhui Liu, Bing Shuai, Yi Zhu, Biagio Brattoli, Hao Chen, Ivan Marsic, and Joseph Tighe. Vidtr: Video transformer without convolutions. In *ICCV*, 2021.

- [289] Ziyuan Zhang, Luan Tran, Feng Liu, and Xiaoming Liu. On learning disentangled representations for gait recognition. *IEEE T-PAMI*, 44(1):345–360, 2020.
- [290] Weisong Zhao, Xiangyu Zhu, Kaiwen Guo, Xiao-Yu Zhang, and Zhen Lei. Grouped knowledge distillation for deep face recognition. *AAAI*, 2023.
- [291] Jinkai Zheng, Xinchun Liu, Xiaoyan Gu, Yaoqi Sun, Chuang Gan, Jiyong Zhang, Wu Liu, and Chenggang Yan. Gait recognition in the wild with multi-hop temporal switch. In *Proceedings of the 30th ACM International Conference on Multimedia*, 2022.
- [292] Jinkai Zheng, Xinchun Liu, Wu Liu, Lingxiao He, Chenggang Yan, and Tao Mei. Gait recognition in the wild with dense 3d representations and a benchmark. In *CVPR*, pages 20228–20237, 2022.
- [293] Jingxiao Zheng, Rajeev Ranjan, Ching-Hui Chen, Jun-Cheng Chen, Carlos D Castillo, and Rama Chellappa. An automatic system for unconstrained video-based face recognition. *IEEE Transactions on Biometrics, Behavior, and Identity Science*, 2(3):194–209, 2020.
- [294] Jingxiao Zheng, Ruichi Yu, Jun-Cheng Chen, Boyu Lu, Carlos D Castillo, and Rama Chellappa. Uncertainty modeling of contextual-connections between tracklets for unconstrained video-based face recognition. In *ICCV*, pages 703–712, 2019.
- [295] Liang Zheng, Liyue Shen, Lu Tian, Shengjin Wang, Jingdong Wang, and Qi Tian. Scalable person re-identification: A benchmark. In *ICCV*, 2015.
- [296] Tianyue Zheng and Weihong Deng. Cross-Pose LFW: A database for studying cross-pose face recognition in unconstrained environments. *Beijing University of Posts and Telecommunications, Tech. Rep*, 5:7, 2018.
- [297] Tianyue Zheng, Weihong Deng, and Jiani Hu. Cross-Age LFW: A database for studying cross-age face recognition in unconstrained environments. *CoRR*, abs/1708.08197, 2017.
- [298] Shaohua Zhou, Volker Krueger, and Rama Chellappa. Probabilistic recognition of human faces from video. *CVIU*, 91(1-2):214–245, 2003.
- [299] Yanjia Zhu, Hongxiang Cai, Shuhan Zhang, Chenhao Wang, and Yichao Xiong. Tinaface: Strong but simple baseline for face detection. *arXiv preprint*, 2020.
- [300] Zheng Zhu, Guan Huang, Jiankang Deng, Yun Ye, Junjie Huang, Xinze Chen, Jiagang Zhu, Tian Yang, Jiwen Lu, Dalong Du, and Jie Zhou. WebFace260M: A benchmark unveiling the power of million-scale deep face recognition. In *CVPR*, pages 10492–10502, 2021.
- [301] Jiaxuan Zhuo, Zeyu Chen, Jianhuang Lai, and Guangcong Wang. Occluded person re-identification. In *ICME*, 2018.
- [302] Hasib Zunair and A Ben Hamza. Synthesis of covid-19 chest x-rays using unpaired image-to-image translation. *Social network analysis and mining*, 11(1), 2021.