

FACE RECOGNITION: REPRESENTATION, INTRINSIC DIMENSIONALITY,  
CAPACITY, AND DEMOGRAPHIC BIAS

By

Sixue Gong

A DISSERTATION

Submitted to

Michigan State University

in partial fulfillment of the requirements  
for the degree of

Computer Science – Doctor of Philosophy

2021

# ABSTRACT

## FACE RECOGNITION: REPRESENTATION, INTRINSIC DIMENSIONALITY, CAPACITY, AND DEMOGRAPHIC BIAS

By

Sixue Gong

Face recognition is a widely adopted technology with numerous applications, such as mobile phone unlock, mobile payment, surveillance, social media and law enforcement. There has been tremendous progress in enhancing the accuracy of face recognition systems over the past few decades, much of which can be attributed to deep learning. Despite this progress, several fundamental problems in face recognition still remain unsolved. These problems include finding a salient representation, estimating intrinsic dimensionality, representation capacity, and demographic bias. With growing applications of face recognition, the need for an accurate, robust, compact and fair representation is evident.

In this thesis, we first develop algorithms to obtain practical estimates of intrinsic dimensionality of face representations, and propose a new dimensionality reduction method to project feature vectors from ambient space to intrinsic space. Based on the study in intrinsic dimensionality, we then estimate capacity of face representation, casting the face capacity estimation problem under the information theoretic framework of capacity of a Gaussian noise channel. Numerical experiments on unconstrained faces (IJB-C) provide a capacity upper bound of  $2.7 \times 10^4$  for FaceNet and  $8.4 \times 10^4$  for SphereFace representation at 1% FAR.

In the second part of the thesis, we address the demographic bias problem in face recognition systems where errors are lower on certain cohorts belonging to specific demographic groups. We propose two de-biasing frameworks that extract feature representations to improve fairness in face recognition. Experiments on benchmark face datasets (RFW, LFW, IJB-A, and IJB-C) show that our approaches are able to mitigate face recognition bias on various demographic groups (biasness drops from 6.83 to 5.07) as well as maintain the competitive performance (i.e., 99.75% on LFW, and 93.70% TAR@0.1% FAR on IJB-C). Lastly, we explore the global distribution of deep face

representations derived from correlations between image samples of within-class and cross-class to enhance the discriminativeness of face representation of each identity in the embedding space. Our new approach to face representation achieves state-of-the-art performance for both verification and identification tasks on benchmark datasets (99.78% on LFW, 93.40% on CPLFW, 98.41% on CFP-FP, 96.2% TAR@0.01% FAR and 95.3% Rank-1 accuracy on IJB-C). Since, the primary techniques we employ in this dissertation are not specific to faces only, we believe our research can be extended to other problems in computer vision, for example, general image classification and representation learning.

Copyright by  
SIXUE GONG  
2021



Dedicated to my parents, Qiaojie Gong and Yan Huang

## ACKNOWLEDGMENTS

The years at MSU have been an unforgettable experience that has been engraved on my heart, full of joys and sorrows, partings and meetings. It is yet a comfort to remember and to thank my teachers, colleagues, friends, and family whose influence contributed to this thesis.

First and foremost, I would like to thank my advisor, Anil K. Jain, for his extraordinary support, patience, guidance, and funding my entire research on face recognition. His suggestions and our discussions contributed immensely to this dissertation. Every summer semester, Dr. Jain encouraged and helped me to find internships, which enriched my working experience with industry and also inspired this dissertation research. Hitherto, he allowed me freedom to explore directions in my own research, and to manage my work-life balance by spending time on my hobbies. I have come to appreciate the wisdom of his way that has guided me effectively and safely through the process.

I owe many inspirational ideas to Vishnu Naresh Boddeti and Xiaoming Liu, who have been like co-advisors to me. It has been a privilege to work with Dr. Boddeti and Dr. Liu, which strengthened my research ability in the field of computer vision and deep learning. This thesis would not have been possible without Dr. Boddeti's and Dr. Liu's guidance and contributions. I deeply appreciate their enthusiasm for novel approaches and their genuinely positive attitude towards science and my research progress.

I am very thankful to Professor Yuan Zhang, and I learned a tremendous amount from her by taking the undergraduate courses of information theory and data compression from her my senior year. Working with Professor Zhang also fostered my interest in research. She later introduced me to the key Intelligent Information Processing Laboratory of the Chinese Academy of Sciences, where I had the opportunity to work with Professors Hu Han and Shiguang Shan. I highly value many useful ideas, comments and practical advice from Professor Han and Professor Shan. Had I not worked with Professor Han and Professor Shan, it is unlikely I would have ended up in the PRIP lab working with Dr. Jain.

I thank all of the members of the PRIP lab for participating in my research and providing valuable

feedback on my work at all times. I am very grateful for their friendship and support.

I thank all my friends in East Lansing. They are like my extended family here - the members of the PRIP lab (Debayan Deb, Joshua Engelsma, Yichun Shi, Kai Cao, Tarang Chugh, Inci M. Baytas, Vishesh Mistry, Divyansh Aggarwal, and Steven Grosz), Dipti Kamath, Manni Liu, Lan Wang, Rahul Dey, and Xiaoxue Wang - for parties, dinners, movies, games, and loving friendship over the years.

I thank my entire family for their love and support, especially my parents, Qiaojie Gong, Yan Huang, my aunt, Baolan Gong, and my cousin, Ruozhu Chen.

Finally, I would also like to thank those that have brought suffering to me. I have learnt important lessons in life and built resilience by embracing the adversity as a great teacher. It offered me valuable chance to gain treasured insights and wisdoms to have myself prepared for a better opportunity of success next time.

# TABLE OF CONTENTS

<b>LIST OF TABLES</b>	<b>xi</b>
<b>LIST OF FIGURES</b>	<b>xiv</b>
<b>LIST OF ALGORITHMS</b>	<b>xx</b>
<b>KEY TO ABBREVIATIONS</b>	<b>xxi</b>
<b>Chapter 1 Introduction: The Importance of Face Representation</b>	<b>1</b>
1.1 Automated Face Recognition	3
1.1.1 Input Data Source	3
1.1.2 Evaluation	5
1.1.3 Face Recognition Pipeline	7
1.2 Face Representation Extractors	8
1.3 Manifold of Face Representation	11
1.3.1 Intrinsic Dimensionality	11
1.3.2 Capacity	13
1.4 Bias in Face Recognition	15
1.5 Face Representation via Graph Neural Network	18
1.6 Thesis Contributions	20
1.7 Thesis Structure	22
<b>Chapter 2 The Intrinsic Dimensionality of Face Representation</b>	<b>23</b>
2.1 Intrinsic Dimensionality	24
2.2 Dimensionality Reduction	25
2.3 Our Approach	25
2.3.1 Estimating Intrinsic Dimensionality	26
2.3.2 Estimating Intrinsic Sapce	30
2.4 Experiments	31
2.4.1 Intrinsic Dimensionality Estimation	31
2.4.2 Intrinsic Space Mapping	36
2.5 Conclusion	43
<b>Chapter 3 The Capacity of Face Representation</b>	<b>44</b>
3.1 Related Work	46
3.2 Capacity of Face Representations	48
3.2.1 Face Representation Model	48
3.2.2 Estimating Uncertainties in Representations	50
3.2.3 Manifold Approximation	54
3.2.4 Decision Theory and Model Capacity	55
3.3 Numerical Experiments	58
3.3.1 Two-Dimensional Toy-Example	59

3.3.2	Datasets and Face Representation Model . . . . .	60
3.3.3	Face Recognition Performance . . . . .	61
3.3.4	Face Representation Capacity . . . . .	63
3.3.5	Ablation Studies . . . . .	66
3.4	Conclusion . . . . .	68
<b>Chapter 4</b>	<b>The Bias in Face Recognition . . . . .</b>	<b>70</b>
4.1	Fairness Learning and De-biasing Algorithms . . . . .	71
4.2	Problem Definition . . . . .	72
4.3	Jointly De-biasing Face Recognition and Demographic Attribute Estimation . . . .	73
4.3.1	Adversarial Learning and Disentangled Representation . . . . .	75
4.3.2	Methodology . . . . .	75
4.3.2.1	Algorithm Design . . . . .	75
4.3.2.2	Network Architecture . . . . .	77
4.3.2.3	Adversarial Training and Disentanglement . . . . .	78
4.3.3	Experiments . . . . .	80
4.3.3.1	Datasets and Pre-processing . . . . .	80
4.3.3.2	Implementation Details . . . . .	81
4.3.3.3	De-biasing Face Verification . . . . .	81
4.3.3.4	De-biasing Demographic Attribute Estimation . . . . .	83
4.3.3.5	Analysis of Disentanglement . . . . .	85
4.3.3.6	Face Verification on Public Testing Datasets . . . . .	87
4.3.3.7	Distributions of Scores . . . . .	88
4.4	Mitigating Face Recognition Bias via Group Adaptive Classifier . . . . .	89
4.4.1	Adaptive Neural Networks . . . . .	91
4.4.2	Methodology . . . . .	93
4.4.2.1	Overview . . . . .	93
4.4.2.2	Adaptive Layer . . . . .	93
4.4.2.3	Automation Module . . . . .	96
4.4.2.4	De-biasing Objective Function . . . . .	96
4.4.3	Experiments . . . . .	97
4.4.3.1	Results on RFW Protocol . . . . .	98
4.4.3.2	Results on Gender and Race Groups . . . . .	104
4.4.3.3	Analysis on Intrinsic Bias and Data Bias . . . . .	106
4.4.3.4	Results on Standard Benchmark Datasets . . . . .	107
4.4.3.5	Visualization and Analysis on Bias of FR . . . . .	108
4.4.3.6	Network Complexity and FLOPs . . . . .	109
4.5	Demographic Estimation . . . . .	109
4.6	Conclusion . . . . .	111
<b>Chapter 5</b>	<b>Adversarial Face Representation Learning via Graph Classification . . .</b>	<b>113</b>
5.1	Adversarial Learning and Graph Classification with GNN . . . . .	113
5.2	Our Approach . . . . .	115
5.2.1	Overall Framework . . . . .	115
5.2.2	Graph Construction . . . . .	117

5.2.3	Discriminator and Adversarial Learning . . . . .	118
5.2.4	Network Training . . . . .	120
5.3	Experiments . . . . .	122
5.3.1	Datasets and Implementation Details . . . . .	122
5.3.2	Ablation Study . . . . .	123
5.3.3	Comparisons with SOTA Methods . . . . .	125
5.3.4	Analysis on Feature Distribution . . . . .	128
5.4	Concluding Remarks . . . . .	130
<b>Chapter 6</b>	<b>Summary and Future Work . . . . .</b>	<b>131</b>
<b>APPENDIX</b>	<b>. . . . .</b>	<b>137</b>
<b>BIBLIOGRAPHY</b>	<b>. . . . .</b>	<b>139</b>

## LIST OF TABLES

Table 2.1	Intrinsic Dimensionality: Graph Distance [1] . . . . .	35
Table 2.2	Intrinsic Dimensionality: KNN [2] . . . . .	35
Table 2.3	Intrinsic Dimensionality: IDEA [3] . . . . .	36
Table 2.4	LFW Face Verification for SphereFace Embedding . . . . .	38
Table 2.5	DeepMDS Training Methods (TAR @ 0.1% FAR) . . . . .	40
Table 3.1	Capacity of Two-Dimensional Toy Example at 1% FAR . . . . .	59
Table 3.2	Face Recognition Results for FaceNet, SphereFace and State-of-the-Art (The state-of-the-art face representation models are not available in the public domain) .	63
Table 3.3	Capacity of Face Representation Model at 1% FAR . . . . .	64
Table 3.4	IJB-C Capacity at 1% FAR Across Intra-Class Uncertainty . . . . .	66
Table 3.5	IJB-C Capacity at 1% FAR Across Manifold Support . . . . .	68
Table 4.1	Statistics of training and testing datasets used in the paper. . . . .	80
Table 4.2	Biasness of Face Recognition and Demographic Attribute Estimation. . . . .	84
Table 4.3	Demographic Classification Accuracy (%) by face features. . . . .	87
Table 4.4	Face Verification Accuracy (%) on RFW dataset. . . . .	87
Table 4.5	Verification Performance on LFW, IJB-A, and IJB-C. . . . .	88
Table 4.6	Performance comparison with SOTA on the RFW protocol [4]. The results marked by (*) are directly copied from [5]. . . . .	99
Table 4.7	Ablation of adaptive strategies on the RFW protocol [4]. . . . .	102
Table 4.8	Ablation of CNN depths and demographics on RFW protocol [4]. . . . .	102
Table 4.9	Ablations on $\lambda$ on RFW protocol (%). . . . .	103
Table 4.10	Verification Accuracy (%) of 5-fold cross-validation on 8 groups of RFW [4].	103

Table 4.11	Ablations on the automation module on RFW protocol (%).	104
Table 4.12	Statistics of dataset folds in the cross-validation experiment.	104
Table 4.13	Verification (%) on gender groups of IJB-C (TAR @ 0.1% FAR).	105
Table 4.14	Verification accuracy (%) on the RFW protocol [4] with varying race/ethnicity distribution in the training set.	106
Table 4.15	Verification performance on LFW, IJB-A, and IJB-C. [Key: <b>Best</b> , <i>Second</i> , <u>Third Best</u> ]	107
Table 4.16	Distribution of ratios between minimum inter-class distance and maximum intra-class distance of face features in 4 race groups of RFW. GAC exhibits higher ratios, and more similar distributions to the reference.	108
Table 4.17	Network complexity and inference time.	109
Table 4.18	Gender distribution of the datasets for gender estimation.	110
Table 4.19	Race distribution of the datasets for race estimation.	111
Table 4.20	Age distribution of the datasets for age estimation	111
Table 5.1	Verification performance (%) of different vertex feature matrices of oracle graphs. A bigger $r$ tolerates more intra-class variations, while a small $r$ , $\mathbf{f}_i^c$ , or $\mathbf{f}_i^p$ strive for minimal intra-class variation. A balance between the learning capability and ideal representations performs the best ( $r = 0.7$ ).	123
Table 5.2	Verification performance (%) of different adjacency matrices of generated graphs.	125
Table 5.3	Verification performance (%) of different $\lambda$ and $\mu$ .	125
Table 5.4	Verification accuracy (%) of our model and SOTA methods on LFW, CPLFW, and CFP-FP. The results marked by (*) are re-implemented by ArcFace [6]. All other baseline results are reported by their respective papers. [Keys: Red: Best, Blue: Second best]	126



Table 5.5 Comparisons of verification performance with SOTA methods on IJB-A, IJB-B, and IJB-C. The evaluation is measured by TAR (%), True Acceptance Rate, at a certain FAR, False Acceptance Rate. For IJB-A, FAR = 0.1%; for IJB-B and IJB-C, FAR = 0.01%. The decimal precision of TAR varies among those reported by SOTA methods. Results reported in this table are unified to one decimal place (0.1). All baseline results are reported by their respective papers. [Keys: Red: Best, Blue: Second best]	127
Table 5.6 Comparisons of face identification performance (%) on the IJB-C dataset (close-set).	128

## LIST OF FIGURES

Figure 1.1	Example images from different datasets. MS-Celeb-1M, CASIA, and LFW contain only still images. IJB-A, IJB-B and IJB-C include both still images and video frames. Three subjects are selected from each dataset, and each row contains images belonging to one subject. . . . .	5
Figure 1.2	A typical pipeline of feature-based FR frameworks comprises of face detection, alignment, normalization, feature extraction, and feature matching. While each of these components affect the performance of FR systems, in this thesis, we focus on the representation function of feature extraction that maps a high-dimensional normalized face image to a $d$ -dimensional vector representation. . . . .	8
Figure 1.3	The error tradeoff characteristics in the form of false non-match rates (FNMR = 1 - TAR) vs. false match rates (FMR = FAR) for one commercial FR algorithm verifying mugshot images, reported by the NIST (National Institute of Standards and Technology) Face recognition Vendor Test [7]. The FMR estimates are computed on impostor pairs of face images with same gender and same race. Each symbol (circle, triangle, square) corresponds to a fixed threshold - their vertical and horizontal displacements reveal, respectively, differences in FNMR and FMR between demographic groups [7]. . . . .	15
Figure 1.4	False positive differentials in verification algorithms provided to NIST. The dots give the false match rates for same-sex and same-race impostor comparisons. The threshold is set for each algorithm to give FMR = 0.0001 on white males (the purple dots in the right hand panel). The algorithms are sorted in order of worst case FMR [7]. . . . .	16
Figure 2.1	<b>Intrinsic Dimension:</b> Our approach is based on two observations: (a) Graph induced geodesic distance between images is able to capture the topology of the image representation manifold more reliably. As an illustration, we show the graph edges for the surface of a unitary hypersphere and a face manifold of ID two, embedded within a 3- <i>dim</i> space. (b) The distribution of the geodesic distances (for distance $r_{max} - 2\sigma \leq r \leq r_{max}$ , where $r_{max}$ is the distance at the mode) has been empirically observed [1] to be similar across different topological structures with the same intrinsic dimensionality. The plot shows the distance distribution for a face representation, unitary hypersphere and a Gaussian distribution of ID two embedded within 3- <i>dim</i> space. [8] . . . . .	27
Figure 2.2	<b>DeepMDS Mapping:</b> A DNN based non-linear mapping is learned to transform the ambient space to a plausible intrinsic space. The network is optimized to preserve distances between pairs of points in the ambient and intrinsic space. . .	30

Figure 2.3	<b>Intrinsic Dimensionality:</b> (a) Geodesic distance distribution, and (b) global minimum of RMSE. . . . .	32
Figure 2.4	Distribution of geodesic distances for different representation models and datasets. . . . .	33
Figure 2.5	$\log \frac{\hat{p}_{\mathcal{M}(r)}}{\hat{p}_{\mathcal{M}(r_{max})}}$ vs $\log \frac{r}{r_{max}}$ plots as we vary number of neighbors $k$ for sphereface representation model on different datasets. . . . .	34
Figure 2.6	Intrinsic Dimensionality of Swiss Roll . . . . .	36
Figure 2.7	<b>Swiss Roll:</b> (a) the original 2000 points from the swiss roll manifold, (b) the 2- <i>dim</i> intrinsic space estimated by Isomap, and (3) the 2- <i>dim</i> intrinsic space estimated by our proposed method DeepMDS. In both cases, the blue and black points, and correspondingly green and red points, are close together in both the intrinsic and ambient space. . . . .	37
Figure 2.8	<b>DeepMDS:</b> Face Verification on IJB-C [9] (TAR @ 0.1% FAR in legend) for the (a) FaceNet-128, (b) FaceNet-512 and (c) SphereFace embeddings. . . . .	38
Figure 2.9	<b>DeepMDS:</b> Face Verification on LFW (BLUFR) dataset for the (a) FaceNet-128, (b) FaceNet-512 and (c) SphereFace embeddings. . . . .	39
Figure 2.10	<b>PCA:</b> Face Verification on IJB-C and LFW (BLUFR) dataset for the (a) FaceNet-128, (b) FaceNet-512 and (c) SphereFace embeddings. . . . .	40
Figure 2.11	<b>Isomap:</b> Face Verification on IJB-C and LFW (BLUFR) dataset for the (a) FaceNet-128, (b) FaceNet-512 and (c) SphereFace embeddings. . . . .	41
Figure 2.12	ROC curve on LFW and IJB-C datasets for the Inception ResNet V1 [10] model trained with different embedding dimensionality on the CASIA-WebFace [11] dataset. . . . .	42
Figure 2.13	<b>Denoising Autoencoder:</b> Face Verification on IJB-C and LFW (BLUFR) dataset for the (a) FaceNet-128, (b) FaceNet-512 and (c) SphereFace embeddings. . . . .	42
Figure 3.1	An illustration of the geometrical structure of our capacity estimation problem: a low-dimensional manifold $\mathcal{M} \in \mathbb{R}^m$ embedded in high dimensional space $\mathcal{P} \in \mathbb{R}^p$ . On this manifold, all the faces lie inside the population hyper-ellipsoid and the embedding of images belonging to each identity or a class are clustered into their own class-specific hyper-ellipsoids. The capacity of this manifold is the number of identities (class-specific hyper-ellipsoids) that can be packed into the population hyper-ellipsoid within an error tolerance or amount of overlap. . . . .	45

Figure 3.2 **Overview of Face Representation Capacity Estimation:** We cast the capacity estimation process in the framework of the sphere packing problem on a low-dimensional manifold. To generalize the sphere packing problem, we replace spheres by hyper-ellipsoids, one per class (subject). Our approach involves three steps; (i) Unfolding and mapping the manifold embedded in high-dimensional space onto a low-dimensional space. (ii) *Teacher-Student* model to obtain explicit estimates of the uncertainty (noise) in the embedding due to data as well as the parameters of the representation, and (iii) The uncertainty estimates are leveraged to approximate the density manifold via multi-variate normal distributions (to keep the problem and its analysis tractable), which in turn facilitates an empirical estimate of the capacity of the *teacher* face representation as a ratio of hyper-ellipsoidal volumes. 49

Figure 3.3 **Manifold Unfolding:** A DNN based non-linear mapping is learned to unfold and project the population manifold into a lower dimensional space. The network is optimized to preserve the geodesic distances between pairs of points in the high and low dimensional space. . . . . 50

Figure 3.4 **Decision Theory and Capacity:** We illustrate the relation between capacity and the discriminant function corresponding to a nearest neighbor classifier. **Left:** Depiction of the notion of decision boundary and probability of false accept between two identical one dimensional Gaussian distributions. Shannon’s definition of capacity corresponds to the decision boundary being one standard deviation away from the mean. **Right:** Depiction of the decision boundary induced by the discriminant function of nearest neighbor classifier. Unlike in the definition of Shannon’s capacity, the size of the ellipsoidal decision boundary is determined by the maximum acceptable false accept rate. The probability of false acceptance can be computed through the cumulative distribution function of a  $\chi^2(r^2, d)$  distribution. 55

Figure 3.5 **Sample Representation Space:** Illustration of a two-dimensional space where the underlying population and class-specific representations (we show four classes) are 2-D Gaussian distributions (solid ellipsoids). Samples from the classes (colored ★) are utilized to obtain estimates of this underlying population and class-specific distributions (solid lines). As a comparison, the support of the samples in the form of a convex hull are also shown (dashed lines). . . . . 59

Figure 3.6 Face recognition performance of the *original* and *student* models on different datasets. We report the face verification performance of both FaceNet and SphereFace face representations, (a) LFW evaluated through the BLUFR protocol, (b) IJB-A, (c) IJB-B, and (d) IJB-C evaluated through their respective matching protocol. . . . 63

Figure 3.7 Capacity estimates across different datasets for the (a) FaceNet [12] and (b) SphereFace [13] representations as function of different false accept rates. Under the limit, the capacity tends to zero as the FAR tends to zero. Similarly, the capacity tends to  $\infty$  as the FAR tends to 1.0. (c) Logarithmic values of capacity on different datasets versus the corresponding TAR @ 0.1% FAR. . . . . 65

Figure 3.8	Example images of classes that correspond to different sizes of the class-specific hyper-ellipsoids, based on the SphereFace representation, for different datasets considered. <i>Top Row</i> : Images of the class with the largest class-specific hyper-ellipsoid for each database. Notice that in the case of a database with predominantly frontal faces (LFW), large variations in facial appearance lead to the greatest uncertainty in the class representation. On more challenging datasets (IJB-B, IJB-C), the face representation exhibits most uncertainty due to pose variations. <i>Bottom Row</i> : Images of the class with the smallest class-specific hyper-ellipsoid for each database. As expected, across all the datasets, frontal face images with the minimal change in appearance result in the least amount of uncertainty in the class representation. . . . .	67
Figure 4.1	Methods to learn different tasks simultaneously. Solid lines are typical feature flow in CNN, while dash lines are adversarial losses. . . . .	74
Figure 4.2	Overview of the proposed De-biasing face (DebFace) network. DebFace is composed of three major blocks, <i>i.e.</i> , a shared feature encoding block, a feature disentangling block, and a feature aggregation block. The solid arrows represent the forward inference, and the dashed arrows stand for adversarial training. During inference, either DebFace-ID ( <i>i.e.</i> , $\mathbf{f}_{ID}$ ) or DemoID can be used for face matching given the desired trade-off between biasness and accuracy. . . . .	76
Figure 4.3	Face Verification AUC (%) on each demographic cohort. The cohorts are chosen based on the three attributes, <i>i.e.</i> , gender, age, and race. To fit the results into a 2D plot, we show the performance of male and female separately. Due to the limited number of face images in some cohorts, their results are gray cells. . . . .	82
Figure 4.4	The overall performance of face verification AUC (%) on gender, age, and race.	83
Figure 4.5	Classification accuracy (%) of demographic attribute estimations on faces of different cohorts, by DebFace and the baselines. For simplicity, we use DebFace-G, DebFace-A, and DebFace-R to represent the gender, age, and race classifier of DebFace. . . . .	84
Figure 4.6	The distribution of face identity representations of BaseFace and DebFace. Both collections of feature vectors are extracted from images of the same dataset. Different colors and shapes represent different demographic attributes. Zoom in for details. . . . .	85
Figure 4.7	Reconstructed Images using Face and Demographic Representations. The first row is the original face images. From the second row to the bottom, the face images are reconstructed from 2) BaseFace; 3) DebFace-ID; 4) DebFace-G; 5) DebFace-R; 6) DebFace-A. Zoom in for details. . . . .	85
Figure 4.8	The percentage of false accepted cross race or age pairs at 1% FAR. . . . .	87

Figure 4.9	BaseFace and DebFace distributions of the similarity scores of the imposter pairs across homogeneous versus heterogeneous gender, age, and race categories. . . . .	89
Figure 4.10	(a) Our proposed group adaptive classifier (GAC) automatically chooses between non-adaptive (“N”) and adaptive (“A”) layer in a multi-layer network, where the latter uses demographic-group-specific kernel and attention. (b) Compared to the baseline with the 50-layer ArcFace backbone, GAC improves face verification accuracy in most groups of RFW dataset [4], especially under-represented groups, leading to mitigated FR bias. GAC reduces biasness from 1.11 to 0.60. . . . .	90
Figure 4.11	A comparison of approaches in adaptive CNNs. . . . .	92
Figure 4.12	Overview of the proposed GAC for mitigating FR bias. GAC contains two major modules: the adaptive layer and the automation module. The adaptive layer consists of adaptive kernels and attention maps. The automation module is employed to decide whether a layer should be adaptive or not. . . . .	94
Figure 4.13	ROC of (a) baseline and (b) GAC evaluated on all pairs of RFW. . . . .	100
Figure 4.14	8 false positive and false negative pairs on RFW given by the baseline but successfully verified by GAC. . . . .	101
Figure 4.15	(a) For each of the three $\tau$ in automatic adaptation, we show the average similarities of pair-wise demographic kernel masks, <i>i.e.</i> , $\bar{\theta}$ , at 1-48 layers (y-axis), and 1-15K training steps (x-axis). The number of adaptive layers in three cases, <i>i.e.</i> , $\sum_1^{48}(\bar{\theta} > \tau)$ at $15K^{th}$ step, are 12, 8, and 2, respectively. (b) With two race groups (White, Black in PCSO [14]) and two models (baseline, GAC), for each of the four combinations, we compute pair-wise correlation of face representations using any two of 1K subjects in the same race, and plot the histogram of correlations. GAC reduces the difference/bias of two distributions. . . . .	102
Figure 4.16	The first row shows the average faces of different groups in RFW. The next two rows show gradient-weighted class activation heatmaps [15] at the $43^{th}$ convolutional layer of the GAC and baseline. The higher diversity of heatmaps in GAC shows the variability of parameters in GAC across groups. . . . .	104
Figure 4.17	Demographic Attribute Classification Accuracy on each group. The red dashed line refers to the average accuracy on all images in the testing set. . . . .	110

Figure 5.1 (a) In GANs, during training an image generator gradually produces higher quality faces so that a CNN-based discriminator could not distinguish fake from real faces. (b) Analogously, given input faces, our embedding network for face recognition learns to extract discriminative features and connect features as a graph, with the goal that a graph neural network (GNN)-based discriminator could not distinguish generated graphs from oracle graphs — the graph of ideal face representations. During inference, our embedding network can extract more discriminative features that form oracle-like graph, just like GAN’s generator synthesizes photo-realistic faces. 114

Figure 5.2 Overview of the proposed adversarial face representation learning via graph classification. Solid arrows present forward pass, and dashed arrows denote backward propagation. The training alternates between  $E(\cdot)$  and  $D(\cdot)$ . For the shared inference (solid blue arrows), a set of face images are first taken by the embedding network  $E(\cdot)$  to extract feature representations. These feature vectors are then converted into graph structure by the graph constructor  $G(\cdot)$  in which an oracle graph and a generated graph are constructed. During the training of  $D(\cdot)$  (yellow arrows), the two types of graphs are received by the graph discriminator  $D(\cdot)$  that is required to make predictions on the category of the graphs.  $D(\cdot)$  is then updated based on the gradient sent back from the loss function  $\mathcal{L}_D$ . In the course of training on  $E(\cdot)$  (red arrows), only generated graphs are delivered to  $D(\cdot)$ , and  $E(\cdot)$  receives feedbacks from  $\mathcal{L}_A$  whose goal is to drive  $D(\cdot)$  to make errors on generated graphs. . . . . 116

Figure 5.3 *Construction of generated graph and oracle graph.* In this example, the input image set comprises 9 images of 3 subjects, 3 images per subject. The image of each subject is surrounded by a circle with a unique color, indicating its identity. Each image is projected to a point in the 2D Euclidean feature space. The following graphs are constructed: (a) a generated graph, where each vertex  $v_i$  is represented by its feature vector, with a directed edge from  $v_i$  to  $v_j$  if  $v_j$  is one of the top 2 nearest neighbors of  $v_i$ ; (b) an oracle graph created by center points, where each vertex is represented by the mean vector of its identity with a bidirectional edge connecting two vertices of the same subject; (c) a radius constraint is used to allow tolerable intra-subject variations, where vertices move towards center directions (denoted by dashed arrows) to meet the radius requirement. For the vertex within the radius, the left most one in this example, it stays the same. (d) an oracle graph controlled by  $r$ , where the distance of each vertex to its center is reduced by the ratio of  $r$ , with a bidirectional edge connecting two vertices of the same subject. . . . . 119

Figure 5.4 Two examples of generated graphs being updated by adversarial learning at 4 instances during the training process. . . . . 129

Figure 5.5 t-SNE visualization of the face representations in a 2D space. Each identity is represented by a unique color. The initial face representations extract from CurricularFace, and the updated representations learned via adversarial graph classification are shown in (a) and (b), respectively. . . . . 130

## LIST OF ALGORITHMS

Algorithm 1 Face Representation Capacity Estimation . . . . .	58
---	----



## KEY TO ABBREVIATIONS

### Acronyms / Abbreviation

FR	Face Recognition
SOTA	State-of-the-art
CCTV	Closed Circuit Television
ROC	Receiving Operating Characteristic
TAR	True Acceptance Rate
FAR	False Acceptance Rate
CMC	Cumulative Match Characteristic
DIR	Detection & Identification Rate
$k$ -NN	$k$ -nearest neighbors
PCA	Principal Component Analysis
LDA	Linear Discriminant Analysis
LBP	Local Binary Patterns
LQP	Local Quantized Patterns
HOG	Histogram of Oriented Gradients
SIFT	Scale Invariant Feature Transform
DNN	Deep Neural Network
CNN	Convolutional Neural Network
IND	Intrinsic Dimensionality
GAC	Group Adaptive Classifier
MDS	Multidimensional Scaling
RMSE	Root Mean Squared Error
GAN	Generative Adversarial Network
GNN	Graph Neural Network

# Chapter 1

## Introduction: The Importance of Face Representation

In biometrics and computer vision communities, face recognition (FR) has emerged as one of the major research fields focusing on the design of algorithms that can automatically authenticate people's identities based on their digital face images. With the rapid proliferation of face images or "selfies" on social media websites, such as Facebook, Twitter, and Instagram, researchers in the FR community have access to abundant images and videos of human face, which has rapidly accelerated the development of FR systems and extended its applications. For example, FR systems are widely adopted for security-related applications (*e.g.*, access control, surveillance systems), forensic applications (criminal identity verification), and entertainment applications on desktops and mobile devices, for example, mobile apps for face photo editing. As a convenient authentication tool, FR requires minimal interaction with users and can even operate under uncontrolled environments and at a distance [16]. Compared to other biometric traits (*i.e.*, iris, fingerprints, voice, etc.), in addition to identity, a human face image contains several useful information, including demographics (gender, age, race/ethnicity), facial expression, and emotion cues. Such rich information, on the other hand, can undermine the reliability of FR systems. This is because some facial characteristics, such as facial expressions, can deform crucial face features, and lead to large intra-person facial variations

that are difficult to compensate. One of the key steps in any FR system is to impart robustness towards challenges caused by variations in face images by extracting salient facial features. Instead of a raw face image, a vector of representative features, also referred to as a *face representation*, is used to distinguish the identity. A good representation method is capable of reducing the intra-person variations while maintaining or even enhancing inter-person differences.

When using *deep learning* models [6, 13, 17–19] to extract face representations, current state-of-the-art (SOTA) FR systems claim to surpass human capability in certain scenarios [20]. Despite this tremendous progress, some fundamental questions in face representation learning remain:

- How compact can the representation be without any loss in recognition performance?
- Given a face representation, how many identities can it resolve? Or in other words, what is the capacity of a given face representation?
- Does a FR system generate representations which is equally discriminable for faces in different demographic groups?
- Is it possible to enhance the saliency of face representation for every subject in a target population by means of the distance distribution in the embedding space?

First, a scientific basis for estimating the compactness and the capacity of a given face representation will not only benefit the evaluation and comparison of different representation methods, but will also benefit the development of compact face representations (small template size) with high search efficiency and establish an upper bound on the scalability of an automatic FR system. Second, FR systems are known to exhibit biased performance against certain demographic groups [7, 14, 21]. Given the importance of automated FR-driven decisions, deploying biased FR systems especially for law enforcement is potentially unethical [22]. Some state and local governments in the United States have curtailed the use of face recognition for these reasons, with cities including San Francisco and Boston enacting their own bans. It is necessary to develop fair and unbiased FR systems to avoid their negative societal impact. While reducing the bias among demographic groups is important, these groups are pre-defined based on one's demographic attributes. It is also crucial to improve the representation for each individual regardless of his or her gender, age, and race/ethnicity. The main

goal of this thesis is to develop practical tools to reason about the compactness and capacity of a given face representation, and design algorithms for fairer and more discriminative face representations of each identity in every demographic group. To begin with, we first give a brief background on automated face recognition. Then we introduce the motivation, goals, and problem domains of each work on automated face recognition addressed in this dissertation in Sec. 1.3 to Sec. 1.5, separately. We also summarize the contributions and the structure of this thesis at the end of the chapter.

## 1.1 Automated Face Recognition

### 1.1.1 Input Data Source

As a visual pattern recognition task, FR can be performed on a variety of input data source, such as 1) a single 2D image, 2) a set of 2D images (video frames), and 3) 3D face images. A single 2D image is often used as the input of face verification systems. In some scenarios, for example, in a surveillance environment, a clip of video captured by CCTV systems is taken as the input of FR systems. Although multiple frames in a video clip provide rich information at different time stamps, unconstrained FR in video-surveillance settings is still a challenging problem, because video frames tend to be of poor quality with motion blur and unfavorable viewing angles. 3D sensors, including depth sensors, may provide extra information and help improve the accuracy of FR, but they are expensive with relatively larger acquisition time. This thesis focuses on the more commonly deployed scenario where a single 2D image is the input data.

A key aspect of deep learning based FR algorithms is the training data used to learn face representations. Data collection and annotation are extremely important for supervised face representation learning. Widely adopted publicly available face datasets used in this thesis for training and testing face representation models are listed below. Examples of face images in these datasets are shown in Fig. 1.1.

**CASIA WebFace [11]:** A collection of labeled images downloaded from the web (based on names of famous personalities as keywords) popular for training deep neural networks. It consists

of 494,414 images across 10,575 subjects, with an average of about 500 face images per subjects. This dataset is primarily used for training face representation models.

**MS-Celeb-1M [23]:** The first released version of this dataset contained around 10 million images of 100K celebrities. About 100 images were retrieved for each identity by the Bing search engine using the celebrity’s name. With no filtering of the retrieved images, the quality of the dataset is greatly muddled by label noise, duplicated images, and non-face images, making it hard to be used directly for representation learning [24]. For this reason, there have been several cleaned versions of the dataset ([6, 13, 25, 26]). In this thesis, the version of [6] is used as the dataset for training FR models, which contains 5,822,653 images of 85,742 subjects.

**LFW [27]:** 13,233 face images of 5,749 subjects, downloaded from the web. These images exhibit limited variations in pose, illumination, and expression, since only faces that could be detected by the Viola-Jones face detector [28] were included in the dataset. One limitation of this dataset is that only 1,680 subjects among the total of 5,749 subjects have more than one face image.

**IJB-A [29]:** IARPA Janus Benchmark-A (IJB-A) contains 500 subjects with a total of 25,813 images (5,399 still images and 20,414 video frames), an average of 51 images per subject. Compared to the LFW and CASIA datasets, the IJB-A dataset is more challenging due to the presence of: i) large pose variations making it difficult to detect all the faces using a commodity face detector, ii) a mix of images and videos, and iii) wider geographical variation of subjects. The face locations are provided with the IJB-A dataset (and used in our experiments in this thesis when needed).

**IJB-B [30]:** IARPA Janus Benchmark-B (IJB-B) dataset is a superset of the IJB-A dataset consisting of 1,845 subjects with a total of 76,824 images (21,798 still images and 55,026 video frames from 7,011 videos), an average of 41 images per subject. Images in this dataset are labeled with ground truth bounding boxes and other covariate meta-data such as occlusions, facial hair and skin tone. A key motivation for the IJB-B dataset is to make the face dataset less constrained compared to IJB-A dataset and have a more uniform geographical distribution of subjects across the globe.

**IJB-C [9]:** IARPA Janus Benchmark-C (IJB-C) dataset consists of 3,531 subjects with a total



Figure 1.1 Example images from different datasets. MS-Celeb-1M, CASIA, and LFW contain only still images. IJB-A, IJB-B and IJB-C include both still images and video frames. Three subjects are selected from each dataset, and each row contains images belonging to one subject.

of 31,334 (21,294 face and 10,040 non-face) still images and 11,779 videos (117,542 frames), an average of 39 images per subject. This dataset emphasizes faces with full pose variations, occlusions and diversity of subject occupation and geographical origin. Images in this dataset are labeled with ground truth bounding boxes and other covariates such as occlusions, facial hair and skin tone.

### 1.1.2 Evaluation

There are many applications where FR techniques are successfully used to perform a specific task. Among those tasks, two primary tasks are considered to evaluate an FR system:

- Verification (authentication) – one to one match.
- Identification (recognition) – one to many match.

**Verification.** The task of verification generally aims at identity authentication with user interaction, to verify if a given face matches the identity that is claimed. To evaluate a verification

system, face images are first divided into two groups: 1) *genuine* group where people gain access using their own identity; 2) *impostor* group where people gain access using false identities. A face image is compared with other face images from genuine group and impostor group, respectively, which corresponds to genuine pairs (a pair of face images from the same identity), and impostor pairs (a pair of face images from different identities). A Receiving Operating Characteristic (ROC) curve is utilized to evaluate the FR performance on verification tasks, where the percentage of genuine access is reported as the True Acceptance Rate (TAR) and the percentage of impostor falsely gaining access is reported as the False Acceptance Rate (FAR) for a given match threshold,  $\tau$ . Let  $I_1^e$  and  $I_2^e$  denote a genuine pair of face images, and  $I_1^m$  and  $I_2^m$  denote an impostor pair of face images. Then TAR and FAR can be formulated as [31]:

$$TAR(\tau) = \frac{|\{\mathbf{E}|s(I_1^e, I_2^e) > \tau\}|}{|\mathbf{E}|}, \quad FAR(\tau) = \frac{|\{\mathbf{M}|s(I_1^m, I_2^m) > \tau\}|}{|\mathbf{M}|}, \quad (1.1)$$

where  $\mathbf{E}$  is the set of genuine pairs,  $\mathbf{M}$  is the set of impostor pairs, and  $s(\cdot, \cdot)$  is a given similarity function.

**Identification.** The task of identification is mostly aimed at identity search without user interaction, for example, surveillance systems. Similar to evaluation on verification systems, the identification test is conducted by dividing face images into two groups: 1) *probe* images whose identities are unknown, denoted as  $\mathbf{P}$ ; 2) *gallery* images that belong to people with known identities, denoted as  $\mathbf{G}$ . In general, each subject has one face image in the gallery. Based on the relationship between the probe and gallery identities, the evaluation is split into two different settings: 1) *closed-set* identification where all probe identities are assumed to be among the gallery identities; 2) *open-set* identification where probe identities are not necessarily in the gallery. For closed-set, a Cumulative Match Characteristic (CMC) curve is used to report the correct identification rate for each cumulative rank (the number of candidates returned). By ordering the similarity scores between a probe image  $I^p \in \mathbf{P}$  and images in the gallery  $I^g \in \mathbf{G}$ , the rank of  $I^p$  is computed as the number of identity in the gallery whose similarity scores are higher or equal to the correct identity

$I_p^g$ :

$$\text{rank}(I^p) = |\{\mathbf{G} | s(I^g, I^p) \geq s(I_p^g, I^p); I^g \in \mathbf{G}\}| \quad (1.2)$$

For a given rank  $r$ , the correct identification rate in a CMC curve is  $\frac{|\{\mathbf{P} | \text{rank}(I^p) \leq r\}|}{|\mathbf{P}|}$ . In the case of open-set, images in the probe set  $\mathbf{P}$  can be classified into known subjects  $\mathbf{S}$  (the subjects that appear in the gallery) and unknown subjects  $\mathbf{U}$  (the subjects that are not in the gallery), i.e.,  $\mathbf{P} = \mathbf{S} \cup \mathbf{U}$ . A Detection and Identification Rate (DIR) curve is plotted to show the correct identification rates with respect to the false acceptance rates (FAR). Here, the definition of FAR is different from that in an ROC curve. For a given threshold  $\theta$ , a false acceptance occurs when the similarity of an unknown probe  $I^p \in \mathbf{U}$  to one of the gallery subjects is higher than  $\theta$ . FAR computes the average probability as [32]:

$$\text{FAR}(\theta) = \frac{|\{\mathbf{P} | \max_{\mathbf{G}} s(I^g, I^p) \geq \theta; I^p \in \mathbf{U}\}|}{|\mathbf{U}|}. \quad (1.3)$$

The DIR for a given rank  $r$  is calculated on the known probe set  $\mathbf{S}$  as [32]:

$$\text{DIR}(\theta, r) = \frac{|\{\mathbf{P} | \text{rank}(I^p) \leq r \wedge s(I^g, I^p) \geq \theta; I^p \in \mathbf{S}\}|}{|\mathbf{S}|}. \quad (1.4)$$

### 1.1.3 Face Recognition Pipeline

A general FR system consists of four steps, *i.e.*, face detection, alignment and normalization, feature extraction, and feature matching. Fig. 1.2 shows a typical pipeline of feature-based FR systems. First, the face area is localized in an input image, and the face region is cropped from the original image. Next, the cropped face image is aligned (typical alignments include translation, rotation, and scaling) based on the detected facial landmarks (key points on face including eyebrows, eyes, nose, mouth, and jaw silhouette). To reduce the effects of illumination variations, the aligned face images need to be normalized before being used for feature extraction. Then, in the representation stage, we extract a compact set of discriminating geometrical and/or photometrical features of the face so that each face image is represented and stored as a  $d$ -dimensional feature vector. Finally, using a distance metric, the similarity between two feature vectors is measured, also known as the



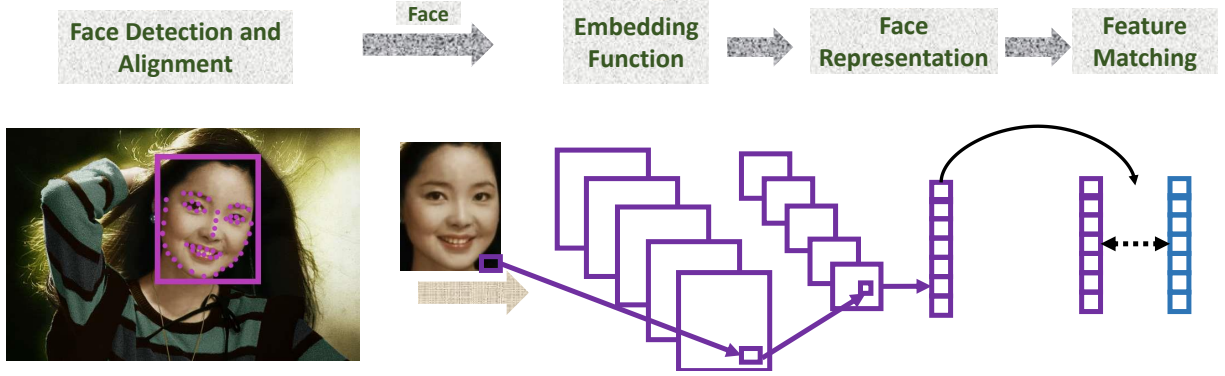


Figure 1.2 A typical pipeline of feature-based FR frameworks comprises of face detection, alignment, normalization, feature extraction, and feature matching. While each of these components affect the performance of FR systems, in this thesis, we focus on the representation function of feature extraction that maps a high-dimensional normalized face image to a  $d$ -dimensional vector representation.

similarity score, to verify if two face images belong to the same person (face verification), or to identify the identity of the face image by assigning it to the label of the nearest gallery image (rank-1 face identification). Since our central focus is on face representations, we refer to the steps before feature extraction as data pre-processing. In this thesis, we mainly employ two facial landmark detection algorithms, MTCNN [33] and Retinaface [34], to detect and align all faces in our training and testing datasets. In particular, each face is cropped from the detected face region and resized to the same size using a similarity transformation based on the detected five facial landmarks, *i.e.*, left eye center, right eye center, nose tip, left mouth corner, and right mouth corner.

## 1.2 Face Representation Extractors

The subject of FR is as old as the field of computer vision [35]. Unsurprisingly, face recognition has received tremendous attention in the computer vision and biometrics communities over the past three decades. Here, we present a few notable approaches to learning face representation (hand-crafted and learning-based methods).

**Hand-crafted Representation.** The first and most well-known global feature extraction method is Eigenfaces [36], in which principal component analysis (PCA) is employed on normalized vectors of training images to find the principal eigenvectors, corresponding to the largest, say  $k$ , eigenvalues.

The obtained eigenvectors are then used as a seed set to represent other face images (not in the training set) via a linear projection operation. To utilize the information when each subject has more than one image in the training set and the images are labeled by subject ID, Fisherfaces [37] uses Fisher’s Linear Discriminant Analysis (LDA) to minimize intra-class variations (among images of the same person) while maximizing inter-class differences (among images belonging to different people). Besides global features, a variety of face representation methods were proposed to encode local facial components such as eyes, nose, and mouth. For example, complex coefficients of Gabor filters [38] (also known as Gabor wavelets) are used to encode both facial shape and local appearance features. In the work of [39], face images are represented by utilizing the bunch graph to collect information from Gabor wavelets convolution values at each facial landmark location. In contrast to Gabor filters, intensity based local elementary descriptors have also been used to represent face images due to their efficient computations, and insensitivity to partial occlusion and pose changes, which includes Local Binary Patterns (LBP) [40], Local Phase Quantization (LPQ) [41], Histogram of Oriented Gradients (HOG) [42], and Scale Invariant Feature Transform (SIFT) [43].

**Learning-based Representation.** Given the success of deep neural network (DNN) in the ImageNet competition [44], DNN-based representation learning have contributed to massive strides in FR capabilities [9, 45]. The defining characteristic of such methods is the use of convolutional neural network (CNN) based feature extractor, a learnable embedding function comprised of several sequential linear and non-linear operators [44]. Among the first attempts at learning face representation using deep learning, Taigman *et al.* [17] presented DeepFace which uses a deep CNN trained to classify faces in CASIA dataset, demonstrating remarkable performance on LFW dataset. As an improved version of DeepID [46], DeepID2 [18] employs both verification and identification tasks as supervision signals to learn robust discriminating representations. The following approaches have explored different loss functions to improve the discriminability of the embedding vector. Researchers from Google [12] use a massive dataset of about 200 million face images of 8 million identities to train a CNN directly for face verification (called FaceNet). They optimize a triplet loss function which is based on triplets of images comprising a pair of similar and a pair of dissimilar

faces. The loss function is formulated as:

$$\mathcal{L}_{triplet} = \sum_{i=1}^M [\alpha + d(\mathbf{r}^a, \mathbf{r}^+) - d(\mathbf{r}^a, \mathbf{r}^-)]_+, \quad (1.5)$$

where  $M$  is the number of triplets,  $\mathbf{r}^a$ ,  $\mathbf{r}^+$  and  $\mathbf{r}^-$  are the representations of an anchor face image, a positive (same identity as the anchor) and a negative (different identity from the anchor) face image, respectively.  $[x]_+ = \max\{0, x\}$ ,  $\alpha$  is a margin parameter and  $d(\cdot)$  is the squared euclidean distance. Inspired by DNN-based image classification, a major part of the efforts has been made to develop new loss functions on top of the softmax layer based on cross-entropy loss:

$$\mathcal{L}_{cross-entropy}(\mathbf{r}, y; \mathbf{W}, \mathbf{b}) = - \sum_{k=1}^K \mathbb{I}(k = y) \log \frac{e^{\mathbf{W}_y^T \mathbf{r} + \mathbf{b}_y}}{\sum_{j=1}^K e^{\mathbf{W}_j^T \mathbf{r} + \mathbf{b}_j}}, \quad (1.6)$$

where  $\mathbf{r}$  and  $y$  are the representation and the identity label of an input face image,  $\mathbf{W}$  and  $\mathbf{b}$  are the parameters of the softmax layer, and  $K$  is the number of classes (unique identities) in the training set. Wen *et al.* [47] propose center loss that is augmented with Eq. (1.6) to reduce the intra-class variations. In the L2-constrained softmax [48], a feature vector  $\mathbf{r}$  is first normalized by its  $l_2$  norm to lie on a hyper-sphere and then scaled by a constant factor. SphereFace [13] introduces angular softmax (A-softmax) in which the original softmax is modified to directly optimize angles between  $\mathbf{W}_y$  and  $\mathbf{r}$ , resulting in angularly distributed features. Other softmax modifications enforce extra intra-class concentration and inter-class variance to face features by adding a margin penalty to the decision boundary [6, 19, 49]. In CosFace [19], both the representation  $\mathbf{r}$  and the weight vectors  $\mathbf{W}$  are  $l_2$  normalized to compute their cosine similarity, based on which a cosine margin term is introduced to further broaden the decision boundary in an angular space. ArcFace [6] adds an additive angular margin to the angle between the representation  $\mathbf{r}$  and its target weight vector  $\mathbf{W}_y$  via the arc-cosine function. All these face representation methods share the same objective: increasing inter-class distances and reducing intra-class variations.

## 1.3 Manifold of Face Representation

A face representation is obtained by an embedding function that transforms the raw pixel representation of the image to a point in a high-dimensional feature space. Learning or estimating such a mapping is motivated by two goals: (a) compactness of the representation, and (b) effectiveness of the mapping for FR. Given the methods introduced in Sec. 1.2, the latter topic has received substantial attention. Yet, there has been little focus on the dimensionality of the representation itself. Another topic related to representation compactness is the capacity of face representation. *Given a face representation, how many identities can it resolve?* In this work, we develop algorithms to estimate the intrinsic dimensionality and capacity of face representations, and design a new dimensionality reduction method to obtain compact representations.

### 1.3.1 Intrinsic Dimensionality

The dimensionality of face representations extracted from deep networks has ranged from hundreds to thousands of dimensions. For instance, current SOTA face representations have 128, 512, 1,024 and 4,096 dimensions for FaceNet [12], ResNet [50], SphereFace [13], and VGG [51], respectively. The choice of dimensionality is often determined by practical considerations, such as, ease of learning the embedding function [52], constraints on system memory, etc., instead of assigning effective dimensionality needed for image representation. This naturally raises the following fundamental but related question, *How compact can the representation be without any loss in recognition performance?* In other words, *what is the intrinsic dimensionality of the representation?* Subsequently, *how can one obtain such a compact representation?*

The intrinsic dimensionality (IND) of a representation refers to the minimum number of parameters (or degrees of freedom) necessary to capture the information present in the representation [53]. Equivalently, it refers to the dimensionality of the  $m$ -dimensional manifold,  $\mathcal{M}$ , embedded within the  $d$ -dimensional ambient (representation) space  $\mathcal{P}$  where  $m \leq d$ . This notion of intrinsic dimensionality is notably different from common linear dimensionality estimates obtained through

*e.g.*, PCA. This linear dimension corresponds to the best linear subspace necessary to retain a desired fraction of the variations in the data. In principle, linear dimensionality can be as large as the ambient dimension if the variation factors are highly entangled with each other.

The ability to estimate the intrinsic dimensionality of a given face representation is useful in a number of ways. At a *fundamental* level, the IND determines the true capacity and complexity of variations in the data captured by the representation, through the embedding function. In fact, IND can be used to gauge the information content in the representation, due to its linear relation with Shannon entropy [54, 55]. Also, it provides an estimate of the amount of redundancy built into the representation which relates to its generalization capability. On a *practical* level, knowledge of the IND is crucial for devising optimal unsupervised strategies to obtain face features that are minimally redundant, while retaining its full ability to recognize faces of different identities. Recognition in the intrinsic space can provide significant savings, both in memory requirements for the templates as well as processing time, across downstream tasks like large-scale face matching in the encrypted domain [56]. Lastly, the gap between ambient and intrinsic dimensionalities of a representation can serve as a useful indicator to drive the development of algorithms that can directly learn highly compact embeddings.

Estimating the IND of a given face representation is, however, a challenging task. Such estimates are crucially dependent on the density variations in the representation, which in itself is difficult to estimate as images often lie on a topologically complex curved manifold [57]. More importantly, given an estimate of IND, how do we verify that it truly represents the dimensionality of the complex high-dimensional representation space? An indirect validation of the IND is possible through a mapping that transforms the ambient representation space to the intrinsic representation space while preserving its discriminative ability. However, there is no certainty that such a mapping can be found efficiently. In practice, finding such mappings can be considerably harder than estimating the IND itself.

We overcome both of these challenges by (1) adopting a topological dimensionality estimation technique based on the geodesic distance between points on the manifold, and (2) relying on the

ability of DNNs to approximate the complex mapping function from the ambient space to the intrinsic space (as we see below in Ch. 2). The latter enables validation of the IND estimates through face matching experiments on the corresponding low-dimensional intrinsic representation of feature vectors.

### 1.3.2 Capacity

Consider the following scenario: we would like to deploy a FR system with representation  $M$  in a target application that requires a maximum FAR of  $q\%$ . As subjects are continuously augmented to the gallery, an intuitively known and empirically observed phenomenon occurs: the FR accuracy starts decreasing. This is primarily due to the fact that with more subjects and diverse viewpoints, the representations of the classes will no longer be disjoint. In other words, the FR system based on representation  $M$  can no longer completely resolve all of the users within the  $q\%$  FAR. We define the maximal number of users at which the face representation reaches this limit as the capacity <sup>1</sup> of the representation. Our main contribution in this work, is to determine the capacity in an objective manner without the need for empirical evaluation.

The ability to determine this capacity affords the following benefits: (i) Statistical estimates of the upper bound on the number of identities the face representation can resolve. This would allow for informed deployment of FR systems based on the expected scale of operation; (ii) Estimate the maximal gallery size for the face representation *without* having to exhaustively evaluate the face representation at each scale. Consequently, capacity offers an alternative dataset <sup>2</sup>-agnostic metric for comparing different face representations.

An attractive solution for estimating the capacity of face representations is to leverage the notion of packing bounds <sup>3</sup>; the maximal number of shapes that can be fit, without overlapping, within the support of the representation space. A loose bound on this packing problem can be obtained

---

<sup>1</sup>This is different from the notion of capacity of a space of functions as measured by its Vapnik–Chervonenkis dimension of linear classifiers.

<sup>2</sup>Class of datasets as opposed to a specific dataset.

<sup>3</sup>A generalization of the well studied sphere-packing problem.

as a ratio of the volume of the support space and the volume of the shape. In the context of face representations, the representation support can be modeled as a low-dimensional population manifold  $\mathcal{M} \in \mathbb{R}^m$  embedded within a high-dimensional representation space  $\mathcal{P} \in \mathbb{R}^p$ , while each class <sup>4</sup> can be modeled as its own manifold  $\mathcal{M}_c \subseteq \mathcal{M}$ . Under this setting, a bound on the capacity of the representation can be obtained as a ratio of the volumes of the population and class-specific manifolds. However, adopting this approach to obtain empirical estimates of the capacity presents the following challenges:

1. Estimating the support of the population manifold  $\mathcal{M}$  and the class-specific manifolds  $\mathcal{M}_c$ , especially for a high-dimensional embedding, such as a face representation (typically, several hundred), is an open problem.
2. Estimating the density of the manifolds while accounting for the different sources of noise is a challenging task. In the context of face representations, all the components of a typical face representation pipeline (see Fig. 1.2) are potential sources of noise.
3. Obtaining reliable estimates of the volume of arbitrarily shaped high-dimensional manifolds (for capacity bound), is another open problem.

We propose a framework that addresses the aforementioned challenges to obtain reliable estimates of the capacity of any face representation. Our solution relies on: (1) modeling the face representation as a low-dimensional Euclidean manifold embedded within a high-dimensional space, (2) projecting and unfolding the manifold to a low-dimensional space, (3) approximating the population manifold by a multivariate Gaussian distribution (equivalently, hyper-ellipsoidal support) in the unfolded low-dimensional space, (4) approximating the class-specific manifolds by a multi-variate Gaussian distribution and estimating its support as a function of the specific FAR, and (5) estimating the capacity as a ratio of the volumes of the population and class-specific hyper-ellipsoids. This work is introduced below in Ch. 3.

---

<sup>4</sup>In the case of FR, each class is an identity (subject) and the number of classes corresponds to the number of identities.

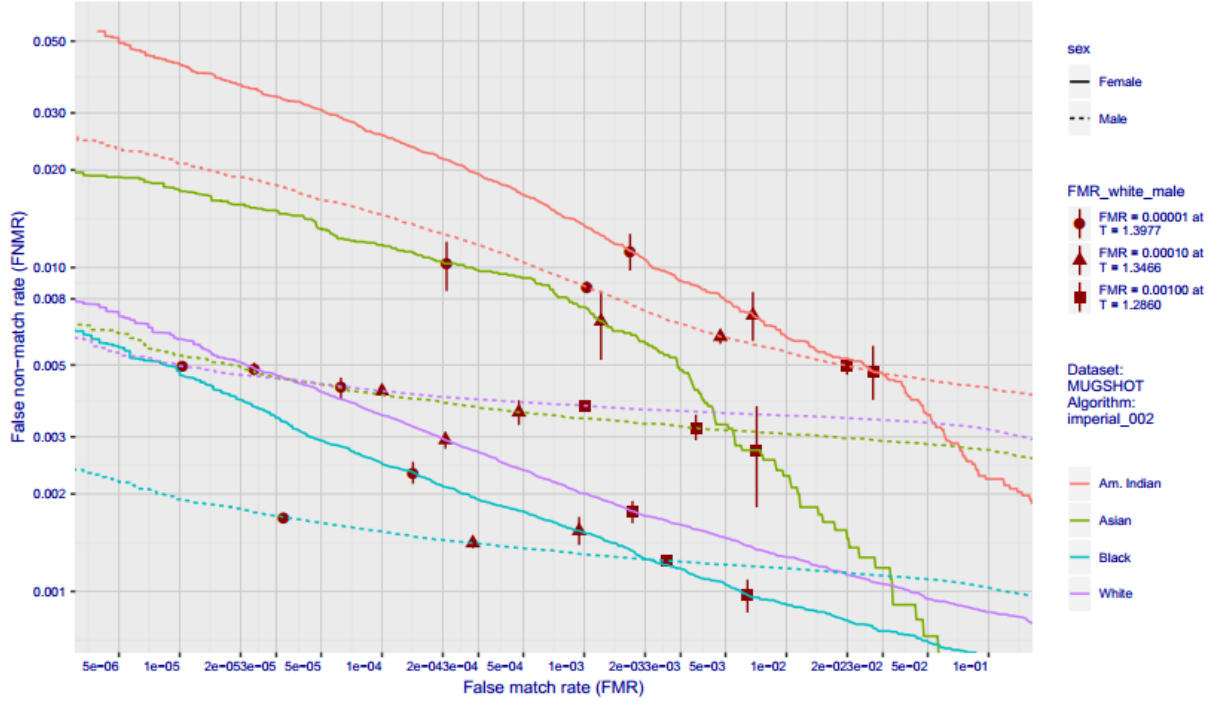


Figure 1.3 The error tradeoff characteristics in the form of false non-match rates ( $\text{FNMR} = 1 - \text{TAR}$ ) vs. false match rates ( $\text{FMR} = \text{FAR}$ ) for one commercial FR algorithm verifying mugshot images, reported by the NIST (National Institute of Standards and Technology) Face recognition Vendor Test [7]. The FMR estimates are computed on impostor pairs of face images with same gender and same race. Each symbol (circle, triangle, square) corresponds to a fixed threshold - their vertical and horizontal displacements reveal, respectively, differences in FNMR and FMR between demographic groups [7].

## 1.4 Bias in Face Recognition

FR systems are known to exhibit discriminatory behaviors against certain demographic groups [7, 14, 21]. Fig. 1.3 shows one commercial FR algorithm that has lower performance in certain demographic groups than others in the 2019 NIST Face Recognition Vendor Test (FRVT) [7]. In fact, all 106 FR algorithms that participated in the NIST FRVT exhibit different levels of biased performances on gender, race, and age groups of a mugshot dataset (see Fig. 1.4). For all the algorithms listed in Fig. 1.4, we see the algorithms that achieve better performance present less sex/race bias. For example, the difference in FMR between the highest and lowest sex/race group is less than 0.1% for the best model shown in the last row of Fig. 1.4. Even so, demographic bias still exists in current FR algorithms. In a time when FR systems are being deployed in the real world for societal benefit,



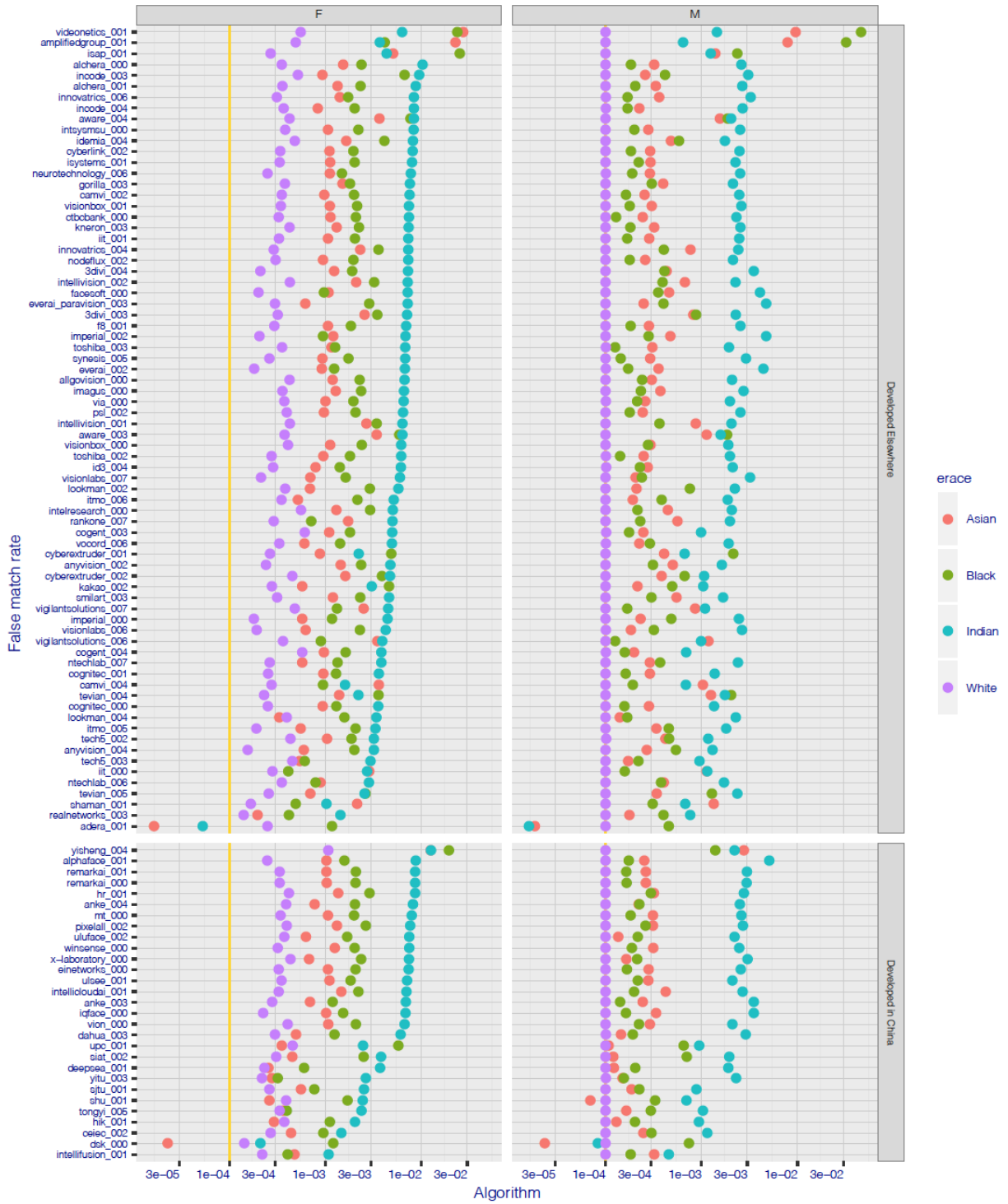


Figure 1.4 False positive differentials in verification algorithms provided to NIST. The dots give the false match rates for same-sex and same-race impostor comparisons. The threshold is set for each algorithm to give FMR = 0.0001 on white males (the purple dots in the right hand panel). The algorithms are sorted in order of worst case FMR [7].

this type of bias <sup>5</sup> is not acceptable. Note that here, we define FR bias as the uneven recognition performance with respect to demographic groups. It is desirable to design unbiased FR algorithms to maintain fairness in FR performance when deploying this technology for law enforcement and other applications.

A natural question arises, *Why does the bias problem exist in FR systems?* First, SOTA FR algorithms [6, 13, 19] rely on CNNs trained on large-scale face datasets. The public training datasets for FR, e.g., CASIA WebFace [11], VGGFace2 [59], and MS-Celeb-1M [23], are collected by scraping face images off the web, with inevitable demographic bias [5]. Previous studies have shown that models trained with imbalanced datasets (unequal number of training samples from different demographic groups) lead to biased discrimination [60, 61]. Similarly, bias in face datasets is transmitted to the FR models through network learning. For example, to minimize the overall loss, a network tends to learn a better representation for faces in the majority group whose number of faces dominate the training set, resulting in unequal discriminabilities. The imbalanced distribution of demographics in face data, is nevertheless, not the only trigger of FR bias. Prior work has shown that even using a demographic balanced dataset [5] or training separate classifiers for each group [14], the performance on some groups is still inferior to the others. This shows the bias in embedding functions. Since the goal of face representation is to map the input face image to a target feature vector with high discriminative power, the bias in the mapping function will result in feature vectors with lower discriminability for certain demographic groups. Furthermore, by studying non-trainable FR algorithms, [14] introduced a new notion of inherent bias, i.e., certain groups are inherently more susceptible to errors in the face matching process.

To tackle the dataset-induced bias, data re-sampling methods have been exploited to balance the data distribution by under-sampling the classes with more samples [62] or over-sampling the classes with less samples [63, 64]. Despite its simplicity, valuable information may be removed by under-sampling, and over-sampling may introduce noisy samples. Naively training on a balanced dataset can still lead to bias [5]. Another common option for imbalanced data training is cost-

---

<sup>5</sup>This is different from the notion of inductive bias in machine learning, defined as "any basis for choosing one generalization [hypothesis] over another, other than strict consistency with the observed training instances" [58].

sensitive learning that assigns weights to different classes based on (i) their frequency or (ii) the effective number of samples [65, 66]. Recent imbalanced learning methods focus on novel objective functions for class-skewed datasets. For instance, Dong *et al.* [67] propose a Class Rectification Loss to incrementally optimize on hard samples of the classes with under-represented attributes. Alternatively, researchers strengthen the decision boundary to impede perturbation from other classes by enforcing margins between hard clusters via adaptive clustering [68], or between rare classes via Bayesian uncertainty estimates [69]. To adapt the aforementioned methods to racial bias mitigation, Wang *et al.* [5] modify the large margin based loss functions by reinforcement learning. However, [5] requires two auxiliary networks, an offline sampling network and a deep Q-learning network, to generate adaptive margin policy for training the FR network, which hinders the learning efficiency.

We propose two different approaches to fair representation learning for FR systems. The first approach, called *DebFace*, utilizes adversarial learning to disentangle a face representation into four components, *i.e.*, gender, age, race/ethnicity, and identity. Each of the four components is independent from others. We de-bias a face representation under the assumption that if no discriminating demographic information is captured by the face representation, it would be unbiased with respect to demographic attributes. More details are given in Ch. 4. The second approach, called *GAC (Group Adaptive Classifier)*, addresses the bias issue in a different way. GAC optimizes the face representation learning on every demographic group in a single network via adaptive convolution kernels and channel-wise attention maps, which increases the network capacity for representing multiple face patterns from different demographic groups.

## 1.5 Face Representation via Graph Neural Network

As introduced in Sec. 1.2, face representation learning is the process of establishing an embedding function by which a face image is transformed to a high-dimensional feature vector. Among the current SOTA DNN-based approaches, different network architectures and loss functions have

been explored to improve FR performance. In early studies [17, 46], deep features are learnt via a face classification objective. However, later studies [18, 48] found a simple classification loss is insufficient to capture discriminative face features, and thus attempted to design appropriate loss functions to enhance the discriminative power of representations. One direction of research is to directly learn an embedding via metric learning, *e.g.*, contrastive loss [18] and triplet loss [12]. The other line of methods adapt the traditional softmax-cross-entropy loss to decrease the intra-class variations [47], or to lengthen the inter-class distances by adding an extra margin between classes in either cosine [19] or angular space [6, 13].

While the FR performance is improved, both types of methods have limitations. For the softmax approach and its variants, the dimensionality of the output softmax vector increases linearly with the number of identities in the training set, which may lead to a bottleneck issue in the computation. Distance metric learning approaches can avoid this issue by acquiring a feature space where distance corresponds to class similarity. However, a carefully designed scheme is required to select the pair or triplet samples from a tremendous number of combinations for large-scale datasets. Furthermore, both classification-based and distance-based objectives tackle merely on each individual sample or at most a triplet of samples in the physical distance metric, but ignore the *general distribution* derived from correlations between samples of within-class and cross-class.

To address the aforementioned shortcomings of the FR loss functions, we propose a representation learning method that utilizes graph classification via adversarial training. In contrast to using a metric as the constraint in the feature space, our idea is to create an ideal feature space, referred to as *oracle space*, where the cluster of feature points in each class is clearly separated from other classes. A deep neural network (DNN) is then trained to generate face features that follow the data distribution in the *oracle space*. In this case, the face representation model can be regarded as the generative model in Generative Adversarial Network (GAN), which has emerged as a useful framework for learning arbitrary distributions from observed data samples. By means of adversarial learning, GAN offers a pleasing option for generative tasks in which the generator is trained to derive the data distribution from observed samples. However, the relationships and inter-dependencies

between feature points are not as simple as the data structure of fixed-size grid images. For this reason, we can no longer use the conventional *discriminator* in GAN as the form of adversarial supervision.

Instead, we consider the data structure in feature space as a directed graph, where each vertex corresponds to an image sample and the edges between vertices represent their dependencies. For the oracle space, features points are connected if they belong to the subject; while in the actual feature space, nodes are linked to their  $k$  nearest neighbors. The discrimination task here is to distinguish between graphs from the oracle space and the generated space. To this end, we employ a graph classifier trained with Graph Neural Network (GNN), as the discriminator that guides the representation model to output features that follow the oracle distribution. The proposed framework is thus capable of learning a generic feature space with enhanced discriminative power for face images, based on a predesigned feature distribution defined on a graph structure. Furthermore, the construction of oracle graphs provides us with an opportunity to take control of the learning complexity so as to flexibly adjust the bias-variance trade-off from the perspective of the training objective.

## 1.6 Thesis Contributions

The research work carried out in addressing the above problems has resulted in a number of contributions to face recognition. The key contributions of this dissertation are briefly described below:

- It is the first attempt to estimate the intrinsic dimensionality of DNN based image representations. Numerical experiments yield an ID estimate of, 12 and 16 for FaceNet [12] and SphereFace [13] face representations, respectively, and 19 for ResNet-34 [70] image representation. The estimates are significantly lower than their respective ambient dimensionalities, 128-*dim* for FaceNet and 512-*dim* for the others. We also propose an unsupervised DNN based dimensionality reduction method under the framework of multi-dimensional

scaling, called DeepMDS. DeepMDS mapping is significantly better than other dimensionality reduction approaches in terms of its discriminative capability.

- It is the first practical attempt at estimating the capacity of DNN based face representations. We consider two such representations, namely, FaceNet [12] and SphereFace [13] consisting of 128-dimensional and 512-dimensional feature vectors, respectively. We propose a noise model for facial embeddings that explicitly accounts for two sources of uncertainty, uncertainty due to data and the uncertainty in the parameters of the representation function. We can estimate capacity as a function of the desired operating point, in terms of the maximum desired probability of false acceptance error by establishing a relationship between the support of the class-specific manifolds and the discriminant function of a nearest neighbor classifier.
- We provide a thorough analysis of deep learning based face recognition performance on three different demographics: (i) gender, (ii) age, and (iii) race. We propose two face recognition frameworks that mitigate demographic bias: (i) DebFace, and (ii) GAC. DebFace generates disentangled representations for both identity and demographic attribute recognition while jointly removing discriminative information from other counterparts. The result indicate both the identity representation and the demographic attribute estimation via DebFace show lower bias on different demographic cohorts. GAC reduces demographic bias and increases robustness of representations for faces in every demographic group by adopting adaptive convolutions and attention techniques. GAC is able to automatically determine the layers to employ dynamic kernels and attention maps, leading to SOTA performance on a demographic-balanced dataset and three benchmark datasets.
- We propose a new framework for face representation learning, which models the data distribution in an oracle space via adversarial learning, to transform raw pixels of an image to a highly discriminative feature vector for face recognition. Graphs are constructed in the feature space to describe the data distribution of feature points with respect to their identities and similarities. We also provide a thorough analysis towards the impact of predefined graphs on the discriminability of the learned face representations. Our graph based approach surpasses

the baseline model and achieves state-of-the-art performance on six benchmark datasets (LFW [27], CPLFW [71], CFP-FP [72], IJB-A [29], IJB-B [30], IJB-C [9]).

## 1.7 Thesis Structure

Ch. 2 of this thesis focuses on the compactness of a face representation, and proposes a new algorithm to reduce the dimensionality of the representation with little degradation in performance. With the dimensionality reduction tool developed in Ch. 2, Ch. 3 estimates the capacity of a face representation on a more compact feature space, attempting to overcome the curse of dimensionality that may lead to over-estimated capacity values. In Ch. 4, the issue of demographic bias in FR systems is addressed. Besides an empirical analysis of the unequal verification performance in different demographic groups, we introduce new strategies to mitigate such bias for fairer face representation learning. A new framework for face representation learning is presented in Ch. 5, which utilizes adversarial learning to acquire oracle feature distribution in the form of a  $k$ -NN graph. The last chapter discusses the conclusions of this dissertation and presents directions for future work. The experimental results of the work in this thesis were previously presented in [8, 73–75].

## Chapter 2

# The Intrinsic Dimensionality of Face Representation

This chapter addresses the following questions pertaining to the intrinsic dimensionality of any given face representation: (i) estimate its intrinsic dimensionality, (ii) develop a deep neural network based non-linear mapping, dubbed DeepMDS, that transforms the ambient representation to the minimal intrinsic space, and (iii) validate the veracity of the mapping through face matching in the intrinsic space. Experiments on benchmark image datasets (LFW [27] and IJB-C [9]) reveal that the intrinsic dimensionality of deep neural network representations is significantly lower than the dimensionality of the ambient features. For instance, SphereFace’s [13] 512–dim face representation has an intrinsic dimensionality of 16 on IJB-C dataset. Further, the DeepMDS mapping is able to obtain a representation of significantly lower dimensionality while maintaining discriminative ability to a large extent, 59.75% TAR @ 0.1% FAR in 16–dim *vs* 71.26% TAR in 512– dim on IJB-C.

The key contributions and findings of this chapter are:

- The first attempt to estimate the intrinsic dimensionality of DNN based face representations.
- An unsupervised DNN based dimensionality reduction method under the framework of multidimensional scaling, called DeepMDS.
- Numerical experiments yield an IND estimate of, 12 and 16 for FaceNet [12] and SphereFace [13]



face representations, respectively. The estimates are significantly lower than their respective ambient dimensionalities, 128-*dim* for FaceNet and 512-*dim* for SphereFace.

– DeepMDS mapping is significantly better than other dimensionality reduction approaches (e.g., PCA and Isomap [76]) in terms of its discriminative capability.

## 2.1 Intrinsic Dimensionality

Existing approaches for estimating intrinsic dimensionality can be broadly classified into two groups: projection methods and geometric methods. The projection methods [77–79] determine the dimensionality by principal component analysis on local subregions of the data and estimating the number of dominant eigenvalues. These approaches have classically been used in the context of modeling facial appearance under different illumination conditions [80] and object recognition with varying pose [81]. While they serve as an efficient heuristic, they do not provide reliable estimates of intrinsic dimension. Geometric methods [2, 82–86], on the other hand, model the intrinsic topological geometry of the data and are based on the assumption that the volume of a  $m$ -dimensional set scales with its size  $\epsilon$  as  $\epsilon^m$  and hence the number of neighbors less than  $\epsilon$  also behaves the same way.

Our approach in this chapter is based on the topological notion of correlation dimension [82, 83], the most popular type of fractal dimensions. The correlation dimension implicitly uses nearest-neighbor distance, typically based on the Euclidean distance. However, Granata *et al.* [1] observe that leveraging the manifold structure of the data, in the form of geodesic distances induced by a neighborhood graph of the data, provides more realistic estimates of the IND. Building upon this observation we base our IND estimates on the geodesic distance between points. We believe that estimating the intrinsic dimensionality would serve as the first step towards understanding the bound on the minimal required dimensionality for representing faces and aid in the development of novel algorithms that can achieve this limit.

## 2.2 Dimensionality Reduction

There is a tremendous body of work on the topic of estimating low-dimensional approximations of data manifolds lying in high-dimensional space. These include linear approaches such as Principal Component Analysis [87], Multidimensional Scaling (MDS) [88] and Laplacian Eigenmaps [89] and their corresponding non-linear spectral extensions, Locally Linear Embedding [90], Isomap [76] and Diffusion Maps [91]. Another class of dimensionality reduction algorithms leverage the ability of deep neural networks to learn complex non-linear mappings of data, including deep autoencoders [92], denoising autoencoders [93, 94] and learning invariant mappings either with the contrastive loss [95] or with the triplet loss [12]. While the autoencoders can learn a compact representation of data, such a representation is not explicitly designed to retain discriminative ability. Both the contrastive loss and the triplet loss have a number of limitations; (1) requires similarity and dissimilarity labels and cannot be trained in an unsupervised setting, (2) requires an additional hyper-parameter, maximum margin of separation, which is difficult to pre-determine, especially for an arbitrary representation, and (3) does not maintain the manifold structure in the low-dimensional space. In this work, we too leverage DNNs to approximate the non-linear mapping from the ambient to the intrinsic space. However, we consider an unsupervised setting (i.e., no similarity or dissimilarity labels) and cast the learning problem within the framework of MDS i.e., preserving the ambient graph induced geodesic distance between points in the intrinsic space.

## 2.3 Our Approach

Our goal in this work is to compress a given face representation space. We achieve this in two stages <sup>1</sup>: (1) estimate the intrinsic dimensionality of the ambient face representation, and (2) learn the DeepMDS model to map the ambient representation space  $\mathcal{P} \in \mathbb{R}^d$  to the intrinsic representation space  $\mathcal{M} \in \mathbb{R}^m$  ( $m \leq d$ ). The IND estimates are based on the one presented by [1] which relies on

---

<sup>1</sup>Traditional single-stage dimensionality reduction methods use visual aids to arrive at the final IND and intrinsic space, e.g., plotting the projection error against the IND values and looking for a “knee” in the curve.

two key ideas: (1) using graph induced geodesic distances to estimate the correlation dimension of the face representation topology, and (2) the similarity of the distributions of geodesic distances across different topological structures with the same intrinsic dimensionality. The DeepMDS model is optimized to preserve the interpoint geodesic distances between the feature vectors in the ambient and intrinsic space, and is trained in a stage-wise manner that progressively reduces the dimensionality of the representation. Basing the projection method on DNNs, instead of spectral approaches like Isomap, addresses the scalability and out-of-sample-extension problems suffered by spectral methods. Specifically, DeepMDS is trained in a stochastic fashion, which allows it to scale. Furthermore, once trained, DeepMDS provides a mapping function in the form of a feed-forward network that maps the ambient feature vector to its corresponding intrinsic feature vector. Such a map can easily be applied to new test data.

### 2.3.1 Estimating Intrinsic Dimensionality

We define the notion of intrinsic dimension through the classical concept of *topological dimension* of the support of a distribution. This is a generalization of the concept of dimension of a linear space <sup>2</sup> to a non-linear manifold. Methods for estimating the topological dimension are all based on the assumption that the behavior of the number of neighbors of a given point on an  $m$ -dimensional manifold embedded within a  $d$ -dimensional space scales with its size  $\epsilon$  as  $\epsilon^m$ . In other words, the density of points within an  $\epsilon$ -ball ( $\epsilon \rightarrow 0$ ) in the ambient space is independent of the ambient dimension  $d$  and varies only according to its intrinsic dimensionality  $m$ . Given a collection of points  $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$ , where  $\mathbf{x}_i \in \mathbb{R}^d$ , the cumulative distribution of the pairwise distances  $C(r)$  between the  $n$  points can be estimated as,

$$C(r) = \frac{2}{n(n-1)} \sum_{i < j=1}^n H(r - \|\mathbf{x}_i - \mathbf{x}_j\|) = \int_0^r p(r) dr \quad (2.1)$$

---

<sup>2</sup>Linear dimension is the minimum number of independent vectors necessary to represent any given point in this space as a linear combination.

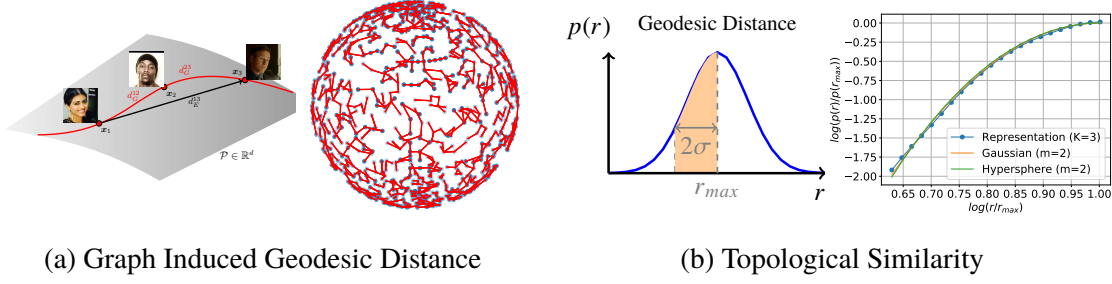


Figure 2.1 **Intrinsic Dimension:** Our approach is based on two observations: (a) Graph induced geodesic distance between images is able to capture the topology of the image representation manifold more reliably. As an illustration, we show the graph edges for the surface of a unitary hypersphere and a face manifold of ID two, embedded within a 3-*dim* space. (b) The distribution of the geodesic distances (for distance  $r_{max} - 2\sigma \leq r \leq r_{max}$ , where  $r_{max}$  is the distance at the mode) has been empirically observed [1] to be similar across different topological structures with the same intrinsic dimensionality. The plot shows the distance distribution for a face representation, unitary hypersphere and a Gaussian distribution of ID two embedded within 3-*dim* space. [8]

where  $H(\cdot)$  is the Heaviside function and  $p(r)$  is the probability distribution of the pairwise distances. In this work, we choose the correlation dimension [82], a particular type of topological dimension, to represent the intrinsic dimension of the face representation. It is defined as,

$$m = \lim_{r \rightarrow 0} \frac{\ln C(r)}{\ln r} \implies \lim_{r \rightarrow 0} C(r) \propto r^m \quad (2.2)$$

Therefore, the intrinsic dimension is crucially dependent on the accuracy with which the probability distribution can be estimated at very small length-scales (distances), i.e.,  $r \rightarrow 0$ . Significant efforts have been devoted to estimating the intrinsic dimension through line fitting in the  $\ln C(r)$  vs  $\ln r$  space around the region where  $r \rightarrow 0$  i.e.,

$$\begin{aligned} m &= \lim_{(r_2 - r_1) \rightarrow 0} \frac{\ln C(r_2) - \ln C(r_1)}{\ln r_2 - \ln r_1} \\ &= \lim_{r \rightarrow 0} \frac{d \ln C(r)}{d \ln r} = \lim_{r \rightarrow 0} \frac{p(r)}{C(r)} r = \lim_{r \rightarrow 0} m(r) \end{aligned} \quad (2.3)$$

The main drawback with this approach is the need for reliable estimates of  $p(r)$  at very small length scales, which is precisely where the estimates are most unreliable when data is limited,

especially in very high-dimensional spaces. Granata et al. [1] present an elegant solution to this problem through three observations, (i) estimates of  $m(r)$  can be stable even as  $r \rightarrow 0$  if the distance between points is computed as the graph induced shortest path between points instead of the euclidean distance, as is commonly the case, (ii) the probability distribution  $p(r)$  at intermediate length-scales around the mode of  $p(r)$  i.e.,  $(r_{max} - 2\sigma) \leq r \leq r_{max}$  can be conveniently used to obtain reliable estimates of IND, and (iii) the distributions  $p(r)$  of different topological geometries are similar to each other as long as the intrinsic dimensionality is the same, or in other words the distribution  $p(r)$  depends only on the intrinsic dimensionality and not on the geometric support of the manifolds.

Fig. 2.1 provides an illustration of these observations. Consider two different manifolds, faces and the surface of a  $(m + 1)$ -dimensional unitary hypersphere (henceforth referred to as  $m$ -hypersphere  $S^m$ ), with intrinsic dimensionality of  $m = 2$  but embedded within 3-*dim* Euclidean space. Beyond the nearest neighbor, the distance  $r$  between any pair of points in the manifold is computed as the shortest path between the points as induced by the graph connecting all the points in the representation. Fig. 2.1b shows the distribution of  $\log \frac{p(r)}{p(r_{max})}$  vs  $\log \frac{r}{r_{max}}$  in the range  $r_{max} - 2\sigma \leq r \leq r_{max}$ , where  $\sigma$  is the standard deviation of  $p(r)$  and  $r_{max} = \arg \max_r p(r)$  corresponds to the radius of the mode of  $p(r)$ . Interestingly, different topological geometries, namely, a face representation of IND two, a 2-hypersphere and a 2-*dim* Gaussian, all embedded within 3-*dim* Euclidean space have almost identical distributions. More generally, the distribution of  $\log \frac{p(r)}{p(r_{max})}$  vs  $\log \frac{r}{r_{max}}$  in the range  $r_{max} - 2\sigma \leq r \leq r_{max}$  is empirically observed to depend only on the intrinsic dimensionality, rather than the geometrical support of the manifold.

The intrinsic dimensionality of the representation manifold can thus be estimated by comparing the empirical distribution of the pairwise distances  $\hat{p}_M(r)$  on the manifold to that of a known distribution, such as the  $m$ -hypersphere or the Gaussian distribution in the range  $r_{max} - \sigma \leq r \leq r_{max}$ . We first show the derivation for estimating the intrinsic dimensionality  $m$  that minimizes the Root Mean Squared Error (RMSE) with respect to a  $m$ -hypersphere. The distribution of the geodesic distance  $p_{S^m}(r)$  of  $m$ -hypersphere can be analytically expressed as,  $p_{S^m}(r) = c \sin^{m-1}(r)$ , where  $c$

is a constant and  $m$  is the IND. Given  $\hat{p}_{\mathcal{M}}(r)$ , we minimize the RMSE between the distributions as,

$$\min_{c,m} \int_{r_{max}-2\sigma}^{r_{max}} \|\log \hat{p}_{\mathcal{M}}(r) - \log(c) - (m-1) \log(\sin[r])\|^2$$

which upon simplification yields,

$$\min_m \int_{r_{max}-2\sigma}^{r_{max}} \left\| \log \frac{\hat{p}_{\mathcal{M}}(r)}{\hat{p}_{\mathcal{M}}(r_{max})} - (m-1) \log \left( \sin \left[ \frac{\pi r}{2r_{max}} \right] \right) \right\|^2$$

The above optimization problem can be solved via a least-squares fit after estimating the standard deviation,  $\sigma$ , of  $p(r)$ . First we estimate  $\sigma$  for the  $m$ -hypersphere by approximating the distribution  $\hat{p}_{\mathcal{M}}(r)$  by a univariate Gaussian distribution around the mode of  $p_{\mathcal{M}}(r)$ . So, given samples  $S = \{r_1, \dots, r_T\}$  from the distribution  $p(r)$ , the variance around the mode can be estimated as,  $\sigma^2 = \frac{1}{T} \sum_{t=1}^T (r_t - r_{max})^2$ , where  $r_{max}$  is the radius at the mode of  $\hat{p}_{\mathcal{M}}(r)$ . Then, we estimate the distribution  $\log \frac{\hat{p}_{\mathcal{M}}(r)}{\hat{p}_{\mathcal{M}}(r_{max})}$  vs  $\log \left( \sin \left[ \frac{\pi r}{2r_{max}} \right] \right)$  and solve the following least-squares fit problem:

$$\min_m \sum_{S \cap r_{max}-2\sigma \leq r_i \leq r_{max}} (y_i - (m-1)x_i)^2$$

where  $y_i = \log \frac{\hat{p}_{\mathcal{M}}(r_i)}{\hat{p}_{\mathcal{M}}(r_{max})}$  and  $x_i = \log \left( \sin \left[ \frac{\pi r}{2r_{max}} \right] \right)$ . Such a procedure could, in principle, result in a fractional estimate of dimension. If one only requires integer solutions, the optimal value of  $m$  can be estimated by rounding-off the least squares fit solution.

In the case of comparison to a Gaussian distribution, the intrinsic dimensionality can also be estimated by comparing to the geodesic distance distribution for points sampled from a Gaussian distribution as,

$$\min_d \int_{r_{max}-2\sigma}^{r_{max}} \left\| \log \frac{p(r)}{p(r_{max})} + (d-1) \frac{r^2}{4\sigma^2} \right\|_2^2 \quad (2.4)$$

The solution of this optimization problem can be found following the same procedure described above for a  $m$ -hypersphere.

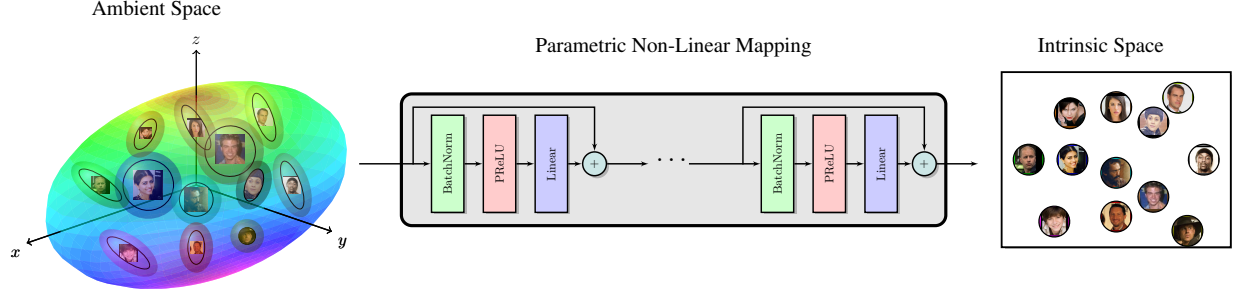


Figure 2.2 **DeepMDS Mapping:** A DNN based non-linear mapping is learned to transform the ambient space to a plausible intrinsic space. The network is optimized to preserve distances between pairs of points in the ambient and intrinsic space.

### 2.3.2 Estimating Intrinsic Sapce

The intrinsic dimensionality estimates obtained in the previous subsection alludes to the existence of a mapping, that can transform the ambient representation to the intrinsic space, but does not provide any solutions to find said mapping. The mapping itself could potentially be very complex and our goal of estimating it is practically challenging.

We base our solution to estimate a mapping from the ambient to the intrinsic space on Multidimensional scaling (MDS) [88], a classical mapping technique that attempts to preserve the distances (similarities) between points after embedding them in a low-dimensional space. Given data points  $X = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$  in the ambient space and  $Y = \{\mathbf{y}_1, \dots, \mathbf{y}_n\}$  the corresponding points in the intrinsic low-dimensional space, the MDS problem is formulated as,

$$\min \sum_{i < j} (d_H(\mathbf{x}_i, \mathbf{x}_j) - d_L(\mathbf{y}_i, \mathbf{y}_j))^2 \quad (2.5)$$

where  $d_H(\cdot)$  and  $d_L(\cdot)$  are distance (similarity) metrics in the ambient and intrinsic space, respectively. Different choices of the metric, leads to different dimensionality reduction algorithms. For instance, classical metric MDS is based on Euclidean distance between the points while using the geodesic distance induced by a neighborhood graph leads to Isomap [76]. Similarly, many different distance metrics have been proposed corresponding to non-linear mappings between the ambient space and the intrinsic space. A majority of these approaches are based on spectral decompositions and suffer

many drawbacks, (i) computational complexity scales as  $O(n^3)$  for  $n$  data points, (ii) ambiguity in the choice of the correct non-linear function, and (iii) collapsed embeddings on more complex data [95].

To overcome these limitations, we employ a DNN to approximate the non-linear mapping that transforms the ambient representation,  $\mathbf{x}$ , to the intrinsic space,  $\mathbf{y}$  by a parametric function  $\mathbf{y} = f(\mathbf{x}; \boldsymbol{\theta})$  with parameters  $\boldsymbol{\theta}$ . We learn the parameters of the mapping within the MDS framework,

$$\min_{\boldsymbol{\theta}} \sum_{i=1}^n \sum_{j=1}^n [d_H(\mathbf{x}_i, \mathbf{x}_j) - d_L(f(\mathbf{x}_i; \boldsymbol{\theta}), f(\mathbf{x}_j; \boldsymbol{\theta}))]^2 + \lambda \|\boldsymbol{\theta}\|_2^2$$

where the second term is a regularizer with a hyperparameter  $\lambda$ . Fig. 2.2 shows an illustration of the DNN based mapping.

In practice, directly learning the mapping from the ambient to the intrinsic space is very challenging, especially for disentangling a complex manifold under high levels of compression. We adopt a curriculum learning [96] approach to overcome this challenge and progressively reduce the dimensionality of the mapping in multiple stages. We start with easier sub-tasks and progressively increase the difficulty of the tasks. For example, a direct mapping from  $\mathbb{R}^{512} \rightarrow \mathbb{R}^{15}$  is instead decomposed into multiple mapping functions  $\mathbb{R}^{512} \rightarrow \mathbb{R}^{256} \rightarrow \mathbb{R}^{128} \rightarrow \mathbb{R}^{64} \rightarrow \mathbb{R}^{32} \rightarrow \mathbb{R}^{15}$ . We formulate the learning problem for  $L$  mapping functions ( $\mathbf{y}^l = f_l(\mathbf{x}; \boldsymbol{\theta})$ ) as:

$$\min_{\boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_L} \sum_{i=1}^n \sum_{j=1}^n \sum_{l=1}^L \alpha_l [d_H(\mathbf{x}_i, \mathbf{x}_j) - d_L(\mathbf{y}_i^l, \mathbf{y}_j^l)]^2 + \lambda \|\boldsymbol{\theta}_l\|_2^2$$

where  $\boldsymbol{\theta}_l$  are the parameters of the  $l$ -th mapping. Appropriately scheduling the  $\alpha_l$  weights enables us to set it up as a curriculum learning problem.

## 2.4 Experiments

### 2.4.1 Intrinsic Dimensionality Estimation

In this section, first we will estimate the intrinsic dimensionality of multiple face representations over multiple datasets of varying complexity. Then, we will evaluate the efficacy of the proposed



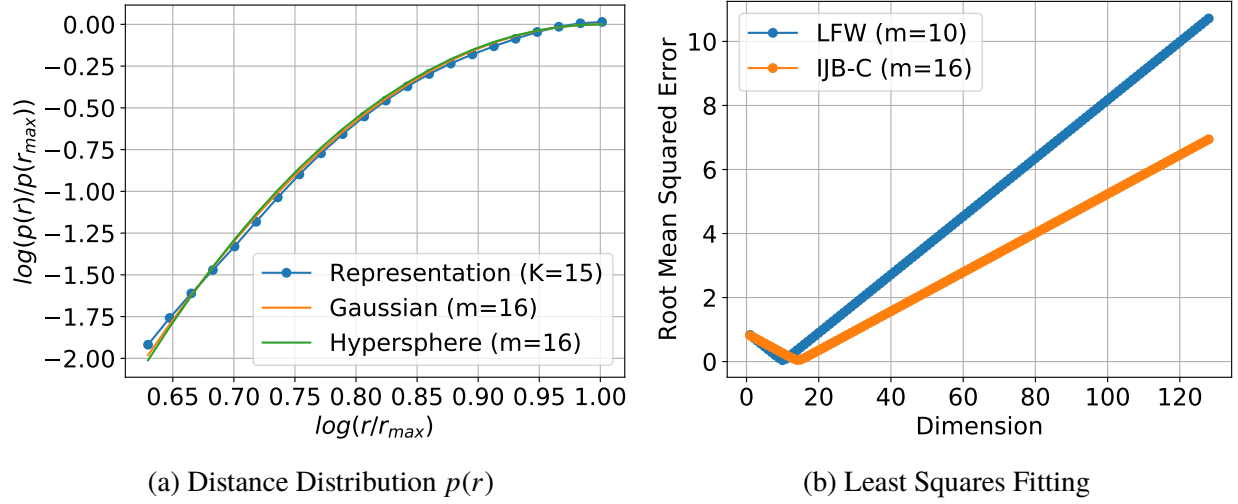


Figure 2.3 **Intrinsic Dimensionality:** (a) Geodesic distance distribution, and (b) global minimum of RMSE.

DeepMDS model in finding the mapping from the ambient to the intrinsic space while maintaining its discriminative ability.

**Datasets.** We consider two different face datasets for face verification, LFW [27], and IJB-C [9]. Recall that DeepMDS is an unsupervised method, so category information associated with the faces is neither used for intrinsic dimensionality estimation nor for learning the mapping from the ambient to intrinsic space.

**Representation Models.** For the face-verification task, we consider multiple publicly available SOTA face embedding models, namely, 128-*dim* FaceNet [12] representation and 512-*dim* SphereFace [13] representation. In addition, we also evaluate a 512-*dim* variant of FaceNet<sup>3</sup> that outperforms the 128-*dim* version. All of these representations are learned from the CASIA WebFace [11] dataset, consisting of 494,414 images across 10,575 subjects.

#### Baseline Methods.

- **Intrinsic Dimensionality:** We select two different algorithms for estimating the intrinsic dimensionality of a given representation, a classical k-nearest neighbor based estimator [2] and “Intrinsic Dimensionality Estimation Algorithm” (IDEA) [3].
- **Dimensionality Reduction:** We compare DeepMDS against three dimensionality reduction

<sup>3</sup><https://github.com/davidsandberg/facenet>

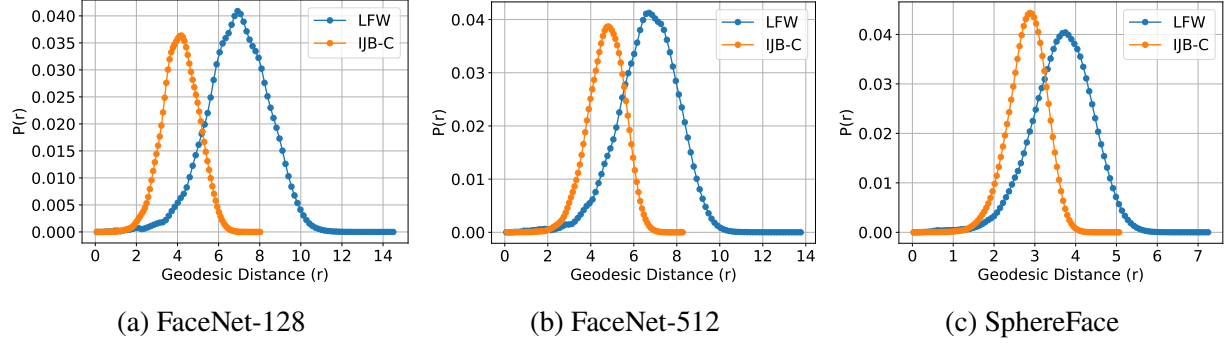
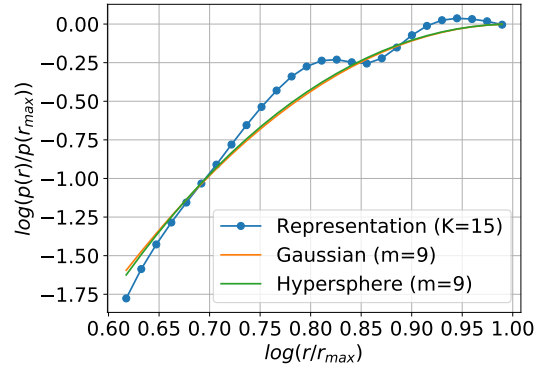
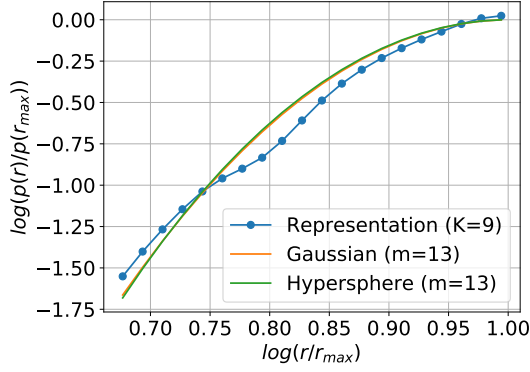
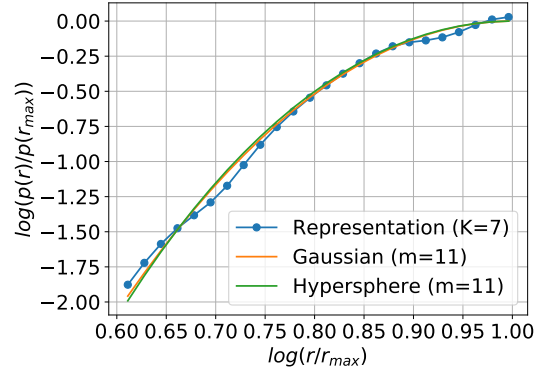
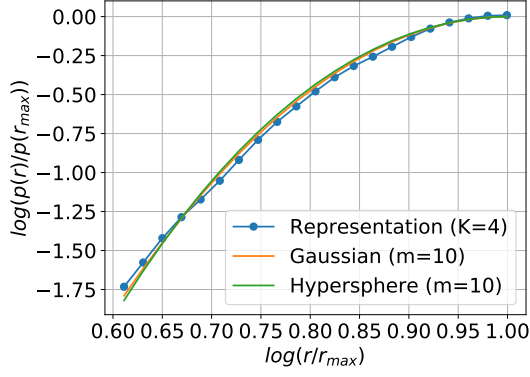


Figure 2.4 Distribution of geodesic distances for different representation models and datasets.

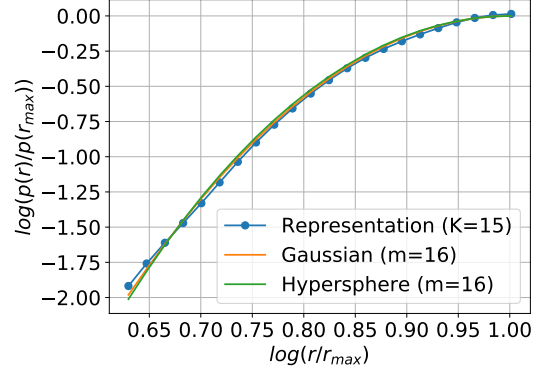
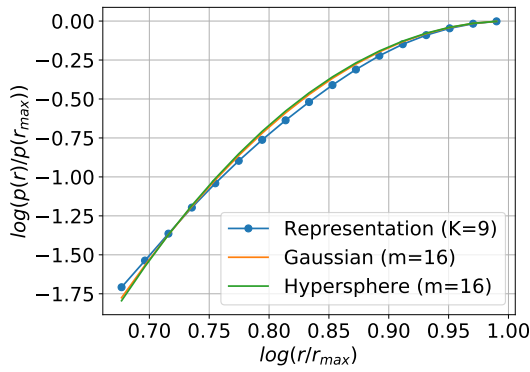
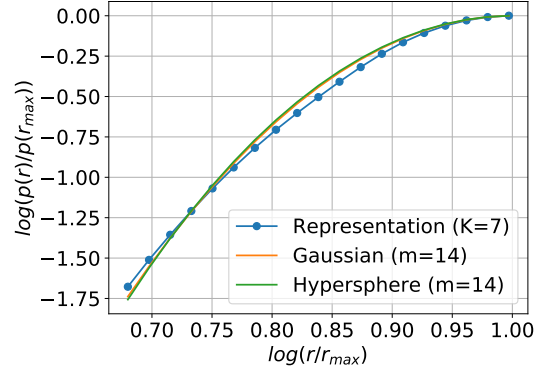
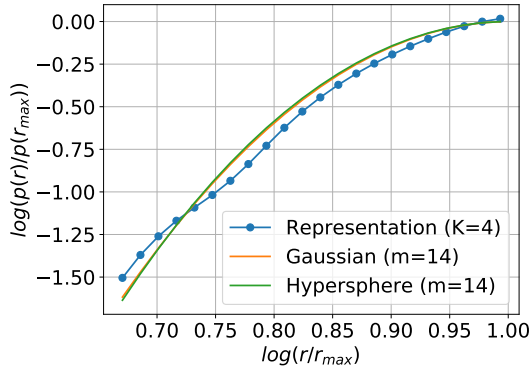
algorithms, principal component analysis (PCA) for linear dimensionality reduction, Isomap [76] and denoising autoencoders [94] (DAE).

**Implementation Details:** The IND estimates for all the methods we evaluate are dependent on the number of neighbors  $k$ . For the baselines,  $k$  is used to compute the parameters of the probability density. For our method,  $k$  parameterizes the construction of the neighborhood graph. For the latter, the choice of  $k$  is constrained by three factors; (1)  $k$  should be small enough to avoid shortcuts between points that are close to each other in the Euclidean space, but are potentially far away in the corresponding intrinsic manifold due to highly complicated local curvatures. (2) On the other hand,  $k$  should also be large enough to result in a connected graph i.e., there are no isolated data samples, and (3)  $k$  that best matches the geodesic distance distribution of a hypersphere of the same IND i.e.,  $k$  that minimizes the RMSE. Fig. 2.3a shows the distance distributions for SphereFace with  $k = 15$ , a 16-hypersphere and a 16-*dim* Gaussian. The close similarity of the pairwise distance distributions of these manifolds in the graph induced geodesic distance space suggests that the IND of SphereFace (512-dim ambient space) is 16. Fig. 2.3b shows the optimal RMSE for SphereFace at different values of  $m$ . The distribution of geodesic distances  $p(r)$  for each of the datasets and representation models is shown in Fig. 2.4. Fig. 2.5 shows the plot of  $\log \frac{\hat{p}_M(r)}{\hat{p}_M(r_{max})}$  vs  $\log \frac{r}{r_{max}}$ , as we vary the number of neighbors  $k$ , for the SphereFace representation model on the LFW and IJB-C datasets.

For all the approaches we select the  $k$ -nearest neighbors using cosine similarity for SphereFace, and arc-length,  $d(\mathbf{x}_1, \mathbf{x}_2) = \cos^{-1} \left( \frac{\mathbf{x}_1^T \mathbf{x}_2}{\|\mathbf{x}_1\| \|\mathbf{x}_2\|} \right)$ , for FaceNet features, as the latter are normalized to



(a) SphereFace on LFW Dataset



(b) SphereFace on IJB-C Dataset

Figure 2.5  $\log \frac{\hat{p}_M(r)}{\hat{p}_M(r_{\max})}$  vs  $\log \frac{r}{r_{\max}}$  plots as we vary number of neighbors  $k$  for sphereface representation model on different datasets.

Table 2.1 Intrinsic Dimensionality: Graph Distance [1]

<i>Representation</i>	dataset	<b>k</b>			
		4	7	9	15
FaceNet-128	LFW	<b>10*</b>	13	11	18
	IJB-C	10	10	10	<b>11*</b>
FaceNet-512	LFW	<b>10*</b>	11	11	17
	IJB-C	11	11	12	<b>12*</b>
SphereFace	LFW	<b>10*</b>	11	13	9
	IJB-C	14	14	16	<b>16*</b>

Table 2.2 Intrinsic Dimensionality: KNN [2]

<i>Representation</i>	dataset	<b>k</b>			
		4	7	9	15
FaceNet-128	LFW	10	10	11	11
	IJB-C	10	10	9	9
FaceNet-512	LFW	8	8	8	9
	IJB-C	10	10	9	9
Sphereface	LFW	6	7	7	8
	IJB-C	6	6	5	5

reside on the surface of a unitary hypersphere. Finally, for simplicity, we round the IND estimates to the nearest integer for all the methods.

**Experimental Results:** Tab. 2.1 reports the IND estimates from the graph method for different values of  $k^4$  and for different representation models across different datasets. Tab. 2.2 and Tab. 2.3 reports the IND estimates from the  $k$ -nearest neighbor approach [2] and IDEA [3], respectively, for different representation models across different datasets that we consider. These approaches are known to underestimate the intrinsic dimensionality [79]. We make a number of observations from our results: (1) Surprisingly, the IND estimates across all the datasets, feature representations and IND methods are significantly lower than the dimensionality of the ambient space, between 10 and 20, suggesting that face representations could, in principle, be almost  $10\times$  to  $50\times$  more compact. (2) Both the  $k$ -NN based estimator [2] and the IDEA estimator [3] are less sensitive to the number of nearest neighbors in comparison to the graph distance based method [1], but are known to underestimate IND for sets with high intrinsic dimensionality [79]. As reported in the tables,

<sup>4\*</sup> denotes final IND estimate that satisfies all constraints on  $k$ .

Table 2.3 Intrinsic Dimensionality: IDEA [3]

<i>Representation</i>	dataset	<b>k</b>			
		4	7	9	15
FaceNet-128	LFW	14	13	13	12
	IJB-C	14	11	10	9
FaceNet-512	LFW	12	10	10	10
	IJB-C	14	11	10	9
Sphereface	LFW	10	9	9	9
	IJB-C	8	7	6	5

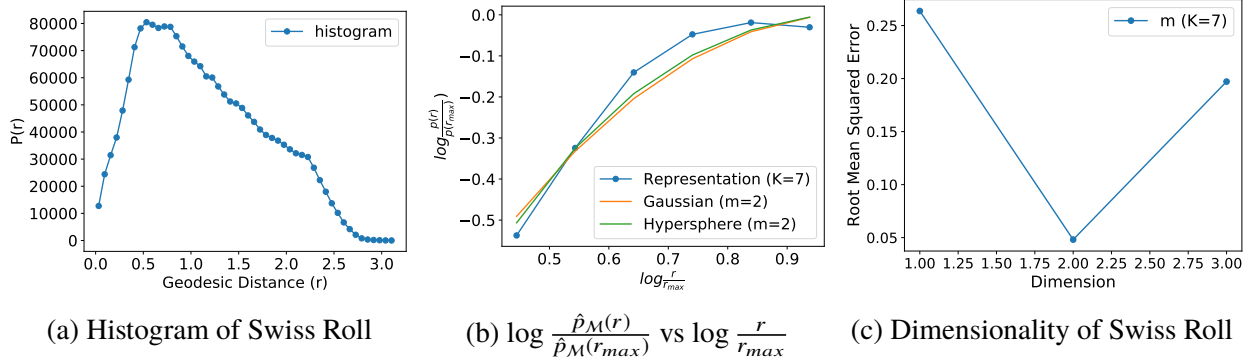


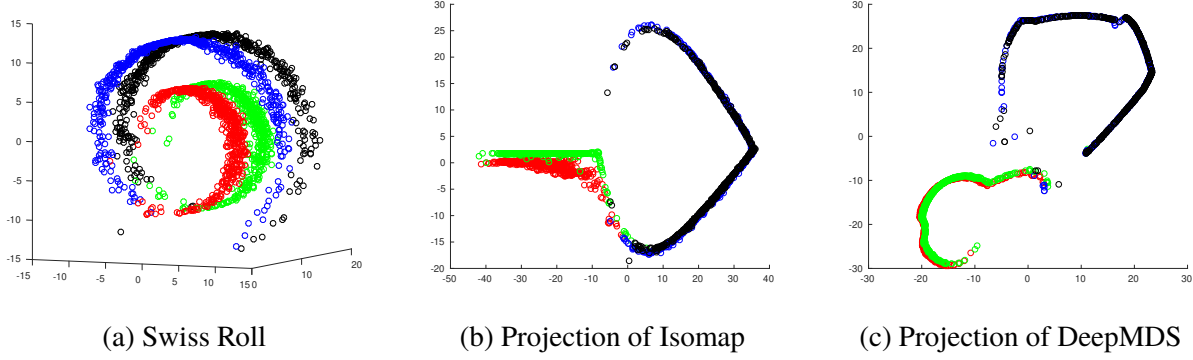
Figure 2.6 Intrinsic Dimensionality of Swiss Roll

IND estimates of the two baseline methods are lower than the estimates of the graph distance based approach that we use.

**Swiss Roll.** We also consider swiss roll dataset, as a means of providing visual validation of the estimated intrinsic space on a known dataset. First, we estimate the intrinsic dimensionality of the swiss roll dataset and then we learn a low-dimensional mapping from the ambient 3-*dim* space to the intrinsic space. We sample 2,000 points from the swiss roll dataset and use these points for the experiments. For this dataset, the intrinsic dimensionality estimate is 2 (See Fig. 2.6), which is indeed the ground truth intrinsic dimensionality of swiss-roll.

## 2.4.2 Intrinsic Space Mapping

Given the estimates of the dimensionality of the intrinsic space, we learn the mapping from the ambient space to a *plausible* intrinsic space with the goal of retaining the discriminative ability of the representation. The true intrinsic representation (IND and space) is unknown and therefore



**Figure 2.7 Swiss Roll:** (a) the original 2000 points from the swiss roll manifold, (b) the 2-*dim* intrinsic space estimated by Isomap, and (3) the 2-*dim* intrinsic space estimated by our proposed method DeepMDS. In both cases, the blue and black points, and correspondingly green and red points, are close together in both the intrinsic and ambient space.

not feasible to validate directly. However, verifying its discriminate power can serve to indirectly validate both the IND estimate and the learned intrinsic space.

**Implementation Details:** We first extract face features through the representation methods i.e., FaceNet-128, FaceNet-512 and SphereFace. The architecture of the proposed DeepMDS model is based on the idea of skip connection laden residual units [70]. We train the mapping from the ambient to intrinsic space in multiple stages with each stage comprising of two residual units. Once the individual stages are trained, all the  $L$  projection models are jointly fine-tuned to maintain the pairwise distances in the intrinsic space. We adopt a similar network structure (residual units) and training strategy (stagewise training and fine-tuning) for the stacked denoising autoencoder baseline. From an optimization perspective, training the autoencoder is more computationally efficient than the DeepMDS model,  $O(n)$  vs  $O(n^2)$ .

The parameters of the network are learned using the Adam [97] optimizer with a learning rate of  $3 \times 10^{-4}$  and the regularization parameter  $\lambda = 3 \times 10^{-4}$ . We observed that using the cosine-annealing scheduler [98] was critical to learning an effective mapping.

**Experimental Results:** We evaluate the efficacy of the learned projections, namely PCA, Isomap and DeepMDS, in the learned intrinsic space and compare their respective performance in the ambient space. Face representations are evaluated in terms of verification (TAR @ FAR) performance. Given

Table 2.4 LFW Face Verification for SphereFace Embedding

<i>Dimension</i>	<b>Dimension Reduction method</b>			
	PCA	Isomap	DAE	DeepMDS
512	96.74%			
256	<b>96.75%</b>	92.88%	77.80%	96.73%
128	<b>96.80%</b>	93.18%	32.95%	96.44%
64	91.71%	95.00%	32.04%	<b>96.50%</b>
32	66.38%	95.31%	11.71%	<b>96.31%</b>
16	32.67%	89.47%	27.53%	<b>95.95%</b>
10 (ID)	16.04%	77.31%	6.73%	<b>92.33%</b>

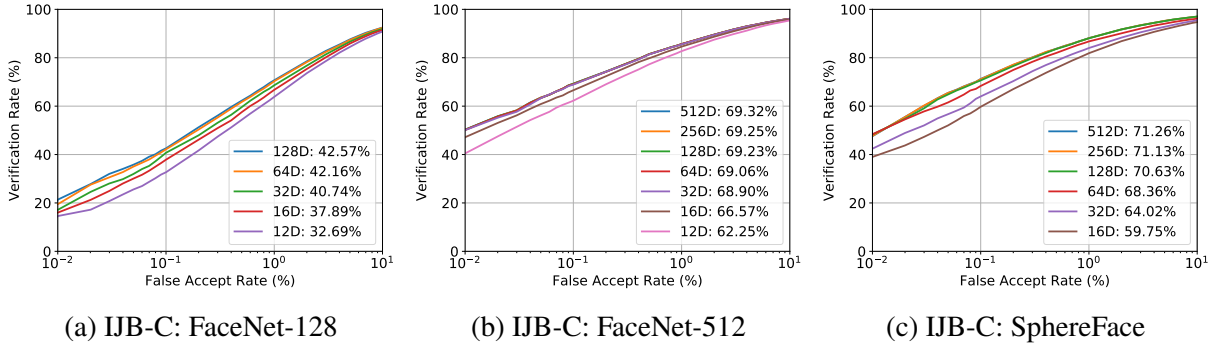


Figure 2.8 **DeepMDS**: Face Verification on IJB-C [9] (TAR @ 0.1% FAR in legend) for the (a) FaceNet-128, (b) FaceNet-512 and (c) SphereFace embeddings.

the IND estimate, designing an appropriate scheme for mapping the intrinsic manifold is much more challenging than the IND estimation itself. To show how dimensionality of the intrinsic space influences the performance of face representations, we evaluate and compare their performance at multiple intermediate spaces.

Face verification is performed on the IJB-C dataset following its verification protocol and on the LFW dataset following the BLUFR [99] protocol. Fig. 2.8 shows the ROC curves for the IJB-C dataset using face representations projected to multiple intermediate spaces and the intrinsic space by DeepMDS. The face verification ROC curves of DeepMDS on LFW dataset for FaceNet-128, FaceNet-512 and SphereFace representation models are shown in Fig. 2.9. Tab. 2.4 reports the verification rate at FAR of 0.1% on the LFW dataset. Fig. 2.10 shows the face verification ROC curves of PCA on the IJB-C and LFW (BLUFR) datasets for all the three representation models. Similarly, Fig. 2.11 and Fig. 2.13 show the face verification ROC curves of the Isomap and Denoising Autoencoder baselines, respectively.

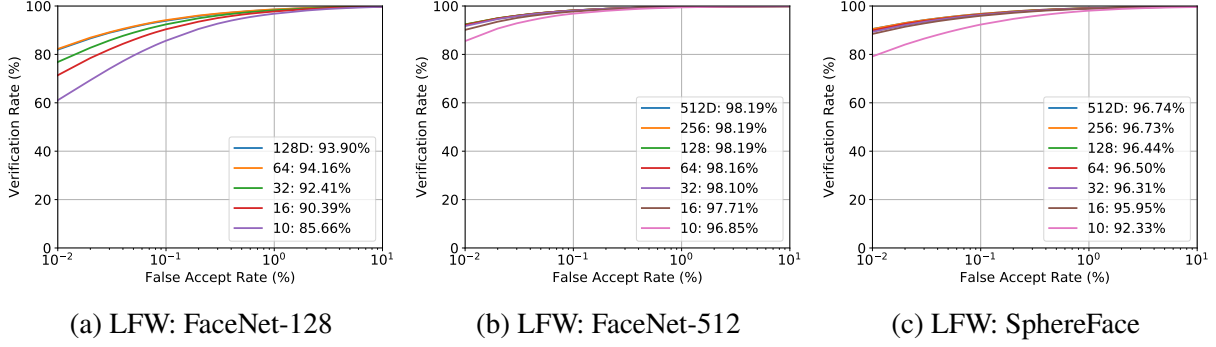


Figure 2.9 **DeepMDS: Face Verification** on LFW (BLUFR) dataset for the (a) FaceNet-128, (b) FaceNet-512 and (c) SphereFace embeddings.

We make the following observations from these results: (1) for all the verification experiments, the performance of the DeepMDS features up to 32 dimensions for faces is comparable to the original 128-*dim* and 512-*dim* features. The 10-*dim* space of DeepMDS on LFW, consisting largely of frontal face images with minimal pose variations and facial occlusions, achieves a TAR of 92.33% at 0.1% FAR, a loss of about 4.5% compared to the ambient space. The 12-*dim* space of DeepMDS on IJB-C, with full pose variations, occlusions and diversity of subject, achieves a TAR of 62.25% at 0.1% FAR, compared to 69.32% in the ambient space. (2) the proposed DeepMDS model is able to learn a low-dimensional space up to the IND with a performance penalty of 5%-10% for compression factors of 30 $\times$  to 40 $\times$  for 512-*dim* representations, underscoring the fact that learning a mapping from ambient to intrinsic space is more challenging than estimating the IND itself. (3) In the task of face verification, we observe that the DeepMDS model is able to retain significantly more discriminative ability compared to the baseline approaches even at high levels of compression. While Isomap is more competitive than the other baselines it suffers from some drawbacks: (i) Due to its iterative nature, it does not provide an explicit mapping function for new (unseen) data samples, while the autoencoder and DeepMDS models can map such data samples. Therefore, Isomap cannot be utilized to evaluate verification accuracy on a validation/test set, and (ii) Computational complexity of Isomap is  $O(n^3)$  and hence does not scale well to large datasets (IJB-C) and needs approximations, such as Nyström approximation [57], for tractability.

**Ablation Study:** Here we demonstrate the efficacy of the stagewise learning process for training



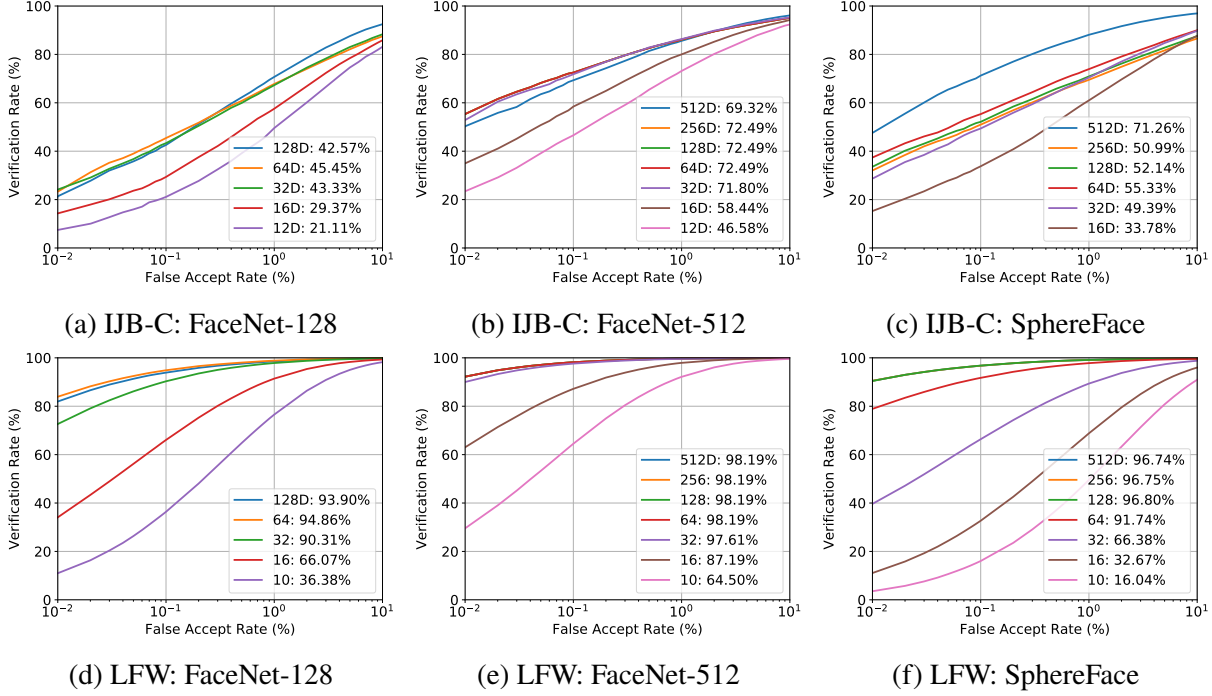


Figure 2.10 **PCA**: Face Verification on IJB-C and LFW (BLUFR) dataset for the (a) FaceNet-128, (b) FaceNet-512 and (c) SphereFace embeddings.

Table 2.5 DeepMDS Training Methods (TAR @ 0.1% FAR)

Method	Direct	Direct+IS	Stagewise + Finetune	Stagewise
<b>TAR</b>	80.25	86.15	90.42	<b>92.33</b>

the DeepMDS model. All models have the same capacity. We consider four variants: (1) **Direct** mapping from the ambient to intrinsic space, (2) **Direct+IS**: direct mapping from ambient to intrinsic space with intermediate supervision at each stage i.e., optimize aggregate intermediate losses, (3) **Stagewise** learning of the mapping, and (4) **Stagewise+Fine-Tune**: the projection model trained stagewise and then fine-tuned. Tab. 2.5 compares the results of these variations on the LFW dataset (BLUFR protocol). Our results suggest that stagewise learning of the non-linear projection models is more effective at progressively disentangling the ambient representation. Similar trend was observed on the larger dataset, IJB-C. In fact, stagewise training with fine-tuning was critical in learning an effective projection, both for DeepMDS as well as DAE.

**Direct Training:** Our findings in this work, that many current DNN representations can be significantly compressed, namely begs the question: *can we directly learn embedding functions*

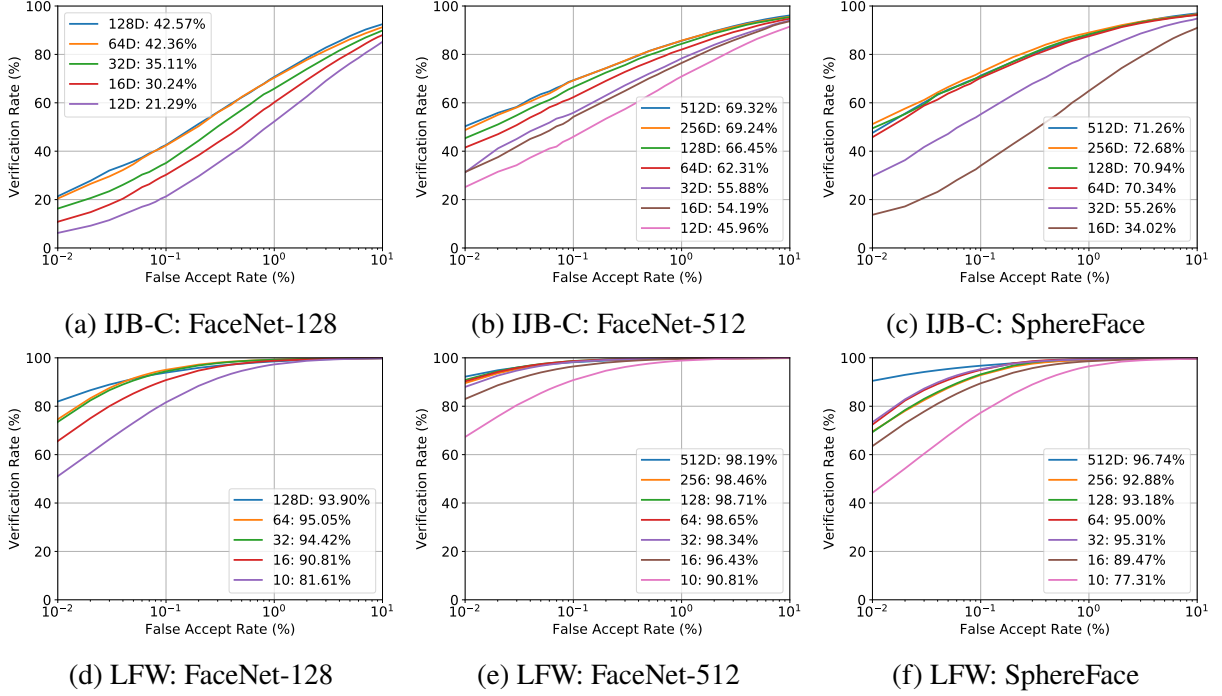
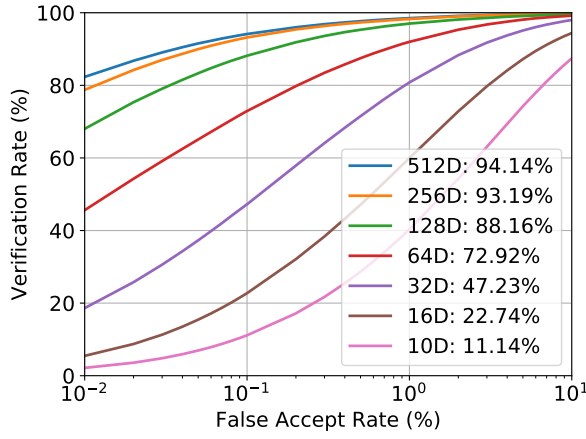


Figure 2.11 **Isomap**: Face Verification on IJB-C and LFW (BLUFR) dataset for the (a) FaceNet-128, (b) FaceNet-512 and (c) SphereFace embeddings.

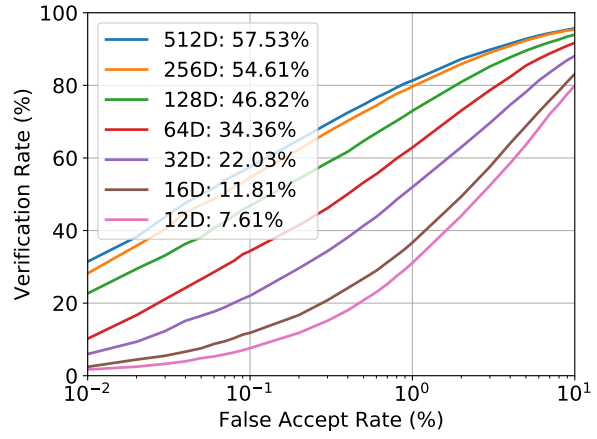
that yield compact and discriminative embeddings in the first place? Taigman et al. [52] study this problem in the context of learning face embeddings, and noted that a compact feature space creates a bottleneck in the information flow to the classification layer and hence increases the difficulty of optimizing the network when training from scratch. Given the significant developments in network architectures and optimization tools since then, we attempt to learn highly compact embedding directly from raw-data, using current best-practices, while circumventing the chicken-and-egg problem of not knowing the target intrinsic dimensionality before learning the embedding function.

We train <sup>5</sup> the Inception ResNet V1 [10] on the CASIA-WebFace [11] for embeddings of different sizes. Fig. 2.12 shows the ROC curves on the LFW and IJB-C datasets. The models suffer significant loss in performance as we decrease the dimensionality of the embeddings. In comparison the proposed DeepMDS based dimensionality reduction retains its discriminative ability even at high levels of compression. These results call for the development of algorithms that can directly learn compact and effective image representations.

<sup>5</sup>We build off of the publicly available implementation at <https://github.com/davidsandberg/facenet>

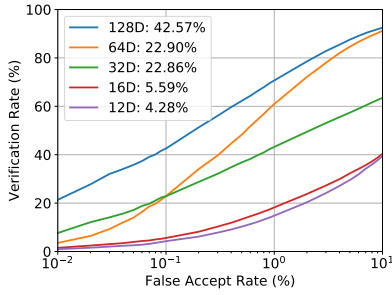


(a) LFW

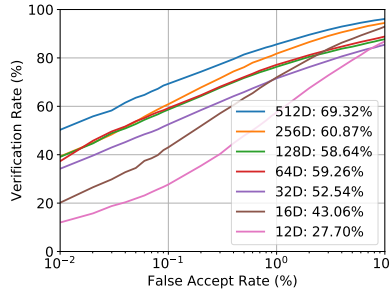


(b) IJB-C

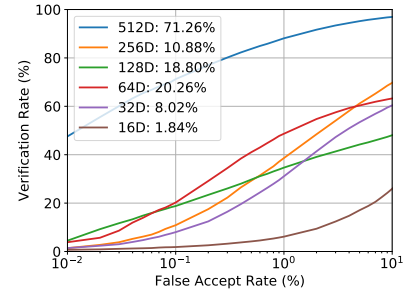
Figure 2.12 ROC curve on LFW and IJB-C datasets for the Inception ResNet V1 [10] model trained with different embedding dimensionality on the CASIA-WebFace [11] dataset.



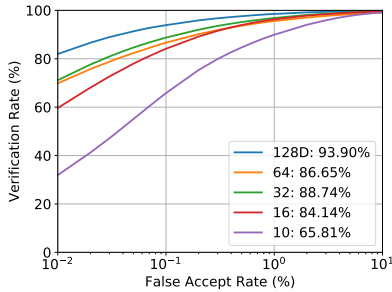
(a) IJB-C: FaceNet-128



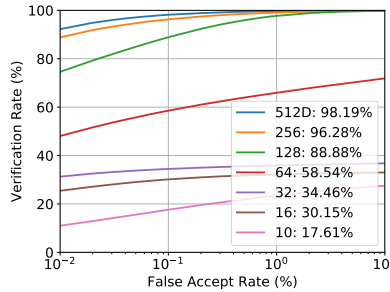
(b) FaceNet-512



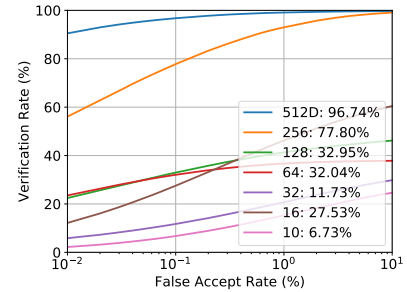
(c) SphereFace



(d) FaceNet-128



(e) FaceNet-512



(f) SphereFace

Figure 2.13 **Denoising Autoencoder:** Face Verification on IJB-C and LFW (BLUFR) dataset for the (a) FaceNet-128, (b) FaceNet-512 and (c) SphereFace embeddings.

## 2.5 Conclusion

The work in this chapter addressed two questions, given a DNN based face representation, what is the minimum number of degrees of freedom in the representation i.e., its intrinsic dimension and can we find a mapping between the ambient and intrinsic space while maintaining the discriminative capability of the representation? Contributions of the work include, (i) a graph induced geodesic distance based approach to estimate the intrinsic dimension, and (ii) DeepMDS, a non-linear projection to transform the ambient space to the intrinsic space. Experiments on multiple DNN based face representations yielded IND estimates of 9 to 16, which are significantly lower than the ambient dimension ( $10\times$  to  $40\times$ ). The DeepMDS model was able to learn a projection from ambient to the intrinsic space while preserving its discriminative ability, to a large extent, on the LFW, and IJB-C datasets. Our findings in this chapter suggest that face representations could be significantly more compact and call for the development of algorithms that can directly learn more compact face representations.

## Chapter 3

# The Capacity of Face Representation

In the previous chapter we estimate IND of face presentation, and propose a dimensionality reduction method to map high-dimensional feature vectors into low-dimensional space. We have yet to give any application related to intrinsic representation space. In this chapter we study a specific problem of face representation applying the dimensionality reduction algorithm based on the notion of IND. The problem is defined as, *given a face representation, how many identities can it resolve?* In other words, *what is the capacity of the face representation?* We cast the face capacity problem in terms of packing bounds on a low-dimensional manifold embedded within a deep representation space. By explicitly accounting for the manifold structure of the representation as well two different sources of representational noise: *epistemic* (model) uncertainty and *aleatoric* (data) variability, our approach is able to estimate the capacity of a given face representation. To demonstrate the efficacy of our approach, we estimate the capacity of two deep neural network based face representations, namely 128-dimensional FaceNet and 512-dimensional SphereFace.

At the core, we leverage advances in deep neural networks (DNNs) for multiple aspects of our solution, relying on their ability to approximate complex non-linear mappings. Firstly, we utilize DNNs to approximate the non-linear function for projecting and unfolding the high-dimensional face representation into a low-dimensional representation, while preserving the local geometric structure of the manifold. Secondly, we utilize DNNs to aid in approximating the density and support of the

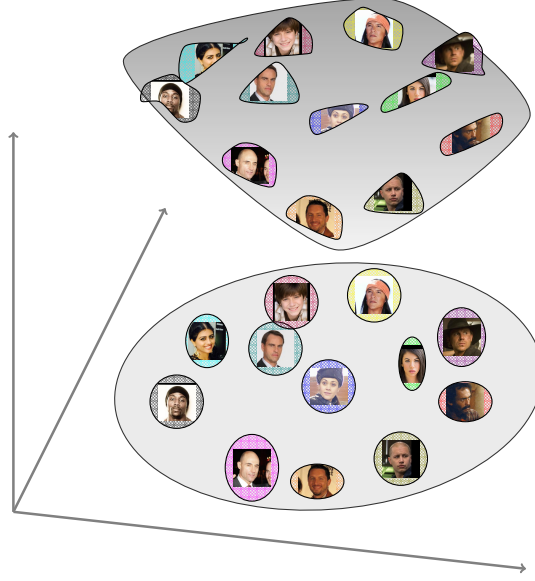


Figure 3.1 An illustration of the geometrical structure of our capacity estimation problem: a low-dimensional manifold  $\mathcal{M} \in \mathbb{R}^m$  embedded in high dimensional space  $\mathcal{P} \in \mathbb{R}^p$ . On this manifold, all the faces lie inside the population hyper-ellipsoid and the embedding of images belonging to each identity or a class are clustered into their own class-specific hyper-ellipsoids. The capacity of this manifold is the number of identities (class-specific hyper-ellipsoids) that can be packed into the population hyper-ellipsoid within an error tolerance or amount of overlap.

low-dimensional manifold in the form of multivariate Gaussian distributions as a function of the desired FAR. The key technical contributions of this work are:

1. Explicitly accounting for and modeling the manifold structure of the face representation in the capacity estimation process. This is achieved through a DNN based non-linear projection and unfolding of the representation into an equivalent low-dimensional Euclidean space while preserving the local geometric structure of the manifold.
2. A noise model for facial embeddings that explicitly accounts for two sources of uncertainty, uncertainty due to data and the uncertainty in the parameters of the representation function.
3. Establishing a relationship between the support of the class-specific manifolds and the discriminant function of a nearest neighbor classifier. Consequently, we can estimate capacity as a function of the desired operating point, in terms of the maximum desired probability of false acceptance error.

4. The first practical attempt at estimating the capacity of DNN based face representations. We consider two such representations, namely, FaceNet [12] and SphereFace [13] consisting of 128-dimensional and 512-dimensional feature vectors, respectively.

Numerical experiments suggest that our proposed model can provide reasonable estimates of capacity. For FaceNet and SphereFace, the upper bounds on the capacity are  $3.5 \times 10^5$  and  $1.1 \times 10^4$ , respectively, for LFW [27], and  $2.2 \times 10^3$  and  $3.6 \times 10^3$ , respectively, for IJB-C [9] at a FAR of 0.1%. This implies that, on average, the representation should have a true accept rate (TAR) of 100% at FAR of 0.1% for  $2.2 \times 10^3$  and  $3.6 \times 10^5$  subject identities for the more challenging IJB-C database and relative less challenging LFW database (see Fig. 1.1c and 1.1f for examples of faces in LFW and IJB-C). As such, the capacity estimates represent an upper bound on the maximal scalability of a given face representation. However, empirically, the FaceNet representation only achieves a TAR of 43% at a FAR of 0.1% on the IJB-C dataset with 3,531 subjects and a TAR of 94% at a FAR of 0.1% on the LFW dataset with 5,749 subjects.

### 3.1 Related Work

The focus of a gajority of the face recognition literature has been on the accuracy of facial recognition on benchmark datasets. In contrast, our goal in this work is to characterize the maximal discriminative capacity of a given face representation at a specified error tolerance.

A number of approaches have been proposed to analyze various performance metrics of biometric recognition systems, primarily using information theoretic concepts. Schmid *et al.* [100, 101] derive analytical bounds on the probability of error and capacity of biometric systems through large deviation analysis on the distribution of the similarity scores. Bhatnagar *et al.* [102] formulated performance indices for biometric authentication. They obtained the capacity of a biometric system following Shannon’s channel capacity formulation along with a rate-distortion theory framework to estimate the FAR. Similarly, Wang *et al.* [103] proposed an approach to model and predict the performance of a face recognition system based on an analysis of the similarity scores. The common

theme across this entire body of work is that the performance bounds of these systems are analyzed purely based on the similarity scores obtained as part of the matching process. In contrast, our work directly analyzes the geometry of the representation space of face recognition systems.

To alleviate the curse of dimensionality, we unfold the face manifold to a lower dimensional space by using the solution we introduce in Ch. 2. In the work of Ch. 2, we first estimate the intrinsic dimensionality of the representation and use the proposed DeepMDS to learn a non-linear mapping that preserves the discriminative performance of the representation to a large extent. But with the goal of estimating the capacity in this chapter, it is opposed to that of preserving the discriminative performance of the representation in the previous chapter. Therefore, while the latter does not necessitate preserving the local geometric structure of the manifold, the former is critically dependent on the ability of the dimensionality reduction technique to preserve the local geometric structure of the manifold.

In the context of estimating distributions, Gaussian Processes [104] are a popular and powerful tool to model distributions over functions, offering nice properties such as uncertainty estimates over function values, robustness to over-fitting, and principled ways for hyper-parameter tuning. A number of approaches have been proposed for modeling uncertainties in deep neural networks [105–107]. Along similar lines, Kendall *et al.* [108] studied the benefits of explicitly modeling *epistemic*<sup>1</sup> (model) and *aleatoric*<sup>2</sup> (data) uncertainties [109] in Bayesian deep neural networks for semantic segmentation and depth estimation tasks. Drawing inspiration from this work, we account for these two sources of uncertainties in the process of mapping a normalized facial image into a low-dimensional face representation.

Capacity estimates to determine the uniqueness of other biometric modalities, namely fingerprints and iris have been reported. Pankanti *et al.* [110] derived an expression for estimating the probability of a false correspondence between minutiae-based representations from two arbitrary fingerprints belonging to two different fingers. Zhu *et al.* [111] later developed a more realistic model of fingerprint individuality through a finite mixture model to represent the distribution of minutiae in

---

<sup>1</sup>Uncertainty due to lack of information about a process.

<sup>2</sup>Uncertainty stemming from the inherent randomness of a process.



fingerprint images, including minutiae clustering tendencies and dependencies in different regions of the fingerprint image domain. Daugman [112] proposed an information theoretic approach to compute the capacity of IrisCode. He first developed a generative model of IrisCode based on Hidden Markov Models and then estimated the capacity of IrisCode by calculating the entropy of this generative model. Adler *et al.* [113] proposed an information theoretic approach to estimate the average information contained within a face representation like Eigenfaces [36].

To the best of our knowledge, no such capacity estimation models have been proposed in the literature for representations of faces. Moreover, the distinct nature of representations for fingerprint<sup>3</sup>, iris<sup>4</sup> and face<sup>5</sup> traits does not allow capacity estimation approaches to carry over from one biometric modality to another. Therefore, we believe that a new model is necessary to establish the capacity of face representations.

## 3.2 Capacity of Face Representations

We first describe the setting of the problem and then describe our solution. A pictorial outline of the approach is shown in Fig. 3.2.

### 3.2.1 Face Representation Model

A face representation model  $M$  is a parametric embedding function that maps a face image  $s$  of identity  $c$  to a vector space  $\mathbf{x} \in \mathbb{R}^p$ , i.e.,  $\mathbf{x} = f_M(s; \theta_{\mathcal{P}})$ , where  $\theta_{\mathcal{P}}$  is the set of parameters of the embedding function. For example, in the case of a linear embedding function like Principal Component Analysis (PCA), the parameter set  $\theta_{\mathcal{P}}$  would represent the eigenvectors. And, in the case of a deep neural network based non-linear embedding function,  $\theta_{\mathcal{P}}$  represents the parameters of the network.

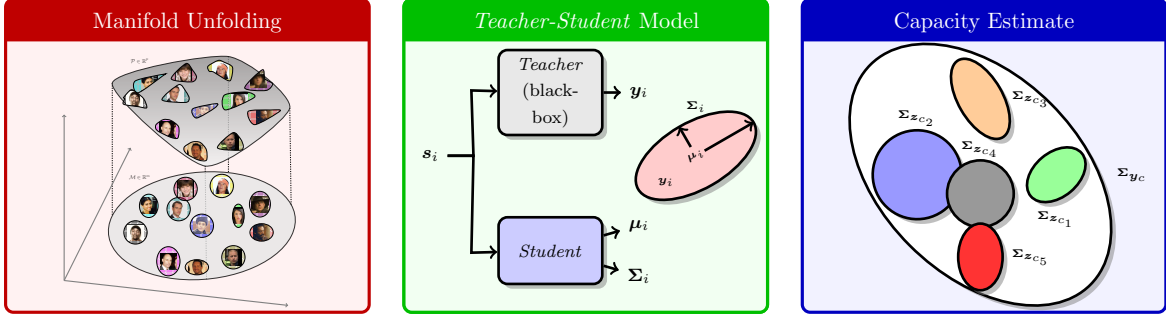
The face embedding process can be approximately cast within the framework of a Gaussian

---

<sup>3</sup>An unordered collection of minutiae points.

<sup>4</sup>A binary representation, called the iris code.

<sup>5</sup>A fixed-length vector of real values.



**Figure 3.2 Overview of Face Representation Capacity Estimation:** We cast the capacity estimation process in the framework of the sphere packing problem on a low-dimensional manifold. To generalize the sphere packing problem, we replace spheres by hyper-ellipsoids, one per class (subject). Our approach involves three steps; (i) Unfolding and mapping the manifold embedded in high-dimensional space onto a low-dimensional space. (ii) *Teacher-Student* model to obtain explicit estimates of the uncertainty (noise) in the embedding due to data as well as the parameters of the representation, and (iii) The uncertainty estimates are leveraged to approximate the density manifold via multi-variate normal distributions (to keep the problem and its analysis tractable), which in turn facilitates an empirical estimate of the capacity of the *teacher* face representation as a ratio of hyper-ellipsoidal volumes.

noise channel as follows. Face representations  $\mathbf{y}$  of an image  $\mathbf{s}$  from the *teacher* are modeled as observations of a true underlying embedding  $\mathbf{x}$  that is corrupted by noise  $\mathbf{z}$ . The nature of the relationship between these entities is determined by the assumptions of a Gaussian channel, namely, (i) additivity of the noise i.e.,  $\mathbf{y} = \mathbf{x} + \mathbf{z}$ , (ii) independence of the true embedding and the additive noise, i.e.,  $\mathbf{x} \perp \mathbf{z}$ , and (iii) all entities,  $\mathbf{y}$ ,  $\mathbf{x}$  and  $\mathbf{z}$  follow a Gaussian distribution, i.e.,  $P_{\mathbf{x}} \sim \mathcal{N}(\boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g)$ ,  $P_{\mathbf{z}} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_z)$  and  $P_{\mathbf{y}} \sim \mathcal{N}(\boldsymbol{\mu}_y, \boldsymbol{\Sigma}_y)$ . Statistical estimates of these parameterized distributions will enable us to compute the capacity of the *teacher* face representation model as described in Section 3.2.3.

For a given black-box face representation, in practice, the embeddings could potentially lie on an arbitrary and unknown low-dimensional manifold. Approximating this manifold through a normal distribution potentially over-estimates the support of the embedding in  $\mathcal{R}^p$ , especially when  $p$  is high, resulting in an over-estimation of the capacity of the representation. To this end, we model the space occupied by the learned face representation as a low-dimensional population manifold  $\mathcal{M} \in \mathbb{R}^m$  embedded within a high-dimensional space  $\mathcal{P} \in \mathbb{R}^p$ . Under this model the features of a given identity  $c$  lie on a manifold  $\mathcal{M}_c \subseteq \mathcal{M}$ . Directly estimating the support and volumes of these

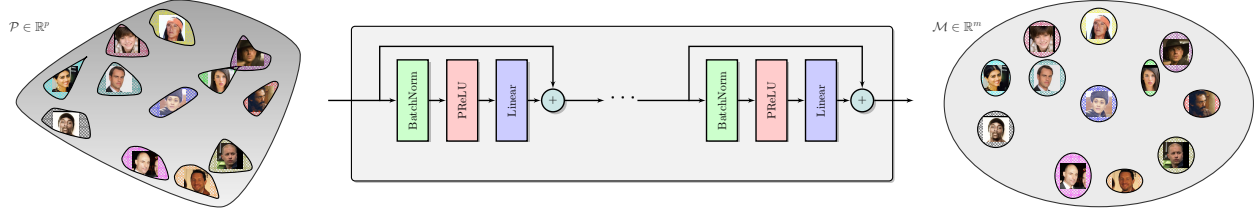


Figure 3.3 **Manifold Unfolding:** A DNN based non-linear mapping is learned to unfold and project the population manifold into a lower dimensional space. The network is optimized to preserve the geodesic distances between pairs of points in the high and low dimensional space.

manifolds is a very challenging task, especially since the manifold could be a highly entangled surface in  $\mathbb{R}^p$ . Therefore, we first learn a mapping that can project and unfold the population manifold onto a low-dimensional space whose density, support and volume can be estimated more reliably. We base our solution for projecting and unfolding the manifold on DeepMDS, which is proposed in the work of Ch. 2.

Specifically, we employ DeepMDS to approximate the non-linear mapping that transforms the population manifold in high-dimensional space,  $\mathbf{x} \in \mathbb{R}^p$ , to the unfolded manifold in low-dimensional space,  $\mathbf{y} \in \mathbb{R}^m$  by a parametric function  $\mathbf{y} = f_P(\mathbf{x}; \boldsymbol{\theta}_M)$  with parameters  $\boldsymbol{\theta}_M$ . We learn the parameters of the mapping within the DeepMDS framework to minimize the following objective,

$$\min_{\boldsymbol{\theta}_M} \sum_{i < j} \left[ d_H(\mathbf{x}_i, \mathbf{x}_j) - d_L(f(\mathbf{x}_i; \boldsymbol{\theta}_M), f(\mathbf{x}_j; \boldsymbol{\theta}_M)) \right]^2 + \lambda \|\boldsymbol{\theta}_M\|_2^2 \quad (3.1)$$

where the second term is a regularizer with a hyperparameter  $\lambda$ . Since our primary goal is to estimate the capacity of the representation we map the manifold into the low-dimensional space while preserving the local geometry of the manifold in the form of pairwise distances. To achieve this goal, we choose  $d_H(\mathbf{x}_i, \mathbf{x}_j) = 1 + \frac{\mathbf{x}_i^T \mathbf{x}_j}{\|\mathbf{x}_i\|_2 \|\mathbf{x}_j\|_2}$  corresponding to the cosine distance between the features in the high dimensional space and  $d_L(\mathbf{y}_i, \mathbf{y}_j) = \|\mathbf{y}_i - \mathbf{y}_j\|_2$  corresponding to the euclidean distance in the low dimensional space. Fig. 3.3 shows an illustration of the DNN based mapping.

### 3.2.2 Estimating Uncertainties in Representations

The projection model learned in the previous section can be used to obtain the population manifold by propagating multiple images from many identities through it. However, this process only provides

point estimates (samples) from the manifold and does not account for the uncertainty in the manifold. Accurately estimating the capacity of the face representation, however, necessitates modeling the uncertainty in the representation stemming from different sources of noise in the process of extracting feature representations from a given facial image.

A probabilistic model for the space of noisy embeddings  $\mathbf{y}$  generated by a black-box facial representation model (*teacher*<sup>6</sup>)  $M_t$  with parameters  $\theta = \{\theta_{\mathcal{P}}, \theta_{\mathcal{M}}\}$  can be formulated as follows:

$$\begin{aligned} p(\mathbf{y}|\mathbf{S}^*, \mathbf{Y}^*) &= \int p(\mathbf{y}|\mathbf{s}, \mathbf{S}^*, \mathbf{Y}^*)p(\mathbf{s}|\mathbf{S}^*, \mathbf{Y}^*)d\mathbf{s} \\ &= \int \int p(\mathbf{y}|\mathbf{s}, \theta)p(\theta|\mathbf{S}^*, \mathbf{Y}^*)p(\mathbf{s}|\mathbf{S}^*, \mathbf{Y}^*)d\theta d\mathbf{s} \end{aligned} \quad (3.2)$$

where  $\mathbf{Y}^* = \{\mathbf{y}_1, \dots, \mathbf{y}_N\}$  and  $\mathbf{S}^* = \{\mathbf{s}_1, \dots, \mathbf{s}_N\}$  are the training samples to estimate the model parameters  $\theta$ ,  $p(\mathbf{y}|\mathbf{s}, \theta)$  is the *aleatoric* (data) uncertainty given a set of parameters,  $p(\theta|\mathbf{S}^*, \mathbf{Y}^*)$  is the *epistemic* (model) uncertainty in the parameters given the training samples and  $p(\mathbf{s}|\mathbf{S}^*, \mathbf{Y}^*) \sim \mathcal{N}(\mu_g, \Sigma_g)$  is the Gaussian approximation (see Section 3.2.3 for justification) of the underlying manifold of noiseless embeddings. Furthermore, we assume that the true mapping between the image  $\mathbf{s}$  and the noiseless embedding  $\mu$  is a deterministic but unknown function i.e.,  $\mu = f(\mathbf{s}, \theta)$ .

The black-box nature of the *teacher* model however only provides  $\mathcal{D} = \{\mathbf{s}_i, \mathbf{y}_i\}_{i=1}^N$ , pairs of facial images  $\mathbf{s}_i$  and their corresponding noisy embeddings  $\mathbf{y}_i$ , a single sample from the distribution  $p(\mathbf{y}|\mathbf{S}^*, \mathbf{Y}^*)$ . Therefore, we learn a *student* model  $M_s$  with parameters  $\mathbf{w}$  to mimic the *teacher* model. Specifically, the *student* model approximates the data dependent *aleatoric* uncertainty  $p(\mathbf{y}_i|\mathbf{s}_i, \mathbf{w}) \sim \mathcal{N}(\mu_i, \Sigma_i)$ , where  $\mu_i$  represents the data dependent mean of the noiseless embedding and  $\Sigma_i$  represents the data dependent uncertainty around the mean. This *student* is an approximation of the unknown underlying probabilistic *teacher*, by which an input image  $\mathbf{s}$  generates noisy embeddings  $\mathbf{y}$  of ideal noiseless embeddings  $\mu$ , for a given set of parameters  $\mathbf{w}$ , i.e.,  $p(\mathbf{y}_i|\mathbf{s}_i, \mathbf{w}) \approx p(\mathbf{y}_i|\mu_i, \theta)$ . Finally, we employ a variational distribution to approximate the *epistemic* uncertainty of the *teacher* i.e.,  $p(\mathbf{w}|\mathbf{S}^*, \mathbf{Y}^*) \approx p(\theta|\mathbf{S}^*, \mathbf{Y}^*)$ .

**Learning:** Given pairs of facial images and their corresponding embeddings from the *teacher* model, we learn a *student* model to mimic the outputs of the *teacher* for the same inputs in accordance

---

<sup>6</sup>We adopt the terminology of teacher-student models from the model compression community [114].

to the probabilistic model described above. We use parameterized functions,  $\mu_i = f(s_i; \mathbf{w}_\mu)$  and  $\Sigma_i = f(s_i; \mathbf{w}_\Sigma)$  to characterize the *aleatoric* uncertainty  $p(y_i|s_i, \mathbf{w})$ , where  $\mathbf{w} = \{\mathbf{w}_\mu, \mathbf{w}_\Sigma\}$ . We choose deep neural networks, specifically convolutional neural networks as our functions  $f(\cdot; \mathbf{w}_\mu)$  and  $f(\cdot; \mathbf{w}_\Sigma)$ . For the *epistemic* uncertainty, while many deep learning based variational inference [105, 115, 116] approaches have been proposed, we use the simple interpretation of dropout as our variational approximation [105]. Practically, this interpretation simply characterizes the uncertainty in the deep neural network weights  $\mathbf{w}$  through a Bernoulli sampling of the weights.

We learn the parameters of our probabilistic model  $\phi = \{\mathbf{w}_\mu, \mathbf{w}_\Sigma, \mu_g, \Sigma_g\}$  through maximum-likelihood estimation i.e., minimizing the negative log-likelihood of the observations  $\mathbf{Y} = \{\mathbf{y}_1, \dots, \mathbf{y}_N\}$ . This translates to optimizing a combination of loss functions:

$$\min_{\phi} \quad \mathcal{L}_s + \lambda \mathcal{L}_g + \gamma \mathcal{L}_{r_s} + \delta \mathcal{L}_{r_g} \quad (3.3)$$

where  $\lambda$ ,  $\gamma$  and  $\delta$  are the weights for the different loss functions,  $\mathcal{L}_{r_s} = \frac{1}{2N} \sum_{i=1}^N \|\Sigma_i\|_F^2$  and  $\mathcal{L}_{r_g} = \frac{1}{2} \|\Sigma_g\|_F^2$  are the regularization terms and  $\mathcal{L}_s$  is the loss function of the student that captures the log-likelihood of a given noisy representation  $\mathbf{y}_i$  under the distribution  $\mathcal{N}(\mu_i, \Sigma_i)$ .

$$\mathcal{L}_s = \frac{1}{2} \sum_{i=1}^N \ln|\Sigma_i| + \frac{1}{2} \text{Trace} \left( \sum_{i=1}^N \Sigma_i^{-1} [(\mathbf{y}_i - \mu_i)(\mathbf{y}_i - \mu_i)^T] \right)$$

$\mathcal{L}_g$  is the log-likelihood of the population manifold of the embedding under the approximation by a multi-variate normal distribution  $\mathcal{N}(\mu_g, \Sigma_g)$ .

$$\mathcal{L}_g = \frac{N}{2} \ln|\Sigma_g| + \frac{1}{2} \text{Trace} \left( \Sigma_g^{-1} \sum_{i=1}^N [(\mathbf{y}_i - \mu_g)(\mathbf{y}_i - \mu_g)^T] \right)$$

For computational tractability we make a simplifying assumption on the covariance matrix  $\Sigma$  by parameterizing it as a diagonal matrix i.e., the off-diagonal elements are set to zero. This parametrization corresponds to independence assumptions on the uncertainty along each dimension of the embedding. The sparse parametrization of the covariance matrix yields two computational

benefits in the learning process. Firstly, it is sufficient for the *student* to predict only the diagonal elements of the covariance matrix. Secondly, positive semi-definitiveness constraints on a diagonal matrix can be enforced simply by forcing all the diagonal elements of the matrix to be non-negative. To enforce non-negativity on each of the diagonal variance values, we predict the log variance,  $l_j = \log \sigma_j^2$ . This allows us to re-parameterize the *student* likelihood in terms of  $\mathbf{l}_i$ :

$$\mathcal{L}_s = \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^d l_i^j + \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^d \frac{(y_i^j - \mu_i^j)^2}{\exp(l_i^j)} \quad (3.4)$$

Similarly, we reparameterize the likelihood of the noiseless embedding as a function of  $\mathbf{l}_g$ , the log variance along each dimension. The regularization terms are also reparameterized as,  $\mathcal{L}_{r_s} = \frac{1}{2N} \sum_{i=1}^N \sum_{j=1}^d \exp(l_i^j)$  and  $\mathcal{L}_{r_g} = \frac{1}{2} \sum_{j=1}^d \exp(l_g^j)$ . We empirically estimate  $\boldsymbol{\mu}_g$  as  $\boldsymbol{\mu}_g = \frac{1}{N} \sum_{i=1}^N \mathbf{y}_i$  and the other parameters  $\boldsymbol{\phi} = \{\mathbf{w}_\mu, \mathbf{w}_\Sigma, \boldsymbol{\Sigma}_g\}$  through stochastic gradient descent [117]. The gradients of the parameters are computed by backpropagating [118] the gradients of the outputs through the network.

**Inference:** The *student* model that has been learned can now be used to infer the uncertainty in the embeddings of the original *teacher* model. For a given facial image  $s$ , the *aleatoric* uncertainty can be predicted by a feed-forward pass of the image  $s$  through the network i.e.,  $\boldsymbol{\mu} = f(s, \mathbf{w}_\mu)$  and  $\boldsymbol{\Sigma} = f(s, \mathbf{w}_\Sigma)$ . The *epistemic* uncertainty can be approximately estimated through Monte-Carlo integration over different samples of model parameters  $\mathbf{w}$ . In practice the parameter sampling is performed through the use of dropout at inference. In summary, the total uncertainty in the embedding of each facial image  $s$  is estimated by performing Monte-Carlo integration over a total of  $T$  evaluations,

$$\hat{\boldsymbol{\mu}}_i = \frac{1}{T} \sum_{t=1}^T \boldsymbol{\mu}_i^t \quad (3.5)$$

$$\hat{\boldsymbol{\Sigma}}_i = \frac{1}{T} \sum_{t=1}^T (\boldsymbol{\mu}_i^t - \hat{\boldsymbol{\mu}}_i) (\boldsymbol{\mu}_i^t - \hat{\boldsymbol{\mu}}_i)^T + \frac{1}{T} \sum_{t=1}^T \boldsymbol{\Sigma}_i^t \quad (3.6)$$

where  $\boldsymbol{\mu}_i^t$  and  $\boldsymbol{\Sigma}_i^t$  are the predicted *aleatoric* uncertainty for each feed-forward evaluation of the network.

### 3.2.3 Manifold Approximation

The student model described in Section 3.2.2 allows us to extract uncertainty estimates of each individual image. Given these estimates the next step is to estimate the density and support of the population and class-specific low-dimensional manifolds.

Multiple existing techniques can be employed for this purpose under different modeling assumptions, ranging from non-parametric models like kernel density estimators and convex-hulls to parametric models like multivariate Gaussian distribution and escribed hyper-spheres. The non-parametric and parametric models span the trade-off between the accuracy of the manifold's shape estimate and the computational complexity of fitting the shape and calculating the volume of the manifold. While the non-parametric models provide more accurate estimates of the density and support of the manifold, the parametric models potentially provide more robust and computationally tractable estimates of the density and volume of the manifolds. For instance, estimating the convex hull of samples in high-dimensional space and its volume is both computationally prohibitive and less robust to outliers.

To overcome the aforestated challenges we approximate the density of the population and class-specific manifolds in the low-dimensional space via multi-variate normal distributions. The choice of the normal distribution approximation is motivated by multiple factors; (a) probabilistically it leads to a robust and computationally efficient estimate of the density of the manifold, (b) geometrically it leads to a hyper-ellipsoidal approximation of the manifold, which in turn allows for efficient and exact estimates of the support and volume of the manifold as a function of the desired false acceptance rate (see Section 3.2.4), and (c) the low-dimensional manifold obtained through projection and unfolding of the high-dimensional representation is implicitly designed, through Eq. 3.1, to cluster the facial images belonging to the same identity, and therefore a normal distribution is a realistic (see Section 3.3.1) approximation of the manifold.

Empirically we estimate the parameters of these distributions as follows. The mean of the population embedding is computed as  $\mu_{y_c} = \frac{1}{C} \sum_{c=1}^C \hat{\mu}^c$ , where  $\hat{\mu}^c = \frac{1}{N_c} \sum_{i=1}^{N_c} \hat{\mu}_i^c$ . The covariance of

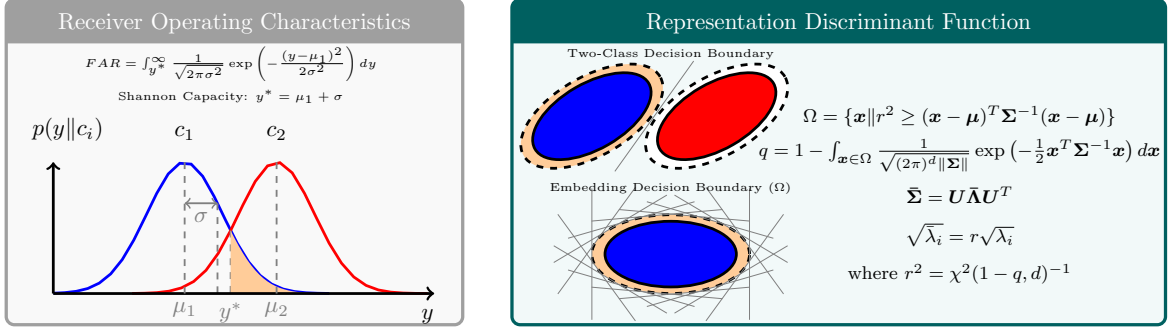


Figure 3.4 **Decision Theory and Capacity:** We illustrate the relation between capacity and the discriminant function corresponding to a nearest neighbor classifier. **Left:** Depiction of the notion of decision boundary and probability of false accept between two identical one dimensional Gaussian distributions. Shannon's definition of capacity corresponds to the decision boundary being one standard deviation away from the mean. **Right:** Depiction of the decision boundary induced by the discriminant function of nearest neighbor classifier. Unlike in the definition of Shannon's capacity, the size of the ellipsoidal decision boundary is determined by the maximum acceptable false accept rate. The probability of false acceptance can be computed through the cumulative distribution function of a  $\chi^2(r^2, d)$  distribution.

the population embedding  $\Sigma_{y_c}$  is estimated as,

$$\begin{aligned} \tilde{\Sigma}^c &= \arg \max_{\hat{\Sigma}^c} \left| \hat{\Sigma}^c + \frac{1}{C} \sum_{c=1}^C (\hat{\mu}^c - \mu_{y_c})(\hat{\mu}^c - \mu_{y_c})^T \right| \\ \Sigma_{y_c} + \Sigma_{z_c} &= \tilde{\Sigma}^c + \frac{1}{C} \sum_{c=1}^C (\hat{\mu}^c - \mu_{y_c})(\hat{\mu}^c - \mu_{y_c})^T \end{aligned} \quad (3.7)$$

where  $\hat{\Sigma}^c = \frac{1}{N_c} \sum_{i=1}^{N_c} \hat{\Sigma}_i^c$ . Along the same lines, the class-specific covariance  $\Sigma_{z_c}$  of a class  $c$  is estimated as,

$$\Sigma_{z_c} = \frac{1}{N_c T} \sum_{i=1}^{N_c} \sum_{t=1}^T \left[ (\mu_i^t - \hat{\mu}_i) (\mu_i^t - \hat{\mu}_i)^T + \Sigma_i^t \right] \quad (3.8)$$

### 3.2.4 Decision Theory and Model Capacity

Thus far, we developed the tools necessary to characterize the face representation manifold and estimate its density. In this section we will determine the support and volume of the population and class-specific manifolds as a function of the specified false accept rate (FAR).



Our representation space is composed of two components: the population manifold of all the classes approximated by a multi-variate Gaussian distribution and the embedding noise of each class approximated by a multi-variate Gaussian distribution. Under these settings, the decision boundaries between the classes that minimizes the classification error rate are determined by discriminant functions [119]. As illustrated in Fig. 3.4, for a two-class problem, the discriminant function is a hyper-plane in  $\mathbb{R}^d$ , with the optimal hyper-plane being equidistant from both the classes. Moreover, the separation between the classes determines the operating point and hence the FAR. In the multi-class setting the optimal discriminant function is the surface encompassed by all the pairwise hyper-planes, which asymptotically reduces to a high-dimensional hyper-ellipsoid. The support of this enclosing hyper-ellipsoid can be determined by the desired operating point in terms of the maximal error probability of false acceptance.

Under the multi-class setting, the capacity estimation problem is equivalent to the geometrical problem of ellipse packing, which seeks to estimate the maximum number of small hyper-ellipsoids that can be packed into a larger hyper-ellipsoid. In the context of face representations the small hyper-ellipsoids correspond to the class-specific enclosing hyper-ellipsoids as described above while the large hyper-ellipsoid corresponds to the space spanned by the population of all classes. The volume  $V$  of a hyper-ellipsoid corresponding to a Mahalanobis distance  $r^2 = (\mathbf{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu})$  with covariance matrix  $\boldsymbol{\Sigma}$  is given by the following expression,  $V = V_d |\boldsymbol{\Sigma}|^{\frac{1}{2}} r^d$ , where  $V_d$  is the volume of the  $d$ -dimensional hypersphere. An upper bound on the capacity of the face representation can be computed simply as the ratio of the volumes of the population and the class-specific hyper-ellipsoids,

$$\begin{aligned}
C &\leq \left( \frac{V_{y_c, z_c}}{V_{z_c}} \right) \\
&= \left( \frac{V_d |\boldsymbol{\Sigma}_{y_c} + \boldsymbol{\Sigma}_{z_c}|^{\frac{1}{2}} r_{y_c}^d}{V_d |\boldsymbol{\Sigma}_{z_c}|^{\frac{1}{2}} r_{z_c}^d} \right) = \left( \frac{|\boldsymbol{\Sigma}_{y_c} + \boldsymbol{\Sigma}_{z_c}|^{\frac{1}{2}} r_{y_c}^d}{|\boldsymbol{\Sigma}_{z_c}|^{\frac{1}{2}} r_{z_c}^d} \right) \\
&= \left( \frac{|\bar{\boldsymbol{\Sigma}}_{y_c, z_c}|^{\frac{1}{2}}}{|\bar{\boldsymbol{\Sigma}}_{z_c}|^{\frac{1}{2}}} \right)
\end{aligned} \tag{3.9}$$

where  $V_{y_c, z_c}$  is the volume of population hyper-ellipsoid and  $V_{z_c}$  is the volume of the class-specific

hyper-ellipsoid. The size of the population hyper-ellipsoid  $r_{y_c}$  is chosen such that a desired fraction of all the classes lie within the hyper-ellipsoid and  $r_{z_c}$  determines the size of the class-specific hyper-ellipsoid.  $\bar{\Sigma}_{y_c, z_c}$  and  $\bar{\Sigma}_{z_c}$  are the effective sizes of the enclosing population and class-specific hyper-ellipsoids respectively. For each of the hyper-ellipsoids the effective radius along the  $i$ -th principal direction is  $\sqrt{\bar{\lambda}_i} = r\sqrt{\lambda_i}$ , where  $\sqrt{\lambda_i}$  is the radius of the original hyper-ellipsoid along the same principal direction.

This geometrical interpretation of the capacity reduces to the Shannon capacity [120] when  $r_{y_c}$  and  $r_{z_c}$  are chosen to be the same i.e., when  $r_{y_c} = r_{z_c}$ . Consequently, in this instance, the choice of  $r_{y_c}$  for the population hyper-ellipsoid implicitly determines the boundary of separation between the classes and hence the operating false acceptance rate (FAR) of the embedding. For instance, when computing the Shannon capacity of the face representation choosing  $r_{y_c}$  such that 95% of the classes are enclosed within the population hyper-ellipsoid would implicitly correspond to operating at a FAR of 5%. However, practical face recognition systems need to operate at lower false accept rates, dictated by the desired level of security.

The geometrical interpretation of the capacity described in Eq. 3.9 directly enables us to compute the representation capacity as a function of the desired operating point as determined by its corresponding false accept rate. The size of the population hyper-ellipsoid  $r_{y_c}$  will be determined by the desired fraction of classes to enclose or alternatively other geometric shapes like the minimum volume enclosing hyper-ellipsoid or the maximum volume inscribed hyper-ellipsoid of a finite set of classes, both of which correspond to a particular fraction of the population distribution. Similarly, the desired false accept rate  $q$  determines the size of the class-specific hyper-ellipsoid  $r_{z_c}$ .

Let  $\Omega = \{\mathbf{x} \mid r^2 \geq (\mathbf{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu})\}$  be the enclosing hyper-ellipsoid. Without loss of generality, assuming that the class-specific hyper-ellipsoid is centered at the origin, the false accept rate  $q$  can be computed as,

$$q = 1 - \int_{\mathbf{x} \in \Omega} \frac{1}{\sqrt{(2\pi)^d |\boldsymbol{\Sigma}|}} \exp\left(-\frac{\mathbf{x}^T \boldsymbol{\Sigma}^{-1} \mathbf{x}}{2}\right) d\mathbf{x} \quad (3.10)$$

Reparameterizing the integral as  $\mathbf{y} = \Sigma^{-\frac{1}{2}}\mathbf{x}$ , we have  $\Omega = \{\mathbf{y} \mid r^2 \geq \mathbf{y}^T \mathbf{y}\}$  and,

$$q = 1 - \int_{\mathbf{y} \in \Omega} \frac{1}{\sqrt{(2\pi)^d}} \exp\left(-\frac{\mathbf{y}^T \mathbf{y}}{2}\right) d\mathbf{y} \quad (3.11)$$

where  $\{y_1, \dots, y_n\}$  are independent standard normal random variables. The Mahalanobis distance  $r^2$  is distributed according to the  $\chi^2(r^2, d)$  distribution with  $d$  degrees of freedom and  $1 - q$  is the cumulative distribution function of  $\chi^2(r^2, d)$ . Therefore, given the desired FAR  $q$ , the corresponding Mahalanobis distance  $r_{z_c}$  can be obtained from the inverse CDF of the  $\chi^2(r_z^2, d)$  distribution. Along the same lines, the size of the population hyper-ellipsoid  $r_{y_c}$  can be estimated from the inverse CDF of the  $\chi^2(r_y^2, d)$  distribution given the desired fraction of classes to encompass. These estimates of  $r_{z_c}$  and  $r_{y_c}$  can be utilized in Eq. 3.9 to estimate the capacity as a function of the desired FAR. Algorithm 1 provides a high-level outline of our complete capacity estimation procedure.

---

**Algorithm 1** Face Representation Capacity Estimation

---

**Input:** Representation  $f_M(\cdot, \theta_P)$ , a face dataset and desired FAR.

**Output:** Capacity estimate at specified FAR.

**Step 1:** Learn parametric mapping  $f_P(\cdot, \theta_M) : \mathbf{x} \rightarrow \mathbf{y}$  (Eq. 3.1)

**Step 2:** Learn *student* model  $M_s$  to mimic and provide uncertainty estimates of *teacher*  $M_t = (M, P)$  (Eq. 3.3)

**Step 3:** Estimate density and support  $\Sigma_{y_c}$  of population manifold (Eq. 3.7)

**Step 4:** Estimate density and support  $\Sigma_{z_c}$  of class-specific manifolds (Eq. 3.8)

**Step 5:** Obtain  $r_{y_c}$  and  $r_{z_c}$  for desired population fraction and FAR, respectively (Eq. 3.11)

**Step 6:** Obtain capacity estimate for desired population fraction and FAR using  $r_{y_c}$  and  $r_{z_c}$  (Eq. 3.9)

---

### 3.3 Numerical Experiments

In this section we will, (a) illustrate the capacity estimation process on a two-dimensional toy example, (b) estimate the capacity of a deep neural network based face representation model, specifically FaceNet and SphereFace on multiple datasets of increasing complexity, and (c) study the effect of different design choices of the proposed capacity estimation approach.

Table 3.1 Capacity of Two-Dimensional Toy Example at 1% FAR

Manifold Support	Population				Class (max area)				Estimated Capacity	Ground-Truth Capacity
	Covariance		Area		Covariance		Area			
	Estimate	Ground-Truth	Estimate	Ground-Truth	Estimate	Ground-Truth	Estimate	Ground-Truth		
Ellipse	$\begin{bmatrix} 10.84 & 0.56 \\ 0.56 & 11.57 \end{bmatrix}$	$\begin{bmatrix} 10.34 & 0.71 \\ 0.71 & 11.79 \end{bmatrix}$	35.15	34.62	$\begin{bmatrix} 4.96 & 0.47 \\ 0.47 & 6.54 \end{bmatrix}$	$\begin{bmatrix} 4.18 & 0.97 \\ 0.97 & 5.86 \end{bmatrix}$	17.84	15.25	1.97	2.27
Convex Hull	–	–	403.91	–	–	–	102.65	–	3.93	2.27

### 3.3.1 Two-Dimensional Toy-Example

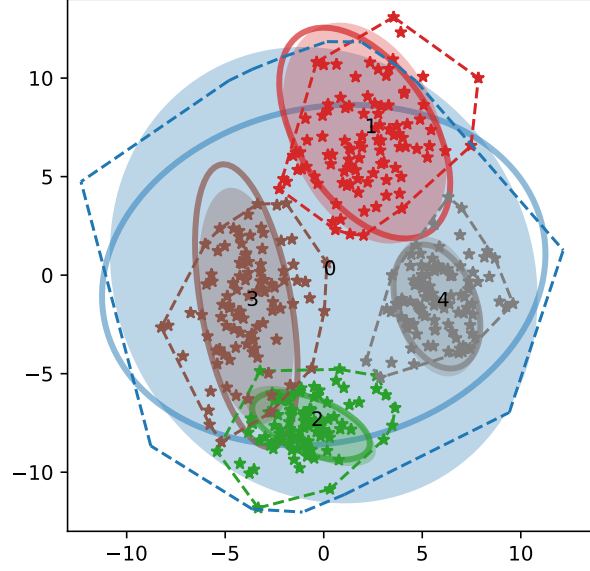


Figure 3.5 **Sample Representation Space:** Illustration of a two-dimensional space where the underlying population and class-specific representations (we show four classes) are 2-D Gaussian distributions (solid ellipsoids). Samples from the classes (colored ★) are utilized to obtain estimates of this underlying population and class-specific distributions (solid lines). As a comparison, the support of the samples in the form of a convex hull are also shown (dashed lines).

We consider an illustrative example to demonstrate the capacity estimation process given a constellation of classes in a two-dimensional representation space. We model the distribution of the population space of classes (class centers to be specific) as a multi-variate normal distribution, while the feature space of each class is modeled as a two-dimensional normal distribution. From this model, we sample 100 different classes from the underlying population distribution and for each of these classes we sample features from the ground truth multi-variate normal distribution for that class. From these samples, we estimate the covariance matrix of the population space distribution and that of the individual classes.

Fig. 3.5 shows the representation space, including the population space and four different classes corresponding to classes with the minimum, mean, median and maximum area from among the 100 population classes that were sampled. As a comparison we also obtain the support of the population and the classes through the convex hull of the samples, even as this presents a number of practical challenges: (1) estimating convex hull in high-dimensions is computationally challenging, (2) convex hull overestimates the support due to outliers, and (3) cannot be easily adapted to obtain support as a function of desired FAR.

The capacity of the representation is now estimated as the ratio of the support area of the population and the class with median area, respectively. Tab. 3.1 shows the capacity estimates so obtained for this simplified representation space. Results on this example suggests that the ellipsoidal approximation of the representation is able to provide more accurate estimates of the capacity of the representation in comparison to the convex hull. Modeling the support of the representation through convex hulls is severely affected by outliers, resulting in an overestimate of the underlying support and area of the representation leading to overestimates of its capacity.

### 3.3.2 Datasets and Face Representation Model

**Datasets.** We utilize multiple large-scale face datasets, both for learning the *teacher* and *student* models as well as for estimating the capacity of the *teacher*. CASIA WebFace dataset [11] is used for training both the *teacher* and *student* models. The capacity of the *teacher* is estimated on LFW [27], IJB-A [29], IJB-B [30], and IJB-C [9].

**Face Representation Models.** We estimate the capacity of two different face representation models: (i) FaceNet introduced by Schroff *et al.* [12], and (ii) SphereFace introduced by Liu *et al.* [13]. These models are illustrative of the state-of-the-art representations for face recognition.

The manifold projection and unfolding function is modeled as a multi-layer deep neural network with multiple residual [50] modules consisting of fully-connected layers. Therefore, for a given image, the low-dimensional representation can be obtained by propagating the image through the *original* face representation model and then through the manifold projection model. We refer to the

combined model, i.e., *original* representation and the projection model, as the *teacher* model. Since the *student* model is purposed to mimic the *teacher* model, we base the *student* network architecture on the *teacher*'s<sup>7</sup> architecture with a few notable exceptions. First, we introduce dropout before every convolutional layer of the network, including all the convolutional layers of the inception [121] and residual [50] modules and every linear layer of the manifold projection and unfolding modules. Second, the last layer of the network is modified to generate two outputs  $\mu$  and  $\Sigma$  instead of the output of the *teacher* i.e., sample  $y$  of the noisy embedding.

### 3.3.3 Face Recognition Performance

Below we provide implementation details for learning the manifold projection and the *student* networks. Subsequently, we demonstrate the ability of the *student* model to maintain the discriminative performance of the *original* models.

**Implementation Details:** We use pre-trained models for both FaceNet<sup>8</sup> and SphereFace<sup>9</sup> as our *original* face representation models. Before we extract features from these models, the face images are pre-processed and normalized to a canonical face image. The faces are detected and normalized using the joint face detection and alignment system introduced by Zhang *et al.* [33]. Given the facial landmarks, the faces are normalized to a canonical image of size 182×182 from which RGB patches of size 160×160 are extracted as the input to the networks.

Given the features extracted from the *original* representation, we train the manifold projection and unfolding networks on the CASIA WebFace dataset. The model is trained to minimize the multi-dimensional scaling loss function described in Eq. 3.1 on randomly selected pairs of features vectors  $x_i$  and  $x_j$  from the dataset. Training is performed using the Adam [97] optimizer with a learning rate of 3e-4 and the regularization parameter  $\lambda = 3 \times 10^{-4}$ . We use a batch size of 256 image pairs and train the model for about 100 epochs.

---

<sup>7</sup>In the scenario where the *teacher* is a black-box model, the design of the student network architecture needs more careful consideration but it also affords more flexibility. See Fig. 3.2 for an illustration of this process.

<sup>8</sup><https://github.com/davidsandberg/facenet>

<sup>9</sup><https://github.com/wy1iu/sphereface>

The *student* is trained to minimize the loss function defined in Eq. 3.3, where the hyperparameters are chosen through cross-validation. Training is performed through stochastic gradient descent with Nesterov Momentum 0.9 and weight decay 0.0005. We used a batch size of 64, a learning rate of 0.01 that is dropped by a factor of 2 every 20 epochs. We observed that it is sufficient to train the *student* model for about 100 epochs for convergence. The *student* model includes dropout with a probability of 0.05 after each convolutional layer and with a probability of 0.2 after each fully-connected layer in the manifold projection layers. At inference each image is passed through the *student* network 1,000 times as a way of performing Monte-Carlo integration through the space of network parameters  $\{\mathbf{w}_\mu, \mathbf{w}_\Sigma\}$ . These sampled outputs are used to empirically estimate the mean and covariance of the image embedding.

**Experiments:** We evaluate and compare the performance of the *original* and *student* models on the four test datasets, namely, LFW, IJB-A, IJB-B and IJB-C. To evaluate the *student* model we estimate the face representation through Monte-Carlo integration. We pass each image through the *student* model 1,000 times to extract  $\{\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i\}_{i=1}^{1000}$  and compute  $\boldsymbol{\mu} = \frac{1}{1000} \sum_{i=1}^{1000} \boldsymbol{\mu}_i$  as the representation. Following standard practice, we match a pair of representations through a nearest neighbor classifier i.e., by computing the euclidean distance  $d_{ij} = \|\mathbf{y}_i - \mathbf{y}_j\|_2^2$  between the low-dimensional projected feature vectors  $\mathbf{y}_i$  and  $\mathbf{y}_j$ .

We evaluate the face representation models on the LFW dataset using the BLUFR protocol [99] and follow the prescribed template based matching protocol, where each template is composed of possibly multiple images of the class, for the IJB-A, IJB-B and IJB-C datasets. Following the protocol in [122], we define the match score between templates as the average of the match scores between all pairs of images in the two templates.

Fig. 3.6 and Tab. 3.2 report the performance of the *original* and *student* models, both FaceNet and SphereFace, on each of these datasets at different operating points. This comparison accounts for both the ability of the projection model to maintain the performance of the *original* high-dimensional representation as well as the ability of the *student* to mimic the *teacher* while providing uncertainty estimates. We make the following observations: (1) The performance of DNN based representation

Table 3.2 Face Recognition Results for FaceNet, SphereFace and State-of-the-Art (The state-of-the-art face representation models are not available in the public domain)

Dataset	<i>Original</i> : FaceNet		<i>Student</i> : FaceNet		<i>Original</i> : SphereFace		<i>Student</i> : SphereFace		State-of-the-Art	
	0.1% FAR	1% FAR	0.1% FAR	1% FAR	0.1% FAR	1% FAR	0.1% FAR	1% FAR	0.1% FAR	1% FAR
LFW (BLUFR)	93.90	98.51	92.83	98.28	96.74	99.11	95.49	98.79	98.88 [123]	N/A
IJB-A	45.92	70.26	43.84	71.72	65.06	85.97	64.13	85.25	94.8	97.1 [124]
IJB-B	48.31	74.47	45.56	74.10	67.58	80.81	64.02	80.63	93.7	97.5 [125]
IJB-C	42.57	78.53	40.74	76.75	71.26	91.67	64.02	88.33	94.7	98.3 [125]

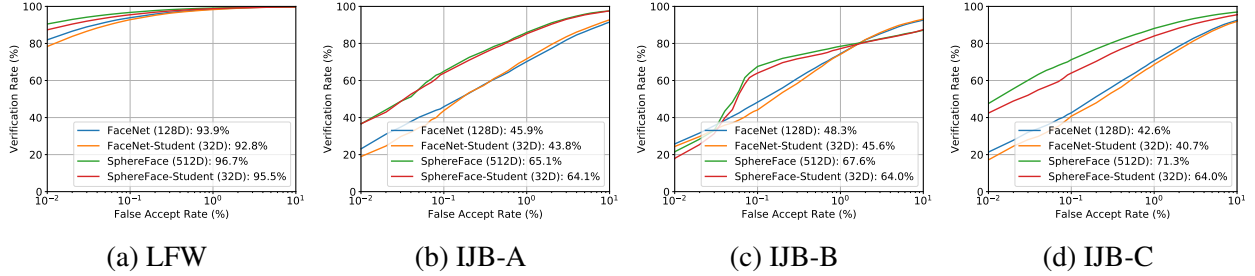


Figure 3.6 Face recognition performance of the *original* and *student* models on different datasets. We report the face verification performance of both FaceNet and SphereFace face representations, (a) LFW evaluated through the BLUFR protocol, (b) IJB-A, (c) IJB-B, and (d) IJB-C evaluated through their respective matching protocol.

on LFW, consisting largely of frontal face images with minimal pose variations and facial occlusions, is comparable to the state-of-the-art. However, its performance on IJB-A, IJB-B and IJB-C, datasets with large pose variations, is lower than state-of-the-art approaches. This is due to the template generation strategy that we employ and the fact that unlike these methods we do not fine-tune the DNN model on the IJB-A, IJB-B and IJB-C training sets. We reiterate that our goal in this work is to estimate the capacity of a generic face representation as opposed to achieving the best verification performance on each individual datasets., and (2) Our results indicate that the *student* models are able to mimic the *teacher* models very well as demonstrated by the similarity of the receiving operating curves.





### 3.3.4 Face Representation Capacity

Having demonstrated the ability of the *student* model to be an effective proxy for the *original* representation manifold, we indirectly estimate the capacity of the *original* model by estimating the capacity of the *student* model.

**Implementation Details:** We estimate the capacity of the face representations by evaluating



Table 3.3 Capacity of Face Representation Model at 1% FAR

Dataset	Faces	FaceNet	SphereFace
LFW		$4.3 \times 10^6$	$2.6 \times 10^5$
IJB-A		$6.3 \times 10^4$	$3.2 \times 10^6$
IJB-B		$6.4 \times 10^4$	$2.4 \times 10^5$
IJB-C		$2.7 \times 10^4$	$8.4 \times 10^4$

Eq. 3.9. For each of the datasets we empirically determine the shape and size of the population hyper-ellipsoid  $\Sigma_{y_c}$  and the class-specific hyper-ellipsoids  $\Sigma_{z_c}$ . These quantities are computed through the predictions obtained by sampling the weights ( $w_\mu, w_\Sigma$ ) of the model, via dropout. We obtain 1,000 such predictions for a given image, by feeding the image through the *student* network a 1,000 different times with dropout. For robustness against outliers we only consider classes with at least two images per class for LFW and five images per class for all the other datasets for the capacity estimates.

**Capacity Estimates:** Tab. 3.3 reports the capacity of DNN based face representations estimated on different datasets at 1% FAR (i.e., when  $r_{y_c} = r_{z_c}$ ). We make the following observations from our numerical results: The upper bound on the capacity estimate of the FaceNet and SphereFace models in constrained scenarios (LFW) is of the order of  $\approx 10^6$ , in unconstrained environments (IJB-A, IJB-B and IJB-C) is of the order of  $\approx 10^5$  under the general model of a hyper-ellipsoid with the class corresponding to maximum noise. Therefore, theoretically, the representation should be able to resolve  $10^6$  and  $10^5$  subjects with a true acceptance rate (TAR) of 100% at a FAR of 1% under the constrained and unconstrained operational settings, respectively. While this capacity estimate is on the order of the population of a large city, in practice, the performance of the representation is lower

than the theoretical performance, about 95% across only 10,000 subjects in the constrained and only 50% across 3,531 subjects in the unconstrained scenarios. These results suggest that our capacity estimates are an upper bound on the actual performance of face recognition systems in practice, especially under unconstrained scenarios. The relative order of the capacity estimates, however, mimics the relative order of the verification accuracy on these datasets as shown in Fig. 3.7c.

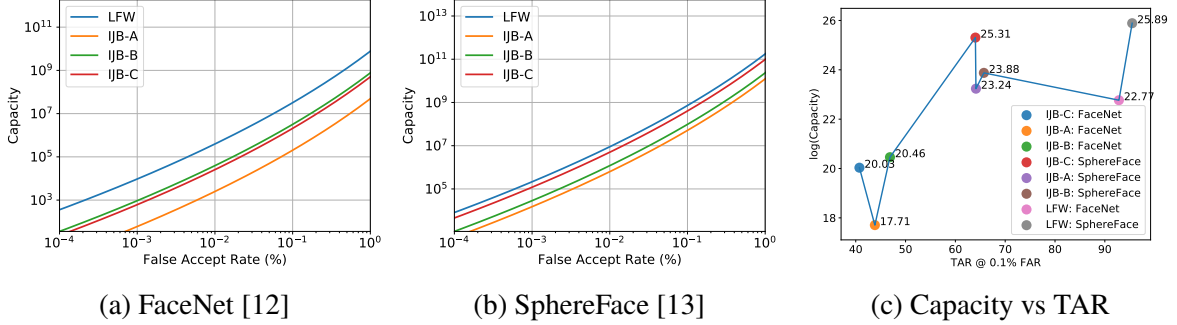


Figure 3.7 Capacity estimates across different datasets for the (a) FaceNet [12] and (b) SphereFace [13] representations as function of different false accept rates. Under the limit, the capacity tends to zero as the FAR tends to zero. Similarly, the capacity tends to  $\infty$  as the FAR tends to 1.0. (c) Logarithmic values of capacity on different datasets versus the corresponding TAR @ 0.1% FAR.

We extend the capacity estimates presented above to establish capacity as a function of different operating points, as defined by different false accept rates. We define  $r_{y_c}$  and  $r_{z_c}$  corresponding to the desired operating points and evaluate Eq. 3.9. In all our experiments we choose  $r_{y_c}$  to encompass 99% of the classes within the population hyper-ellipsoid. Different FARs define different decision boundary contours that, in turn, define the size of the class-specific hyper-ellipsoid. Figures 3.7a and 3.7b shows how the capacity of the representation changes as a function of the FARs for different datasets. We note that at the operating point of  $FAR = 0.1\%$ , the capacity of the maximum face representation is  $\approx 10^5$  in the constrained and  $\approx 10^3$  in the unconstrained case. However, at stricter operating points (FAR of 0.001% or 0.0001%), that is more meaningful at larger scales of operation [126], the capacity of the FaceNet representation is significantly lower (63 and 6, respectively for IJB-C) than the typical desired scale of operation of face recognition systems. These results suggest a significant room for improvement in face representation.

Table 3.4 IJB-C Capacity at 1% FAR Across Intra-Class Uncertainty

Model	Min	Mean	Median	Max
FaceNet	$1.6 \times 10^{14}$	$6 \times 10^8$	$5.0 \times 10^8$	$2.7 \times 10^4$
SphereFace	$4.9 \times 10^{16}$	$1.1 \times 10^{11}$	$9.8 \times 10^{10}$	$8.4 \times 10^4$

### 3.3.5 Ablation Studies

**DNN and PCA:** We seek to compare the capacity of classical PCA based EigenFaces [36] representation of image pixels and the DNN based representation. These are illustrative of the two extremes of various face representations proposed in the literature with FaceNet and SphereFace providing close to state-of-the-art recognition performance. The FaceNet and SphereFace representations are based on non-linear multi-layered deep convolutional network architectures. EigenFaces, in contrast, is a linear model for representing faces. The capacity of Eigenfaces is  $\approx 10^0$ , which is significantly lower than the capacity of DNN based representations. Eigenfaces, by virtue of being based on linear projections of the raw pixel values, is unable to scale beyond a handful of identities, while the DNN representations are able to resolve significantly more number of identities. The relative difference in the capacity is also reflected in the vast difference in the verification performance between the two representations.

**Data Bias:** Our capacity estimates are critically dependent on the representational support of the canonical class. In other words, the capacity expression in Eq. 3.9 depends on  $\Sigma_{z_c}$ , that is representative of the demographics and intra-class variability of the subjects in the population of interest. However, the hyper-ellipsoids corresponding to various classes could potentially be of a different size. For instance, in Fig. 3.1 each class-specific manifold is of different sizes, orientation and shape. Precisely defining or identifying a canonical subject, from among all possible identities, is in itself a challenging task and beyond the scope of this work. In Tab. 3.4 we report the capacity for different choices of classes (subjects) from the IJB-C dataset i.e., classes with the minimum, mean, median and maximum hyper-ellipsoid volume, thereby ranging from classes with very low intra-class variability and classes with very high intra-class variability. Datasets whose class distribution is

similar to the distribution of the data that was used to train the face representation, are expected to exhibit low intra-class uncertainty, while datasets with classes that are out of the training distribution can potentially have high intra-class uncertainty, and consequently lower capacity. Fig. 3.8 show examples of the images corresponding to the lowest and highest intra-class variability in each dataset.

Empirically, we observed that classes with the smallest hyper-ellipsoid are typically classes with very few images and very little variation in facial appearance. Similarly, classes with high intra-class uncertainty are typically classes with a very large number of images spanning a wide range of variations in pose, expression, illumination conditions etc., variations that one can expect under any real-world deployments of face recognition systems. Coupled with the fact that the capacity of the face representation is estimated from a very small sample of the population (less than 11,000 subjects), we argue that the class with large intra-class uncertainty within the datasets considered in this chapter is a reasonable proxy of a canonical subject in unconstrained real-world deployments of face recognition systems.

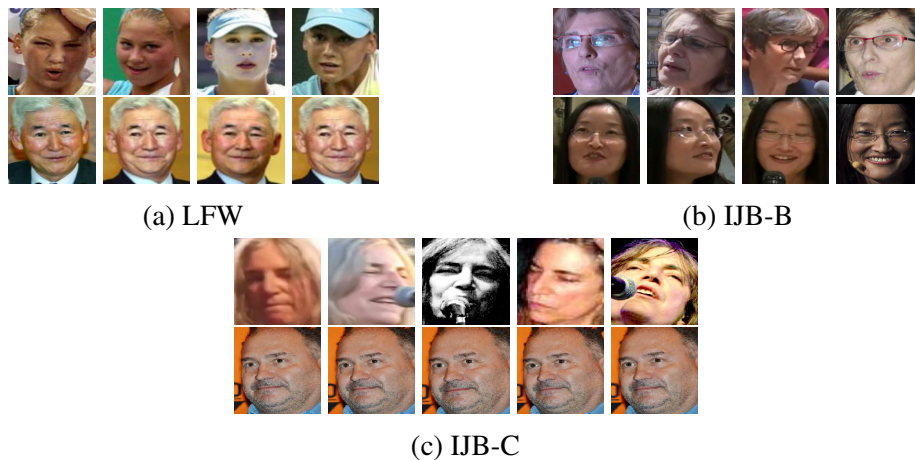


Figure 3.8 Example images of classes that correspond to different sizes of the class-specific hyper-ellipsoids, based on the SphereFace representation, for different datasets considered. *Top Row:* Images of the class with the largest class-specific hyper-ellipsoid for each database. Notice that in the case of a database with predominantly frontal faces (LFW), large variations in facial appearance lead to the greatest uncertainty in the class representation. On more challenging datasets (IJB-B, IJB-C), the face representation exhibits most uncertainty due to pose variations. *Bottom Row:* Images of the class with the smallest class-specific hyper-ellipsoid for each database. As expected, across all the datasets, frontal face images with the minimal change in appearance result in the least amount of uncertainty in the class representation.

Table 3.5 IJB-C Capacity at 1% FAR Across Manifold Support

Model	Hypersphere	Hyper-Ellipsoid (Axis-Aligned)	Hyper-Ellipsoid
FaceNet	$1.5 \times 10^3$	$9.2 \times 10^2$	$2.7 \times 10^4$
SphereFace	$6.7 \times 10^3$	$7.2 \times 10^3$	$8.4 \times 10^4$

**Gaussian Distribution Parameterization:** For the sake of efficiency we made the same modeling assumption for both the global shape of the embedding and the embedding shape of each class. The capacity estimates obtained thus far are by modeling the manifolds as unconstrained hyper-ellipsoids. We now obtain capacity estimates for different modeling assumptions on the shape of these entities. For instance the shapes could also be modeled as hyper-spheres corresponding to a diagonal covariance matrix with the same variance in each dimension. We generalize the hyper-sphere model to an axis aligned hyper-ellipsoid corresponding to a diagonal covariance matrix with possibly different variances along each dimension. Tab. 3.5 shows the capacity estimates on the IJB-C dataset at 1% FAR. We observe that the capacity estimates of the anisotropic Gaussian (hyper-ellipsoid) are two orders of magnitude higher than the capacity estimates of the reduced approximations, hyper-sphere (isotropic Gaussian) and axis-aligned hyper-ellipsoid. At the same time, the isotropic and the axis-aligned hyper-ellipsoid approximations result in very similar capacity estimates.

### 3.4 Conclusion

Face recognition is based on two underlying premises: persistence (invariance of face representation over time) and capacity (number of distinct identities a face representation can resolve). While face longitudinal studies [127] have addressed the persistence property, very little attention has been devoted to the capacity problem that is addressed here. The face representation process was modeled as a low-dimensional manifold embedded in high-dimensional space. We estimated the capacity of a face representation as a ratio of the volume of the population and class-specific manifolds as a function of the desired false acceptance rate. Empirically, we estimated the capacity of two deep neural network based face representations: FaceNet and SphereFace. Numerical results yielded a

capacity of  $10^5$  at a FAR of 1%. At lower FAR of 0.001%, the capacity dropped-off significantly to only 70 under unconstrained scenarios, impairing the scalability of the face representation. There does exist a large gap between the theoretical and empirical verification performance of the representations indicating that there is a significant scope for improvement in the discriminative capabilities of current state-of-the-art face representations.

As face recognition technology makes rapid strides in performance and witnesses wider adoption, quantifying the capacity of a given face representation is an important problem, both from an analytical as well as from a practical perspective. However, due to the challenging nature of finding a closed-form expression of the capacity, we make simplifying assumptions on the distribution of the population and specific classes in the representation space. Our experimental results demonstrate that even this simplified model is able to provide reasonable capacity estimates of a DNN based face representation. Relaxing the assumptions of the approach presented here is an exciting direction of future work, leading to more realistic capacity estimates.

# Chapter 4

## The Bias in Face Recognition

In this chapter we assess the demographic bias in FR algorithms and develop new methods to mitigate the demographic impact on FR performance. We experiment with different de-biasing approaches and network architectures using deep learning. Assessing the models' demographic bias quantitatively on various datasets we see how much bias mitigated in our attempt at improving fairness of face representations extracted from CNNs.

More specifically, in this chapter we propose two different methods to learn a fair face representation, where faces of every group could be equally well-represented. In the first method, we present a de-biasing adversarial network (DebFace) that learns to extract disentangled feature representations for both unbiased face recognition and demographics estimation. The proposed network consists of one identity classifier and three demographic classifiers (for gender, age, and race) that are trained to distinguish identity and demographic attributes, respectively. Adversarial learning is adopted to minimize correlation among feature factors so as to abate bias influence from other factors. We also design a scheme to combine demographics with identity features to strengthen robustness of face representation in different demographic groups.

The second method, group adaptive classifier (GAC), learns to mitigate bias by using adaptive convolution kernels and attention mechanisms on faces based on their demographic attributes. The adaptive module comprises kernel masks and channel-wise attention maps for each demographic

group so as to activate different facial regions for identification, leading to more discriminative features pertinent to their demographics. We also introduce an automated adaptation strategy which determines whether to apply adaptation to a certain layer by iteratively computing the dissimilarity among demographic-adaptive parameters, thereby increasing the efficiency of the adaptation learning.

The experimental results on benchmark face datasets (e.g., RFW [4], LFW, IJB-A, and IJB-C) show that our approach is able to reduce bias in face recognition on various demographic groups as well as maintain the competitive performance.

## 4.1 Fairness Learning and De-biasing Algorithms

We start by reviewing recent advances in fairness learning and de-biasing algorithms. Previous efforts on fairness techniques are proposed to prevent machine learning models from utilizing statistical bias in training data, including adversarial training [128–131], subgroup constraint optimization [132–134], data pre-processing (e.g., weighted sampling [135], and data transformation [136]), and algorithm post-processing [137, 138]. For example, in prior-DNN era, Zhang *et al.* [139] propose a cost-sensitive learning framework to reduce misclassification rate of face identification. To correct the skew of separating hyperplanes of SVM on imbalanced data, Liu *et al.* [140] propose Margin-Based Adaptive Fuzzy SVM that obtains a lower generalization error bound. In the DNN era, face recognition models are trained on large-scale face datasets with highly-imbalanced class distribution [141, 142]. To uncover deep learning bias, Alexander *et al.* [143] develop an algorithm to mitigate the hidden biases within training data. Range Loss [142] learns a robust face representation that makes the most use of every training sample. To mitigate the impact of insufficient class samples, center-based feature transfer learning [141] and large margin feature augmentation [144] are proposed to augment features of under-represented identities and equalize class distribution. Despite their effectiveness, these studies ignore the influence of demographic imbalance on the dataset, which may lead to demographic bias. The studies in [4, 145] address the demographic



bias in FR by leveraging unlabeled faces to improve the performance in groups with fewer samples. Wang *et al.* [5] propose skewness-aware reinforcement learning to mitigate racial bias in FR. Unlike prior work, we design a GAC framework to customize the classifier for each demographic group, which, if successful, would lead to mitigated bias. This framework is presented in the following Sec. 4.4.

Another promising approach learns a fair representation to preserve all discerning information about the data attributes or task-related attributes but eliminate the prejudicial effects from sensitive factors [22, 146–149]. Locatello *et al.* [150] show the feature disentanglement is consistently correlated with increasing fairness of general purpose representations by analyzing 12,600 SOTA models. Accordingly, we propose our second de-biasing framework, *DebFace*, which disentangles face representations to de-bias both FR and demographic attribute estimation. Sec. 4.3 discusses DebFace in more details.

## 4.2 Problem Definition

We now give a specific definition of the problem addressed in this chapter. The ultimate goal of unbiased face recognition is that, given a face recognition system, no statistically significant difference among the performance in different categories of face images. Despite the research on pose-invariant face recognition that aims for equal performance on all poses [151, 152], we believe that it is inappropriate to define variations like pose, illumination, or resolution, as the categories. These are instantaneous *image-related* variations with intrinsic bias. E.g., large-pose or low-resolution faces are inherently harder to be recognized than frontal-view high-resolution faces.

Rather, we would like to define *subject-related* properties such as demographic attributes as the categories. A face recognition system is **biased** if it performs worse on certain demographic cohorts. For practical applications, it is important to consider what demographic biases may exist, and whether these are intrinsic biases across demographic cohorts or algorithmic biases derived from the algorithm itself. This motivates us to analyze the demographic influence on face recognition

performance and strive to reduce algorithmic bias for face recognition systems. One may achieve this by training on a dataset containing uniform samples over the cohort space. However, the demographic distribution of a dataset is often imbalanced and underrepresents demographic minorities while overrepresenting majorities. Naively re-sampling a balanced training dataset may still induce bias since the diversity of latent variables is different across cohorts and the instances cannot be treated fairly during training. To mitigate demographic bias, we propose two face de-biasing frameworks that reduces demographic bias over face identity features while maintain the overall verification performance in the mean time.

### 4.3 Jointly De-biasing Face Recognition and Demographic Attribute Estimation

In this section, we introduce our another framework to address the influence of demographic bias on face recognition. With the technique of adversarial learning, we attack this issue from a different perspective. Specifically, we assume that if the face representation does not carry discriminative information of demographic attributes, it would be unbiased in terms of demographics. Given this assumption, one common way to remove demographic information from face representations is to perform feature disentanglement via adversarial learning (Fig. 4.1b). That is, the classifier of demographic attributes can be used to encourage the identity representation to *not* carry demographic information. However, one issue of this common approach is that, the demographic classifier itself could be biased (*e.g.*, the race classifier could be biased on gender), and hence it will act differently while disentangling faces of different cohorts. This is clearly undesirable as it leads to demographic biased identity representation.

To resolve the chicken-and-egg problem, we propose to *jointly* learn unbiased representations for both the identity and demographic attributes. Specifically, starting from a multi-task learning framework that learns disentangled feature representations of gender, age, race, and identity, respectively, we request the classifier of each task to act as adversarial supervision for the other tasks

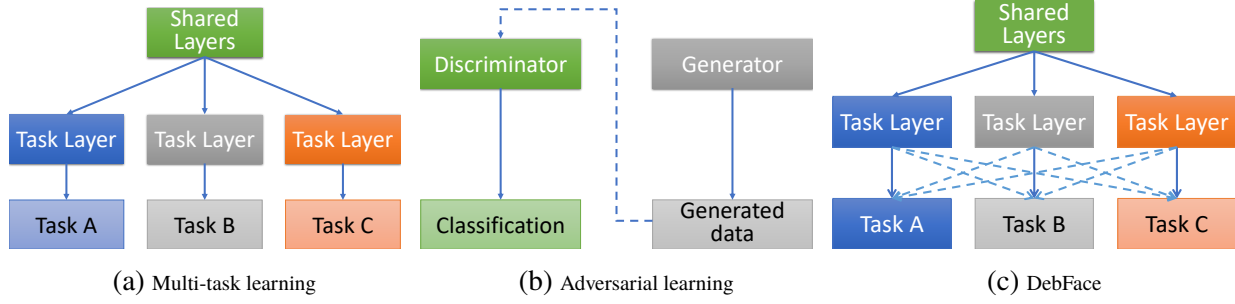


Figure 4.1 Methods to learn different tasks simultaneously. Solid lines are typical feature flow in CNN, while dash lines are adversarial losses.

(e.g., the dash arrows in Fig. 4.1c). These four classifiers help each other to achieve better feature disentanglement, resulting in unbiased feature representations for both the identity and demographic attributes. As shown in Fig. 4.1, our framework is in sharp contrast to either multi-task learning or adversarial learning.

Moreover, since the features are disentangled into the demographic and identity, our face representations also contribute to privacy-preserving applications. It is worth noticing that such identity representations contain little demographic information, which could undermine the recognition competence since demographic features are *part* of identity-related facial appearance. To retain the recognition accuracy on demographic biased face datasets, we propose another network that combines the demographic features with the demographic-free identity features to generate a new identity representation for face recognition.

The key contributions and findings of this work are:  $\diamond$  A thorough analysis of deep learning based face recognition performance on three different demographics: (i) gender, (ii) age, and (iii) race.

$\diamond$  A de-biasing face recognition framework, called DebFace, that generates disentangled representations for both identity and demographics recognition while jointly removing discriminative information from other counterparts.

$\diamond$  The identity representation from DebFace (DebFace-ID) shows lower bias on different demographic cohorts and also achieves SOTA face verification results on demographic-unbiased face recognition.

- ◊ The demographic attribute estimations via DebFace are less biased across other demographic cohorts.
- ◊ Combining ID with demographics results in more discriminative features for face recognition on biased datasets.

### 4.3.1 Adversarial Learning and Disentangled Representation

We first review previous work related to adversarial learning and representation disentanglement. Adversarial learning [153] has been well explored in many computer vision applications. For example, Generative Adversarial Networks (GANs) [154] employ adversarial learning to train a generator by competing with a discriminator that distinguishes real images from synthetic ones. Adversarial learning has also been applied to domain adaptation [155–158]. A problem of current interest is to learn interpretable representations with semantic meaning [159]. Many studies have been learning factors of variations in the data by supervised learning [152, 160–163], or semi-supervised/unsupervised learning [164–167], referred to as disentangled representation. For supervised disentangled feature learning, adversarial networks are utilized to extract features that only contain discriminative information of a target task. For face recognition, Liu *et al.* [161] propose a disentangled representation by training an adversarial autoencoder to extract features that can capture identity discrimination and its complementary knowledge. In contrast, our proposed DebFace differs from prior works in that each branch of a multi-task network acts as both a generator and discriminators of other branches (Fig. 4.1c).

### 4.3.2 Methodology

#### 4.3.2.1 Algorithm Design

The proposed network takes advantage of the relationship between demographics and face identities. On one hand, demographic characteristics are highly correlated to face features. On the other hand, demographic attributes are heterogeneous in terms of data type and semantics [168]. A male person,

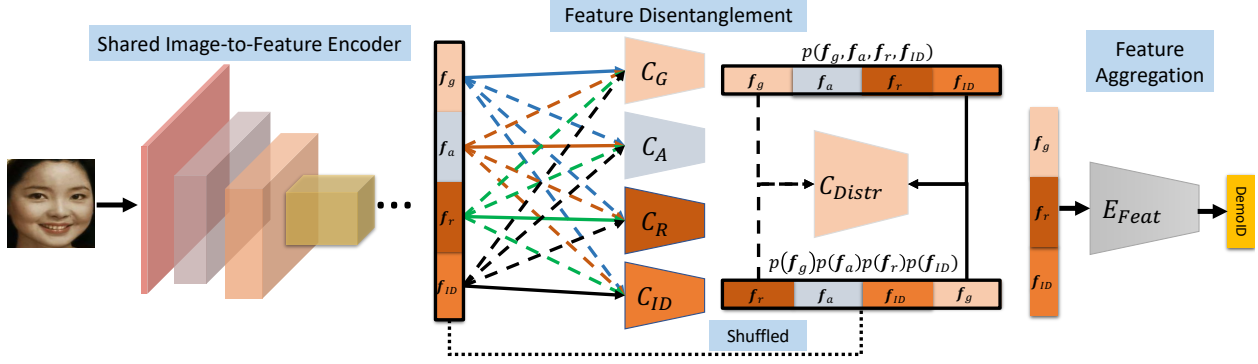


Figure 4.2 Overview of the proposed De-biasing face (DebFace) network. DebFace is composed of three major blocks, *i.e.*, a shared feature encoding block, a feature disentangling block, and a feature aggregation block. The solid arrows represent the forward inference, and the dashed arrows stand for adversarial training. During inference, either DebFace-ID (*i.e.*,  $\mathbf{f}_{ID}$ ) or DemoID can be used for face matching given the desired trade-off between biasness and accuracy.

for example, is not necessarily of a certain age or of a certain race. Accordingly, we present a framework that jointly generates demographic features and identity features from a single face image by considering both the aforementioned attribute correlation and attribute heterogeneity in a DNN.

While our main goal is to mitigate demographic bias from face representation, we observe that demographic estimations are biased as well (see Fig. 4.5). How can we remove the bias of face recognition when demographic estimations themselves are biased? Cook *et al.* [169] investigated this effect and found the performance of face recognition is affected by multiple demographic covariates. We propose a de-biasing network, DebFace, that disentangles the representation into gender (DebFace-G), age (DebFace-A), race (DebFace-R), and identity (DebFace-ID), to decrease bias of both face recognition and demographic estimations. Using adversarial learning, the proposed method is capable of jointly learning multiple discriminative representations while ensuring that each classifier cannot distinguish among classes through non-corresponding representations.

Though less biased, DebFace-ID loses demographic cues that are useful for identification. In particular, race and gender are two critical components that constitute face patterns. Hence, we desire to incorporate race and gender with DebFace-ID to obtain a more integrated face representation. We employ a light-weight fully-connected network to aggregate the representations into a face representation (DemoID) with the same dimensionality as DebFace-ID.

#### 4.3.2.2 Network Architecture

Fig. 4.2 gives an overview of the proposed DebFace network. It consists of four components: the shared image-to-feature encoder  $E_{Img}$ , the four attribute classifiers (including gender  $C_G$ , age  $C_A$ , race  $C_R$ , and identity  $C_{ID}$ ), the distribution classifier  $C_{Distr}$ , and the feature aggregation network  $E_{Feat}$ . We assume access to  $N$  labeled training samples  $\{(\mathbf{x}^{(i)}, y_g^{(i)}, y_a^{(i)}, y_r^{(i)}, y_{id}^{(i)})\}_{i=1}^N$ . Our approach takes an image  $\mathbf{x}^{(i)}$  as the input of  $E_{Img}$ . The encoder projects  $\mathbf{x}^{(i)}$  to its feature representation  $E_{Img}(\mathbf{x}^{(i)})$ . The feature representation is then decoupled into four  $D$ -dimensional feature vectors, gender  $\mathbf{f}_g^{(i)}$ , age  $\mathbf{f}_a^{(i)}$ , race  $\mathbf{f}_r^{(i)}$ , and identity  $\mathbf{f}_{ID}^{(i)}$ , respectively. Next, each attribute classifier operates the corresponding feature vector to correctly classify the target attribute by optimizing parameters of both  $E_{Img}$  and the respective classifier  $C_*$ . For a demographic attribute with  $K$  categories, the learning objective  $\mathcal{L}_{C_{Demo}}(\mathbf{x}, y_{Demo}; E_{Img}, C_{Demo})$  is the standard cross entropy loss function. For the  $n$ -identity classification, we adopt AM-Softmax [170] as the objective function  $\mathcal{L}_{C_{ID}}(\mathbf{x}, y_{id}; E_{Img}, C_{ID})$ . To de-bias all of the feature representations, adversarial loss  $\mathcal{L}_{Adv}(\mathbf{x}, y_{Demo}, y_{id}; E_{Img}, C_{Demo}, C_{ID})$  is applied to the above four classifiers such that each of them will not be able to predict correct labels when operating irrelevant feature vectors. Specifically, given a classifier, the remaining three attribute feature vectors are imposed on it and attempt to mislead the classifier by only optimizing the parameters of  $E_{Img}$ . To further improve the disentanglement, we also reduce the mutual information among the attribute features by introducing a distribution classifier  $C_{Distr}$ .  $C_{Distr}$  is trained to identify whether an input representation is sampled from the joint distribution  $p(\mathbf{f}_g, \mathbf{f}_a, \mathbf{f}_r, \mathbf{f}_{ID})$  or the multiplication of margin distributions  $p(\mathbf{f}_g)p(\mathbf{f}_a)p(\mathbf{f}_r)p(\mathbf{f}_{ID})$  via a binary cross entropy loss  $\mathcal{L}_{C_{Distr}}(\mathbf{x}, y_{Distr}; E_{Img}, C_{Distr})$ , where  $y_{Distr}$  is the distribution label. Similar to adversarial loss, a factorization objective function  $\mathcal{L}_{Fact}(\mathbf{x}, y_{Distr}; E_{Img}, C_{Distr})$  is utilized to restrain the  $C_{Distr}$  from distinguishing the real distribution and thus minimizes the mutual information of the four attribute representations. Both adversarial loss and factorization loss are detailed in Sec. 4.3.2.3. Altogether,

DebFace endeavors to minimize the joint loss:

$$\begin{aligned}
\mathcal{L}(\mathbf{x}, y_{Demo}, y_{id}, y_{Distr}; E_{Img}, C_{Demo}, C_{ID}, C_{Distr}) = & \\
& \mathcal{L}_{C_{Demo}}(\mathbf{x}, y_{Demo}; E_{Img}, C_{Demo}) \\
& + \mathcal{L}_{C_{ID}}(\mathbf{x}, y_{id}; E_{Img}, C_{ID}) \\
& + \mathcal{L}_{C_{Distr}}(\mathbf{x}, y_{Distr}; E_{Img}, C_{Distr}) \\
& + \lambda \mathcal{L}_{Adv}(\mathbf{x}, y_{Demo}, y_{id}; E_{Img}, C_{Demo}, C_{ID}) \\
& + \nu \mathcal{L}_{Fact}(\mathbf{x}, y_{Distr}; E_{Img}, C_{Distr}),
\end{aligned} \tag{4.1}$$

where  $\lambda$  and  $\nu$  are hyper-parameters determining how much the representation is decomposed and decorrelated in each training iteration.

The discriminative demographic features in DebFace-ID are weakened by removing demographic information. Fortunately, our de-biasing network preserves all pertinent demographic features in a disentangled way. Basically, we train another multilayer perceptron (MLP)  $E_{Feat}$  to aggregate DebFace-ID and the demographic embeddings into a unified face representation DemoID. Since age generally does not pertain to a person’s identity, we only consider gender and race as the identity-informative attributes. The aggregated embedding,  $\mathbf{f}_{DemoID} = E_{Feat}(\mathbf{f}_{ID}, \mathbf{f}_g, \mathbf{f}_r)$ , is supervised by an identity-based triplet loss:

$$\mathcal{L}_{E_{Feat}} = \frac{1}{M} \sum_{i=1}^M [\|\mathbf{f}_{DemoID^a}^{(i)} - \mathbf{f}_{DemoID^p}^{(i)}\|_2^2 - \|\mathbf{f}_{DemoID^a}^{(i)} - \mathbf{f}_{DemoID^n}^{(i)}\|_2^2 + \alpha]_+, \tag{4.2}$$

where  $\{\mathbf{f}_{DemoID^a}^{(i)}, \mathbf{f}_{DemoID^p}^{(i)}, \mathbf{f}_{DemoID^n}^{(i)}\}$  is the  $i^{th}$  triplet consisting of an anchor, a positive, and a negative DemoID representation,  $M$  is the number of hard triplets in a mini-batch.  $[x]_+ = \max(0, x)$ , and  $\alpha$  is the margin.

#### 4.3.2.3 Adversarial Training and Disentanglement

As discussed in Sec. 4.3.2.2, the adversarial loss aims to minimize the task-independent information semantically, while the factorization loss strives to dwindle the interfering information statistically.

We employ both losses to disentangle the representation extracted by  $E_{Img}$ . We introduce the adversarial loss as a means to learn a representation that is invariant in terms of certain attributes, where a classifier trained on it cannot correctly classify those attributes using that representation. We take one of the attributes, *e.g.*, gender, as an example to illustrate the adversarial objective. First of all, for a demographic representation  $\mathbf{f}_{Demo}$ , we learn a gender classifier on  $\mathbf{f}_{Demo}$  by optimizing the classification loss  $\mathcal{L}_{C_G}(\mathbf{x}, y_{Demo}; E_{Img}, C_G)$ . Secondly, for the same gender classifier, we intend to maximize the chaos of the predicted distribution [171]. It is well known that a uniform distribution has the highest entropy and presents the most randomness. Hence, we train the classifier to predict the probability distribution as close as possible to a uniform distribution over the category space by minimizing the cross entropy:

$$\mathcal{L}_{Adv}^G(\mathbf{x}, y_{Demo}, y_{id}; E_{Img}, C_G) = - \sum_{k=1}^{K_G} \frac{1}{K_G} \cdot \left( \log \frac{e^{C_G(\mathbf{f}_{Demo})_k}}{\sum_{j=1}^{K_G} e^{C_G(\mathbf{f}_{Demo})_j}} + \log \frac{e^{C_G(\mathbf{f}_{ID})_k}}{\sum_{j=1}^{K_G} e^{C_G(\mathbf{f}_{ID})_j}} \right), \quad (4.3)$$

where  $K_G$  is the number of categories in gender<sup>1</sup>, and the ground-truth label is no longer an one-hot vector, but a  $K_G$ -dimensional vector with all elements being  $\frac{1}{K_G}$ . The above loss function corresponds to the dash lines in Fig. 4.2. It strives for gender-invariance by finding a representation that makes the gender classifier  $C_G$  perform poorly. We minimize the adversarial loss by only updating parameters in  $E_{Img}$ .

We further decorrelate the representations by reducing the mutual information across attributes. By definition, the mutual information is the relative entropy (KL divergence) between the joint distribution and the product distribution. To increase uncorrelation, we add a distribution classifier  $C_{Distr}$  that is trained to simply perform a binary classification using  $\mathcal{L}_{C_{Distr}}(\mathbf{x}, y_{Distr}; E_{Img}, C_{Distr})$  on samples  $\mathbf{f}_{Distr}$  from both the joint distribution and dot product distribution. Similar to adversarial learning, we factorize the representations by tricking the classifier via the same samples so that the predictions are close to random guesses,

$$\mathcal{L}_{Fact}(\mathbf{x}, y_{Distr}; E_{Img}, C_{Distr}) = - \sum_{i=1}^2 \frac{1}{2} \log \frac{e^{C_{Distr}(\mathbf{f}_{Distr})_i}}{\sum_{j=1}^2 e^{C_{Distr}(\mathbf{f}_{Distr})_j}}. \quad (4.4)$$

In each mini-batch, we consider  $E_{Img}(\mathbf{x})$  as samples of the joint distribution  $p(\mathbf{f}_g, \mathbf{f}_a, \mathbf{f}_r, \mathbf{f}_{ID})$ . We

---

<sup>1</sup>In our case,  $K_G = 2$ , *i.e.*, male and female.



Table 4.1 Statistics of training and testing datasets used in the paper.

Dataset	# of Images	# of Subjects	Contains the label of			
			Gender	Age	Race	ID
CACD [172]	163,446	2,000	No	Yes	No	Yes
IMDB [173]	460,723	20,284	Yes	Yes	No	Yes
UTKFace [174]	24,106	-	Yes	Yes	Yes	No
AgeDB [175]	16,488	567	Yes	Yes	No	Yes
AFAD [176]	165,515	-	Yes	Yes	Yes <sup>a</sup>	No
AAF [177]	13,322	13,322	Yes	Yes	No	Yes
FG-NET <sup>2</sup>	1,002	82	No	Yes	No	Yes
RFW [4]	665,807	-	No	No	Yes	Partial
BUPT-Balancedface [5]	1,251,430	28,000	No	No	Yes	Yes
IMFDB-CVIT [178]	34,512	100	Yes	Age Groups	Yes <sup>*</sup>	Yes
Asian-DeepGlint [179]	2,830,146	93,979	No	No	Yes <sup>a</sup>	Yes
MS-Celeb-1M [23]	5,822,653	85,742	No	No	No	Yes
PCSO [180]	1,447,607	5,749	Yes	Yes	Yes	Yes
LFW [27]	13,233	5,749	No	No	No	Yes
IJB-A [29]	25,813	500	Yes	Yes	Skin Tone	Yes
IJB-C [9]	31,334	3,531	Yes	Yes	Skin Tone	Yes

<sup>a</sup> East Asian<sup>\*</sup> Indian

randomly shuffle feature vectors of each attribute, and re-concatenate them into  $4D$ -dimension, which are approximated as samples of the product distribution  $p(\mathbf{f}_g)p(\mathbf{f}_a)p(\mathbf{f}_r)p(\mathbf{f}_{ID})$ . During factorization, we only update  $E_{Img}$  to minimize mutual information between decomposed features.

### 4.3.3 Experiments

#### 4.3.3.1 Datasets and Pre-processing

We utilize 15 total face datasets in this work, for learning the demographic estimation models, the baseline face recognition model, DebFace model as well as their evaluation. To be specific, CACD [172], IMDB [173], UTKFace [174], AgeDB [175], AFAD [176], AAF [177], FG-NET [181], RFW [4], IMFDB-CVIT [178], Asian-DeepGlint [179], and PCSO [180] are the datasets for training and testing demographic estimation models; and MS-Celeb-1M [23], LFW [27], IJB-A [29], and IJB-C [9] are for learning and evaluating face verification models. Tab. 4.1 reports the statistics of training and testing datasets involved in all the experiments of both GAC and DebFace, including the total number of face images, the total number of subjects (identities), and whether the dataset contains the annotation of gender, age, race, or identity (ID). All faces are detected by MTCNN [33]. Each face image is cropped and resized to  $112 \times 112$  pixels using a similarity transformation based

on the detected landmarks.

#### 4.3.3.2 Implementation Details

DebFace is trained on a cleaned version of MS-Celeb-1M [6], using the ArcFace architecture [6] with 50 layers for the encoder  $E_{Img}$ . Since there are no demographic labels in MS-Celeb-1M, we first train three demographic attribute estimation models for gender, age, and race, respectively. For age estimation, the model is trained on the combination of CACD, IMDB, UTKFace, AgeDB, AFAD, and AAF datasets. The gender estimation model is trained on the same datasets except CACD which contains no gender labels. We combine AFAD, RFW, IMFDB-CVIT, and PCSO for race estimation training. All three models use ResNet [50] with 34 layers for age, 18 layers for gender and race. We discuss the evaluation results of the demographic attribute estimation models in Sec. 4.5.

We predict the demographic labels of MS-Celeb-1M with the well-trained demographic models. Our DebFace is then trained on the re-labeled MS-Celeb-1M using SGD with a momentum of 0.9, a weight decay of 0.01, and a batch size of 256. The learning rate starts from 0.1 and drops to 0.0001 following the schedule at 8, 13, and 15 epochs. The dimensionality of the embedding layer of  $E_{Img}$  is  $4 \times 512$ , *i.e.*, each attribute representation (gender, age, race, ID) is a 512-*dim* vector. We keep the hyper-parameter setting of AM-Softmax as [6]:  $s = 64$  and  $m = 0.5$ . The feature aggregation network  $E_{Feat}$  comprises of two linear residual units with P-ReLU and BatchNorm in between.  $E_{Feat}$  is trained on MS-Celeb-1M by SGD with a learning rate of 0.01. The triplet loss margin  $\alpha$  is 1.0. The disentangled features of gender, race, and identity are concatenated into a  $3 \times 512$ -*dim* vector, which inputs to  $E_{Feat}$ . The network is then trained to output a 512-*dim* representation for face recognition on biased datasets.

#### 4.3.3.3 De-biasing Face Verification

**Baseline:** We compare DebFace-ID with a regular face representation model which has the same architecture as the shared feature encoder of DebFace. Referred to as BaseFace, this baseline model

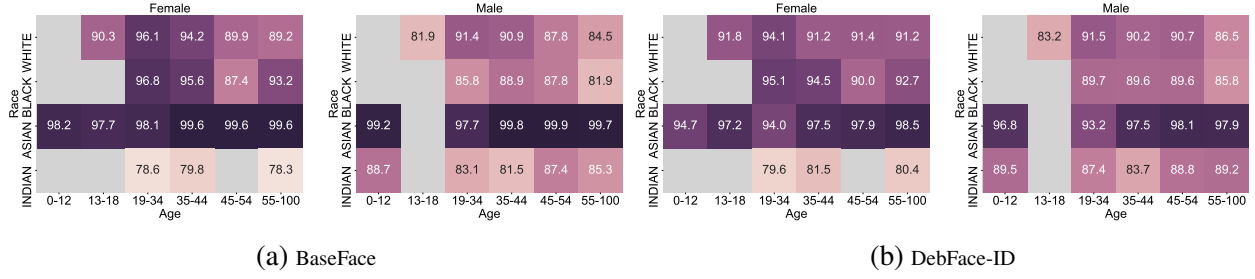


Figure 4.3 Face Verification AUC (%) on each demographic cohort. The cohorts are chosen based on the three attributes, *i.e.*, gender, age, and race. To fit the results into a 2D plot, we show the performance of male and female separately. Due to the limited number of face images in some cohorts, their results are gray cells.

is also trained on MS-Celeb-1M, with the representation dimension of 512.

To show the efficacy of DebFace-ID on bias mitigation, we evaluate the verification performance of DebFace-ID and BaseFace on faces from each demographic cohort. There are 48 total cohorts given the combination of demographic attributes including 2 gender (male, female), 4 race<sup>3</sup> (Black, White, East Asian, Indian), and 6 age group (0 – 12, 13 – 18, 19 – 34, 35 – 44, 45 – 54, 55 – 100). We combine CACD, AgeDB, CVIT, and a subset of Asian-DeepGlint as the testing set. Overlapping identities among these datasets are removed. IMDB is excluded from the testing set due to its massive number of wrong ID labels. For the dataset without certain demographic labels, we simply use the corresponding models to predict the labels. We report the Area Under the Curve (AUC) of the Receiver Operating Characteristics (ROC). We define the degree of bias, termed *biasness*, as the standard deviation of performance across cohorts.

Fig. 4.3 shows the face verification results of BaseFace and DebFace-ID on each cohort. That is, for a particular face representation (*e.g.*, DebFace-ID), we report its AUC on each cohort by putting the number in the corresponding cell. From these heatmaps, we observe that both DebFace-ID and BaseFace present bias in face verification, where the performance on some cohorts are significantly worse, especially the cohorts of Indian female and elderly people. Compared to BaseFace, DebFace-ID suggests less bias and the difference of AUC is smaller, where the heatmap exhibits smoother edges. Fig. 4.4 shows the performance of face verification on 12 demographic cohorts. Both

<sup>3</sup>To clarify, we consider two race groups, Black and White; and two ethnicity groups, East Asian and Indian. The word race denotes both race and ethnicity here.

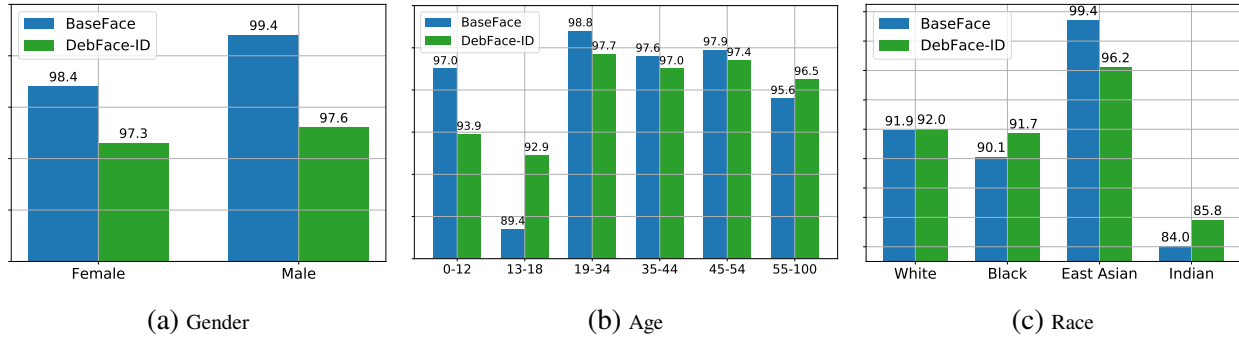


Figure 4.4 The overall performance of face verification AUC (%) on gender, age, and race.

DebFace-ID and BaseFace present similar relative accuracies across cohorts. For example, both algorithms perform worse on the younger age cohorts than on adults; and the performance on the Indian is significantly lower than on the other races. DebFace-ID decreases the bias by gaining discriminative face features for cohorts with less images in spite of the reduction in the performance on cohorts with more samples.

#### 4.3.3.4 De-biasing Demographic Attribute Estimation

**Baseline:** We further explore the bias of demographic attribute estimation and compare demographic attribute classifiers of DebFace with baseline estimation models. We train three demographic estimation models, namely, gender estimation (BaseGender), age estimation (BaseAge), and race estimation (BaseRace), on the same training set as DebFace. For fairness, all three models have the same architecture as the shared layers of DebFace.

We combine the four datasets mentioned in Sec. 4.3.3.3 with IMDB as the global testing set. As all demographic estimations are treated as classification problems, the classification accuracy is used as the performance metric. As shown in Fig. 4.5, all demographic attribute estimations present significant bias. For gender estimation, both algorithms perform worse on the White and Black cohorts than on East Asian and Indian. In addition, the performance on young children is significantly worse than on adults. In general, the race estimation models perform better on the male cohort than on female. Compared to gender, race estimation shows higher bias in terms of age. Both baseline methods and DebFace perform worse on cohorts in age between 13 to 44 than in other age

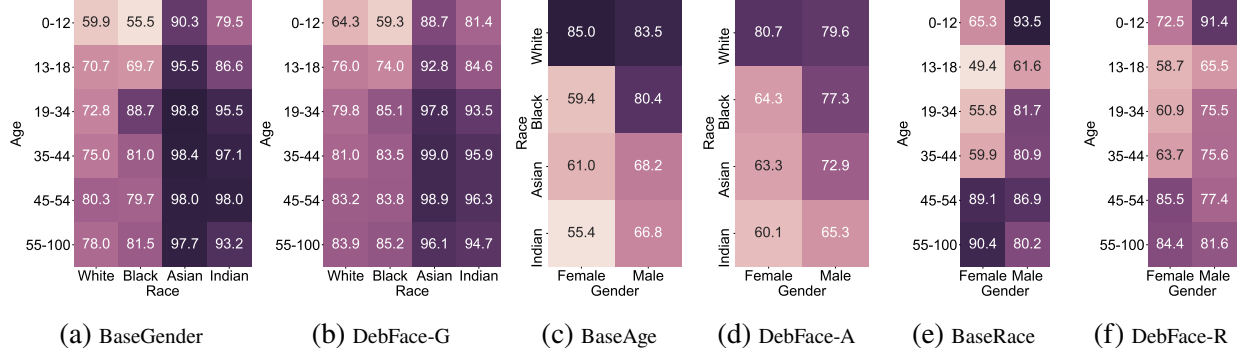


Figure 4.5 Classification accuracy (%) of demographic attribute estimations on faces of different cohorts, by DebFace and the baselines. For simplicity, we use DebFace-G, DebFace-A, and DebFace-R to represent the gender, age, and race classifier of DebFace.

Table 4.2 Biasness of Face Recognition and Demographic Attribute Estimation.

Method	Face Verification				Demographic Estimation		
	All	Gender	Age	Race	Gender	Age	Race
Baseline	6.83	0.50	3.13	5.49	12.38	10.83	14.58
DebFace	<b>5.07</b>	<b>0.15</b>	<b>1.83</b>	<b>3.70</b>	<b>10.22</b>	<b>7.61</b>	<b>10.00</b>

groups.

Similar to race, age estimation still achieves better performance on male than on female. Moreover, the white cohort shows dominant advantages over other races in age estimation. In spite of the existing bias in demographic attribute estimations, the proposed DebFace is still able to mitigate bias derived from algorithms. Compared to Fig. 4.5a, 4.5e, 4.5c, cells in Fig. 4.5b, 4.5f, 4.5d present more uniform colors. We summarize the biasness of DebFace and baseline models for both face recognition and demographic attribute estimations in Tab. 4.2. In general, we observe DebFace substantially reduces biasness for both tasks. For the task with larger biasness, the reduction of biasness is larger.

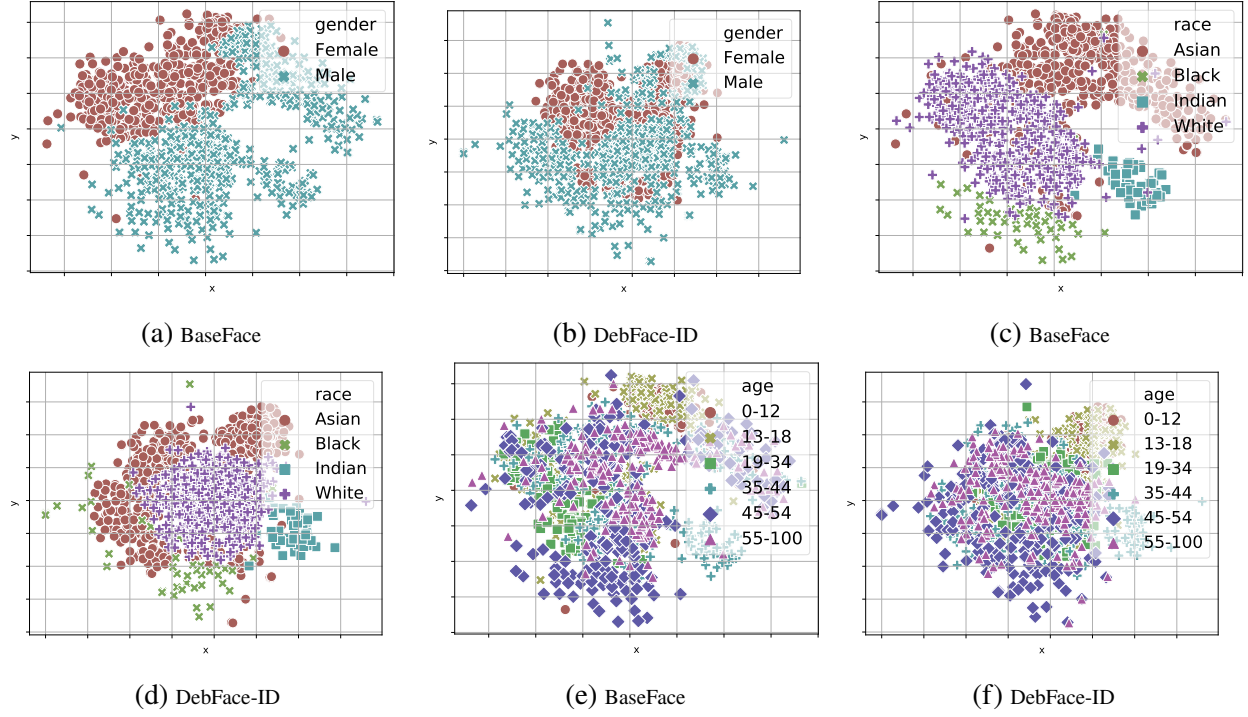


Figure 4.6 The distribution of face identity representations of BaseFace and DebFace. Both collections of feature vectors are extracted from images of the same dataset. Different colors and shapes represent different demographic attributes. Zoom in for details.

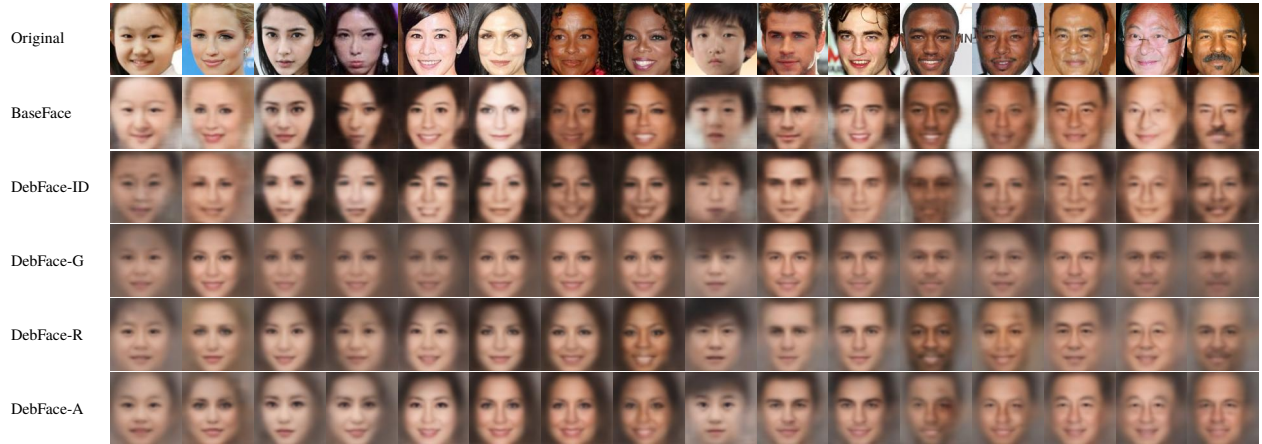


Figure 4.7 Reconstructed Images using Face and Demographic Representations. The first row is the original face images. From the second row to the bottom, the face images are reconstructed from 2) BaseFace; 3) DebFace-ID; 4) DebFace-G; 5) DebFace-R; 6) DebFace-A. Zoom in for details.

#### 4.3.3.5 Analysis of Disentanglement

We notice that DebFace still suffers unequal performance in different demographic groups. It is because there are other latent variables besides the demographics, such as image quality or

capture conditions that could lead to biased performance. Such variables are difficult to control in pre-collected large face datasets. In the framework of DebFace, it is also related to the degree of feature disentanglement. A fully disentangling is supposed to completely remove the factors of bias from demographic information. To illustrate the feature disentanglement of DebFace, we show the demographic discriminative ability of face representations by using these features to estimate gender, age, and race. Specifically, we first extract identity features of images from the testing set in Sec. 4.3.3.1 and split them into training and testing sets. Given demographic labels, the face features are fed into a two-layer fully-connected network, learning to classify one of the demographic attributes. Tab. 4.3 reports the demographic classification accuracy on the testing set. For all three demographic estimations, DebFace-ID presents much lower accuracies than BaseFace, indicating the decline of demographic information in DebFace-ID. We also plot the distribution of identity representations in the feature space of BaseFace and DebFace-ID. From the testing set in Sec. 4.3.3.3, we randomly select 50 subjects in each demographic group and one image of each subject. BaseFace and DebFace-ID are extracted from the selected image set and are then projected from 512-*dim* to 2-*dim* by T-SNE. Fig. 4.6 shows their T-SNE feature distributions. We observe that BaseFace presents clear demographic clusters, while the demographic clusters of DebFace-ID, as a result of disentanglement, mostly overlap with each other.

To visualize the disentangled feature representations of DebFace, we train a decoder that reconstructs face images from the representations. Four face decoders are trained separately for each disentangled component, *i.e.*, gender, age, race, and ID. In addition, we train another decoder to reconstruct faces from BaseFace for comparison. As shown in Fig. 4.7, both BaseFace and DebFace-ID maintain the identify features of the original faces, while DebFace-ID presents less demographic characteristics. No race or age, but gender features can be observed on faces reconstructed from DebFace-G. Meanwhile, we can still recognize race and age attributes on faces generated from DebFace-R and DebFace-A.

Table 4.3 Demographic Classification Accuracy (%) by face features.

Method	Gender	Race	Age
BaseFace	95.27	89.82	78.14
DebFace-ID	73.36	61.79	49.91

Table 4.4 Face Verification Accuracy (%) on RFW dataset.

Method	Gender	Race	Age
BaseFace	95.27	89.82	78.14
DebFace-ID	73.36	61.79	49.91

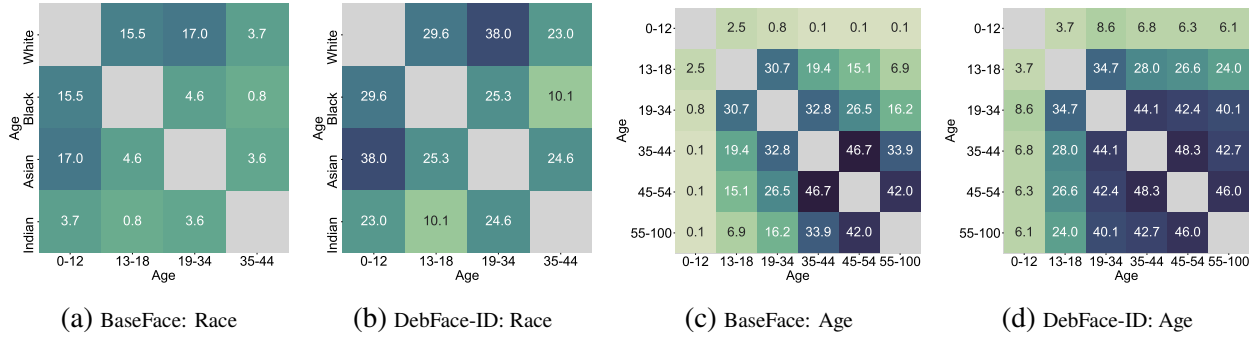


Figure 4.8 The percentage of false accepted cross race or age pairs at 1% FAR.

#### 4.3.3.6 Face Verification on Public Testing Datasets

We report the performance of three different settings, using 1) BaseFace, the same baseline in Sec. 4.3.3.3, 2) DebFace-ID, and 3) the fused representation DemoID. Table 4.5 reports face verification results on on three public benchmarks: LFW, IJB-A, and IJB-C. On LFW, DemoID outperforms BaseFace while maintaining similar accuracy compared to SOTA algorithms. On IJB-A/C, DemoID outperforms all prior works except PFE [184]. Although DebFace-ID shows lower discrimination, TAR at lower FAR on IJB-C is higher than that of BaseFace. To evaluate DebFace on a racially balanced testing dataset RFW [4] and compare with the work [5], we train a DebFace model on BUPT-Balancedface [5] dataset. The new model is trained to reduce racial bias by disentangling ID and race. Tab. 4.4 reports the verification results on RFW. While DebFace-ID gives a slightly lower face verification accuracy, it improves the biasness over [5].

We observe that DebFace-ID is less discriminative than BaseFace, or DemoID, since demographics are essential components of face features. To understand the deterioration of DebFace, we



Table 4.5 Verification Performance on LFW, IJB-A, and IJB-C.

Method	LFW (%)	Method	IJB-A (%) 0.1% FAR	IJB-C @ FAR (%)		
				0.001%	0.01%	0.1%
DeepFace+ [17]	97.35	Yin <i>et al.</i> [182]	$73.9 \pm 4.2$	-	-	69.3
CosFace [19]	99.73	Cao <i>et al.</i> [59]	$90.4 \pm 1.4$	74.7	84.0	91.0
ArcFace [6]	99.83	Multicolumn [183]	$92.0 \pm 1.3$	77.1	86.2	92.7
PFE [184]	99.82	PFE [184]	$95.3 \pm 0.9$	89.6	93.3	95.5
<i>BaseFace</i>	99.38	<i>BaseFace</i>	$90.2 \pm 1.1$	80.2	88.0	92.9
<i>DebFace-ID</i>	98.97	<i>DebFace-ID</i>	$87.6 \pm 0.9$	82.0	88.1	89.5
<i>DemoID</i>	99.50	<i>DemoID</i>	$92.2 \pm 0.8$	83.2	89.4	92.9

analyse the effect of demographic heterogeneity on face verification by showing the tendency for one demographic group to experience a false accept error relative to another group. For any two demographic cohorts, we check the number of falsely accepted pairs that are from different groups at 1% FAR. Fig. 4.8 shows the percentage of such falsely accepted demographic-heterogeneous pairs. Compared to BaseFace, DebFace exhibits more cross-demographic pairs that are falsely accepted, resulting in the performance decline on demographically biased datasets. Due to the demographic information reduction, DebFace-ID is more susceptible to errors between demographic groups. In the sense of de-biasing, it is preferable to decouple demographic information from identity features. However, if we prefer to maintain the overall performance across all demographics, we can still aggregate all the relevant information. It is an application-dependent trade-off between accuracy and de-biasing. DebFace balances the accuracy vs. bias trade-off by generating both debiased identity and debiased demographic representations, which may be aggregated into DemoID if bias is less of a concern.

#### 4.3.3.7 Distributions of Scores

We follow the work of [21] that investigates the effect of demographic homogeneity and heterogeneity on face recognition. We first randomly select images from CACD, AgeDB, CVIT, and Asian-DeepGlint datasets, and extract the corresponding feature vectors by using the models of BaseFace and DebFace, respectively. Given their demographic attributes, we put those images into separate groups depending on whether their gender, age, and race are the same or different. For each group, a

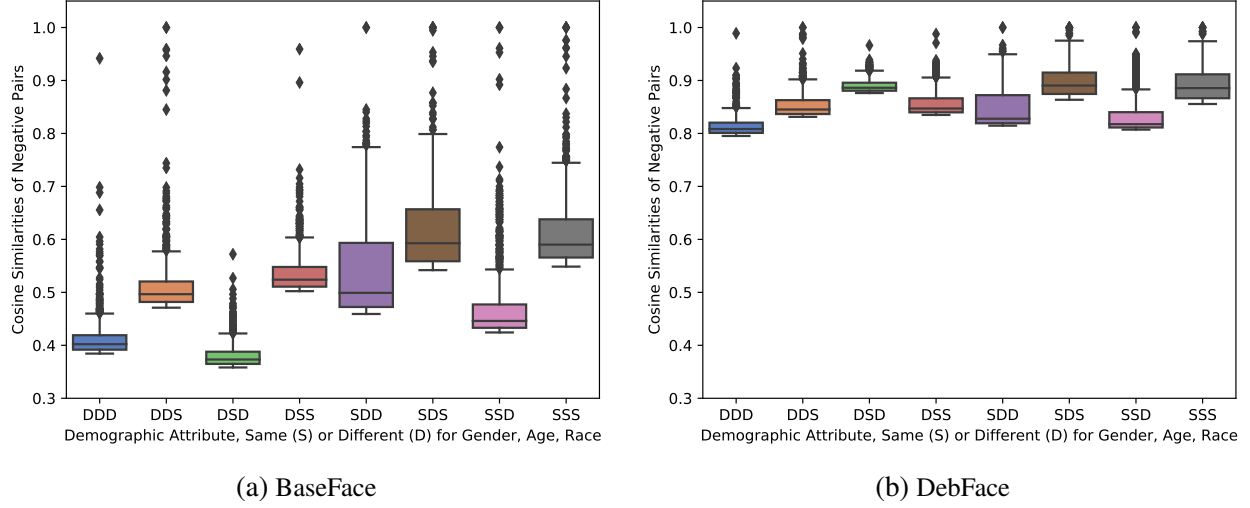


Figure 4.9 BaseFace and DebFace distributions of the similarity scores of the imposter pairs across homogeneous versus heterogeneous gender, age, and race categories.

fixed false alarm rate (the percentage of the face pairs from the same subjects being falsely verified as from different subjects) is set to 1%. Among the falsely verified pairs, we plot the top 10<sup>th</sup> percentile scores of the negative face pairs (a pair of face images that are from different subjects) given their demographic attributes. As shown in Fig. 4.9a and Fig. 4.9b, we observe that the similarities of DebFace are higher than those of BaseFace. One of the possible reasons is that the demographic information is disentangled from the identity features of DebFace, increasing the overall pair-wise similarities between faces of different identities. In terms of de-biasing, DebFace also reflects smaller differences of the score distribution with respect to the homogeneity and heterogeneity of demographics.

## 4.4 Mitigating Face Recognition Bias via Group Adaptive Classifier

In spite the effectiveness of DebFace in mitigating demographic bias, it degenerates the overall recognition performance as well. This motivates us to find another solution to this problem such that the biasness can be reduced without impairing the average recognition performance. In this section,

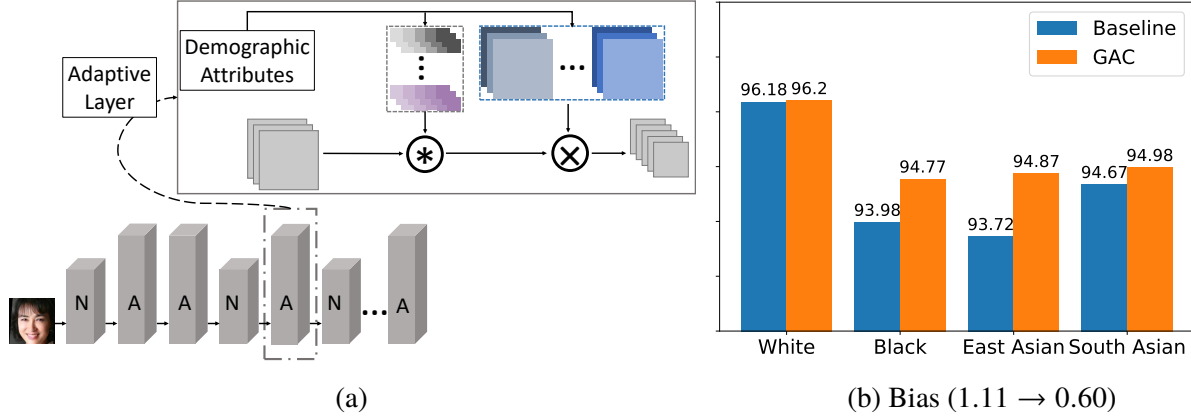


Figure 4.10 (a) Our proposed group adaptive classifier (GAC) automatically chooses between non-adaptive (“N”) and adaptive (“A”) layer in a multi-layer network, where the latter uses demographic-group-specific kernel and attention. (b) Compared to the baseline with the 50-layer ArcFace backbone, GAC improves face verification accuracy in most groups of RFW dataset [4], especially under-represented groups, leading to mitigated FR bias. GAC reduces biasness from 1.11 to 0.60.

we introduce our second approach to mitigating face recognition bias via group adaptive classifier (GAC). The main idea of GAC is to optimize the face representation learning on every demographic group in a single network, despite demographically imbalanced training data. Conceptually, we may categorize face features into two types of patterns: *general pattern* is shared by all faces; *differential pattern* is relevant to demographic attributes. When the differential pattern of one specific demographic group dominates training data, the network learns to predict identities mainly based on that pattern as it is more convenient to minimize the loss than using other patterns, thus bringing it bias towards faces of that specific group. One mitigation is to give the network more capacity to broaden its scope for multiple face patterns from different demographic groups. An unbiased FR model shall rely on not only unique patterns for recognition of different groups, but also general patterns of all faces for improved generalizability. Accordingly, as in Fig. 4.10, we propose GAC to explicitly learn these different feature patterns. GAC includes two modules: the adaptive layer and automation module. The adaptive layer in GAC comprises adaptive convolution kernels and channel-wise attention maps where each kernel and attention map tackle faces in *one* demographic group. We also introduce a new objective function to GAC, which diminishes the variation of average intra-class distance between demographic groups.

Prior work on dynamic CNNs introduce adaptive convolutions to either every layer [185–187], or manually specified layers [188–190]. In contrast, this work proposes an automation module to choose which layers to apply adaptations. As we observed, not all convolutional layers require adaptive kernels for bias mitigation (see Fig. 4.15a). At any layer of GAC, only kernels expressing high dissimilarity are considered as demographic-adaptive kernels. For those with low dissimilarity, their average kernel is shared by all input images in that layer. Thus, the proposed network progressively learns to select the optimal structure for the demographic-adaptive learning. This enables that both non-adaptive layers with shared kernels and adaptive layers are jointly learned in a unified network.

Contributions of this work are summarised as: 1) A new face recognition algorithm that reduces demographic bias and increases robustness of representations for faces in every demographic group by adopting adaptive convolutions and attention techniques; 2) A new adaptation mechanism that automatically determines the layers to employ dynamic kernels and attention maps; 3) The proposed method achieves SOTA performance on a demographic-balanced dataset and three benchmarks.

#### **4.4.1 Adaptive Neural Networks**

Since the main technique applied by GAC is adaptive neural network, we first review recent work related to adaptive learning. Three types of CNN-based adaptive learning techniques are related to our work: adaptive architectures, adaptive kernels, and attention mechanism. Adaptive architectures design new performance-based neural functions or structures, *e.g.*, neuron selection hidden layers [191] and automatic CNN expansion for FR [192]. As CNN advances many AI fields, prior works propose dynamic kernels to realize content-adaptive convolutions. Li *et al.* [193] propose a shape-driven kernel for facial trait recognition where each landmark-centered patch has a unique kernel. A convolution fusion for graph neural networks is introduced by [194] where a set of varying-size filters are used per layer. The works of [195] and [196] use a kernel selection scheme to automatically adjust the receptive field size based on inputs. To better suit input data, [197] splits training data into clusters and learns an exclusive kernel per cluster. Li *et al.* [198] introduce an adaptive CNN for object detection that transfers pre-trained CNNs to a target domain by selecting

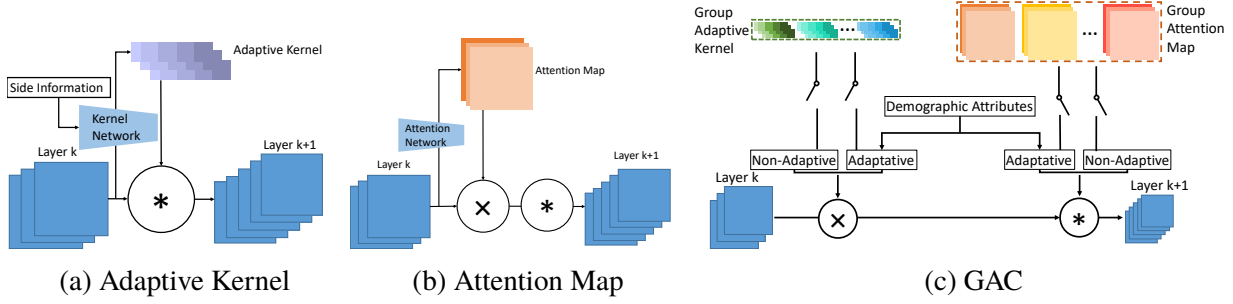


Figure 4.11 A comparison of approaches in adaptive CNNs.

useful kernels per layer. Alternatively, one may feed input images or features into a kernel function to dynamically generate convolution kernels [199–202]. Despite its effectiveness, such individual adaptation may not be suitable given the diversity of faces in demographic groups. Our work is most related to the side information adaptive convolution [185], where in each layer a sub-network inputs auxiliary information to generate filter weights. We mainly differ in that GAC automatically learns where to use adaptive kernels in a multi-layer CNN (see Figs. 4.11a and 4.11c), thus more efficient and capable in applying to a deeper CNN.

As the human perception process naturally selects the most pertinent piece of information, attention mechanisms are designed for a variety of tasks, *e.g.*, detection [203], recognition [204], image captioning [205], tracking [206], pose estimation [190], and segmentation [188]. Typically, attention weights are estimated by feeding images or feature maps into a shared network, composed of convolutional and pooling layers [204, 207–209] or multi-layer perceptron (MLP) [210–213]. Apart from feature-based attention, Hou *et al.* [189] propose a correlation-guided cross attention map for few-shot classification where the correlation between the class feature and query feature generates the attention weights. The work of [186] introduces a cross-channel communication block to encourage information exchange across channels at the convolutional layer. To accelerate the channel interaction, Wang *et al.* [187] propose a 1D convolution across channels for attention prediction. Different from prior work, our attention maps are constructed by demographic information (see Figs. 4.11b and Fig. 4.11c), which improves the robustness of face representations in every demographic group.

## 4.4.2 Methodology

### 4.4.2.1 Overview

Our goal is to train a FR network that is impartial to individuals in different demographic groups. Unlike image-related variations, *e.g.*, large-poses or low-resolution faces are harder to be recognized, demographic attributes are subject-related properties with no apparent impact in recognizability of identity, at least from a layman’s perspective. Thus, an unbiased FR system should be able to obtain equally salient features for faces across demographic groups. However, due to imbalanced demographic distributions and inherent face differences between groups, it was shown that certain groups achieve higher performance even with hand-crafted features [14]. Thus, it is impractical to extract features from different demographic groups that exhibit equal discriminability. Despite such disparity, a FR algorithm can still be designed to *mitigate* the difference in performance.

To this end, we propose a CNN-based group adaptive classifier that utilizes dynamic kernels and attention maps to boost FR performance in all demographic groups considered here. Specifically, GAC has two main modules, an adaptive layer and an automation module. In an adaptive layer, face images or feature maps are convolved with a unique kernel for each demographic group, and multiplied with adaptive attention maps to obtain demographic-differential features for faces in a certain group. The automation module determines in which layers of the network adaptive kernels and attention maps should be applied. As shown in Fig. 4.12, given an aligned face, and its identity label  $y_{ID}$ , a pre-trained demographic classifier first estimates its demographic attribute  $y_{Demo}$ . With  $y_{Demo}$ , the image is then fed into a recognition network with multiple demographic adaptive layers to estimate its identity. In the following, we present these two modules.

### 4.4.2.2 Adaptive Layer

*Adaptive Convolution.* For a standard convolution in CNN, an image or feature map from the previous layer  $X \in \mathbb{R}^{c \times h^X \times w^X}$  is convolved with a single kernel matrix  $K \in \mathbb{R}^{k \times c \times h^K \times w^K}$ , where  $c$  is the number of input channels,  $k$  the number of filters,  $h^X$  and  $w^X$  the input size, and  $h^K$  and  $w^K$  the

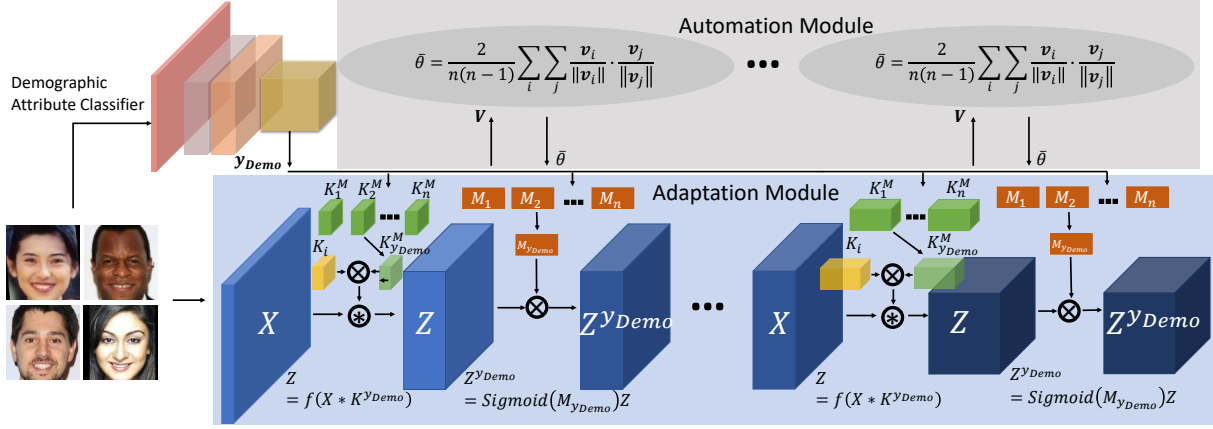


Figure 4.12 Overview of the proposed GAC for mitigating FR bias. GAC contains two major modules: the adaptive layer and the automation module. The adaptive layer consists of adaptive kernels and attention maps. The automation module is employed to decide whether a layer should be adaptive or not.

filter size. Such an operation shares the kernel with every input going through the layer, and is thus agnostic to demographic content, resulting in limited capacity to represent minority groups. To mitigate the bias in convolution, we introduce a trainable matrix of kernel masks  $K^M \in \mathbb{R}^{n \times c \times h^K \times w^K}$ , where  $n$  is the number of demographic groups. In the forward pass, the demographic label  $y_{Demo}$  and kernel matrix  $K^M$  are fed into the adaptive convolutional layer to generate demographic adaptive filters. Let  $K_i \in \mathbb{R}^{c \times h^K \times w^K}$  denote the  $i^{th}$  channel of the shared filter. The  $i^{th}$  channel of adaptive filter for group  $y_{Demo}$  is:

$$K_i^{y_{Demo}} = K_i \otimes K_{y_{Demo}}^M, \quad (4.5)$$

where  $K_{y_{Demo}}^M \in \mathbb{R}^{c \times h^K \times w^K}$  is the  $y_{Demo}^{th}$  kernel mask for group  $y_{Demo}$ , and  $\otimes$  denotes element-wise multiplication. Then the  $i^{th}$  channel of the output feature map is given by  $Z_i = f(X * K_i^{y_{Demo}})$ , where  $*$  denotes convolution, and  $f(\cdot)$  is activation. Unlike conventional convolution, samples in every demographic group have a unique kernel  $K^{y_{Demo}}$ .

*Adaptive Attention.* Each channel filter in a CNN plays an important role in every dimension of the final representation, which can be viewed as a semantic pattern detector [205]. In the adaptive convolution, however, the values of a kernel mask are broadcast along the channel dimension, indicating that the weight selection is spatially varied but channel-wise joint. Hence, we introduce a

channel-wise attention mechanism to enhance the face features that are demographic-adaptive. First, a trainable matrix of channel attention maps  $M \in \mathbb{R}^{n \times k}$  is initialized in every adaptive attention layer. Given  $y_{Demo}$  and the current feature map  $Z \in \mathbb{R}^{k \times h^Z \times w^Z}$ , where  $h^Z$  and  $w^Z$  are the height and width of  $Z$ , the  $i^{th}$  channel of the new feature map is calculated by:

$$Z_i^{y_{Demo}} = \text{Sigmoid}(M_{y_{Demo}i}) \cdot Z_i, \quad (4.6)$$

where  $M_{y_{Demo}i}$  is the entry in the  $y_{Demo}^{th}$  row of  $M$  for the demographic group  $y_{Demo}$  at  $i^{th}$  column. In contrast to the adaptive convolution, elements of each demographic attention map  $M_{y_{Demo}}$  diverge in channel-wise manner, while the single attention weight  $M_{y_{Demo}i}$  is spatially shared by the entire matrix  $Z_i \in \mathbb{R}^{h^Z \times w^Z}$ . The two adaptive matrices,  $K^M$  and  $M$ , are jointly tuned with all the other parameters supervised by the classification loss.

Unlike dynamic CNNs [185] where additional networks are engaged to produce input-variant kernel or attention map, our adaptiveness is yielded by a simple thresholding function directly pointing to the demographic group with no auxiliary networks. Although the kernel network in [185] can generate continuous kernels without enlarging the parameter space, further encoding is required if the side inputs for kernel network are discrete variables. Our approach, in contrast, divides kernels into clusters so that the branch parameter learning can stick to a specific group without interference from individual uncertainties, making it suitable for discrete domain adaptation. Further, the adaptive kernel masks in GAC are more efficient in terms of the number of additional parameters. Compared to a non-adaptive layer, the number of additional parameters of GAC is  $n \times c \times h^K \times w^K$ , while that of [185] is  $s \times k \times c \times h^K \times w^K$  if the kernel network is a one-layer MLP, where  $s$  is the dimension of input side information. Thus, for one adaptive layer, [185] has  $\frac{s \times k}{n}$  times more parameters than ours, which can be substantial given the typical large value of  $k$ , the number of filters.



#### 4.4.2.3 Automation Module

Though faces in different demographic groups are adaptively processed by various kernels and attention maps, it is inefficient to use such adaptations in *every* layer of a deep CNN. To relieve the burden of unnecessary parameters and avoid empirical trimming, we adopt a similarity fusion process to automatically determine the adaptive layers. Since the same fusion scheme can be applied to both types of adaptation, we take the adaptive convolution as an example to illustrate this automatic scheme.

First, a matrix composed of  $n$  kernel masks is initialized in every convolutional layer. As training continues, each kernel mask is updated independently to reduce classification loss for each demographic group. Second, we reshape the kernel masks into 1D vectors  $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_n]$ , where  $\mathbf{v}_i \in \mathbb{R}^l, l = c \times w^K \times h^K$  is the kernel mask of the  $i^{th}$  demographic group. Next, we compute Cosine similarity between two kernel vectors,  $\theta_{ij} = \frac{\mathbf{v}_i}{\|\mathbf{v}_i\|} \cdot \frac{\mathbf{v}_j}{\|\mathbf{v}_j\|}$ , where  $1 \leq i, j \leq n$ . The average similarity of all pair-wise similarities is obtained by  $\bar{\theta} = \frac{2}{n(n-1)} \sum_i \sum_j \theta_{ij}, i \neq j$ . If the dissimilarity  $-\bar{\theta}$  is lower than a pre-defined threshold  $\tau$ , the kernel parameters in this layer reveal the demographic-agnostic property. Hence, we merge the  $n$  kernels into a single kernel by averaging along the group dimension. By definition, a lower  $\tau$  implies more adaptive layers. Given an array of  $\{-\theta_i\}^t$  ( $t$  is the total number of convolutional layers), we first sort the elements from smallest to highest, and this way, layers whose  $-\theta_i$  values are larger than  $\tau$  will be adaptive. Thus, when  $\tau$  decreases, more layers will be adaptive. In the subsequent training, this single kernel can still be updated separately for each demographic group, as the kernel may become demographic-adaptive in later epochs. We monitor the similarity trend of the adaptive kernels in each layer until  $\bar{\theta}$  is stable.

#### 4.4.2.4 De-biasing Objective Function

Apart from the objective function for face identity classification, we also adopt a regress loss function to narrow the gap of the intra-class distance between demographic groups. Let  $g(\cdot)$  denote the inference function of GAC, and  $I_{ijg}$  is the  $i^{th}$  image of subject  $j$  in group  $g$ . Thus, the feature representation of image  $I_{ijg}$  is given by  $\mathbf{r}_{ijg} = g(I_{ijg}, \mathbf{w})$ , where  $\mathbf{w}$  denotes the GAC

parameters. Assuming the feature distribution of each subject is a Gaussian distribution with identity covariance matrix (hyper-sphere), we utilize the average Euclidean distance to every subject center as the intra-class distance of each subject. In particular, we first compute the center point of each identity-sphere:

$$\boldsymbol{\mu}_{jg} = \frac{1}{N} \sum_{i=1}^N g(I_{ijg}, \mathbf{w}), \quad (4.7)$$

where  $N$  is the total number of face images of subject  $j$ . The average intra-class distance of subject  $j$  is as follows:

$$Dist_{jg} = \frac{1}{N} \sum_{i=1}^N (\mathbf{r}_{ijg} - \boldsymbol{\mu}_{jg})^T (\mathbf{r}_{ijg} - \boldsymbol{\mu}_{jg}). \quad (4.8)$$

We then compute the intra-class distance for all subjects in group  $g$  as  $Dist_g = \frac{1}{Q} \sum_{j=1}^Q Dist_{jg}$ , where  $Q$  is the number of total subjects in group  $g$ . This allows us to lower the difference of intra-class distance by:

$$\mathcal{L}_{bias} = \frac{\lambda}{Q \times n} \sum_{g=1}^n \sum_{j=1}^Q \left| Dist_{jg} - \frac{1}{n} \sum_{g=1}^n Dist_g \right|, \quad (4.9)$$

where  $\lambda$  is the coefficient for the de-biasing objective.

### 4.4.3 Experiments

**Datasets** Our bias study uses RFW dataset [4] for testing and BUPT-Balancedface dataset [5] for training. RFW consists of faces in four race/ethnic groups: White, Black, East Asian, and South Asian <sup>4</sup>. Each group contains  $\sim 10K$  images of 3K individuals for face verification. BUPT-Balancedface contains 1.3M images of 28K celebrities and is approximately race-balanced with 7K identities per race. Other than race, we also study gender bias. We combine IMDB [173], UTKFace [174], AgeDB [175], AAF [177], AFAD [176] to train a gender classifier, which estimates gender of faces in RFW and BUPT-Balancedface. The statistics of the datasets are reported in Tab. 4.1. All face images are cropped and resized to  $112 \times 112$  pixels via landmarks detected by RetinaFace [34].

---

<sup>4</sup>RFW [4] uses Caucasian, African, Asian, and Indian to name demographic groups. We adopt these groups and accordingly rename to White, Black, East Asian, and South Asian for clearer race/ethnicity definition.

**Implementation Details** We train a baseline network and GAC on BUPT-Balancedface, using the 50-layer ArcFace architecture [6]. The classification loss is an additive Cosine margin in Cosface [19], with the scale and margin of  $s = 64$  and  $m = 0.5$ . Training is optimized by SGD with a batch size 256. The learning rate starts from 0.1 and drops to 0.0001 following the schedule at 8, 13, 15 epochs for the baseline, and 5, 17, 19 epochs for GAC. We set  $\lambda = 0.1$  for the intra-distance de-biasing.  $\tau = -0.2$  is chosen for automatic adaptation in GAC. Our FR models are trained to extract a 512-dim representation. Our demographic classifier uses a 18-layer ResNet [50]. Comparing GAC and the baseline, the average feature extraction speed per image on Nvidia 1080Ti GPU is 1.4ms and 1.1ms, and the number of model parameters is 44.0M and 43.6M, respectively.

**Performance Metrics** The common group fairness criteria like demographic parity distance [150] are improper to evaluate fairness of learnt representations, since they are designed to measure independence properties of random variables. However, in FR the sensitive demographic characteristics are tied to identities, making these two variables correlated. The NIST report uses false negative and false positive for each demographic group to measure the fairness [7]. Instead of plotting false negative vs. false positives, we adopt a compact quantitative metric, *i.e.*, the standard deviation (STD) of the performance in different demographic groups, previously introduced in [5, 74] and called “biasness”. As bias is considered as systematic error of the estimated values compared to the actual values, here, we assume the average performance to be the actual value. For each demographic group, its biasness is the error between the average and the performance on demographic group. The overall biasness is the expectation of all group errors, which is the STD of performance across groups. We also report average accuracy (Avg) to show the overall FR performance.

#### 4.4.3.1 Results on RFW Protocol

We follow RFW face verification protocol with 6K pairs per race/ethnicity. The models are trained on BUPT-Balancedface with ground truth race and identity labels.

**Compare with SOTA.** We compare the GAC with four SOTA algorithms on RFW protocol, namely, ACNN [185], RL-RBN [5], PFE [184], and DebFace [74]. Since the approach in ACNN [185] is

Table 4.6 Performance comparison with SOTA on the RFW protocol [4]. The results marked by (\*) are directly copied from [5].

Method	White	Black	East Asian	South Asian	Avg ( $\uparrow$ )	STD ( $\downarrow$ )
RL-RBN [5]	96.27	95.00	94.82	94.68	95.19	0.63
ACNN [185]	96.12	94.00	93.67	94.55	94.58	0.94
PFE [184]	<b>96.38</b>	<b>95.17</b>	94.27	94.60	95.11	0.93
ArcFace [6]	96.18*	94.67*	93.72*	93.98*	94.64	0.96
CosFace [19]	95.12*	93.93*	92.98*	92.93*	93.74	0.89
DebFace [74]	95.95	93.67	94.33	94.78	94.68	0.83
GAC	96.20	94.77	<b>94.87</b>	<b>94.98</b>	<b>95.21</b>	<b>0.58</b>

related to GAC, we re-implement it and apply to the bias mitigation problem. First, we train a race classifier with the cross-entropy loss on BUPT-Balancedface. Then the softmax output of our race classifier is fed to a filter manifold network (FMN) to generate adaptive filter weights. Here, FMN is a two-layer MLP with a ReLU in between. Similar to GAC, race probabilities are considered as auxiliary information for face representation learning. We also compare with the SOTA approach PFE [184] by training it on BUPT-Balancedface. As shown in Tab. 4.6, GAC is superior to SOTA w.r.t. average performance and feature fairness. Compared to kernel masks in GAC, the FMN in ACNN [185] contains more trainable parameters. Applying it to each convolutional layer is prone to overfitting. In fact, the layers that are adaptive in GAC ( $\tau = -0.2$ ) are set to be the FMN based convolution in ACNN. As the race data is a four-element input in our case, using extra kernel networks adds complexity to the FR network, which degrades the verification performance. Even though PFE performs the best on standard benchmarks (Tab. 4.15), it still exhibits high biasness. Our GAC outperforms PFE on RFW in both biasness and average performance. Compared to DebFace [74], in which demographic attributes are disentangled from the identity representations, GAC achieves higher verification performance by optimizing the classification for each demographic group, with a lower biasness as well.

To further present the superiority of GAC over the baseline model in terms of bias, we plot Receiver Operating Characteristic (ROC) curves to show the values of True Acceptance Rate (TAR) at various values of False Acceptance Rate (FAR). Fig. 4.13 shows the ROC performance of GAC and the baseline model on RFW. We see that the curves of demographic groups generated by GAC

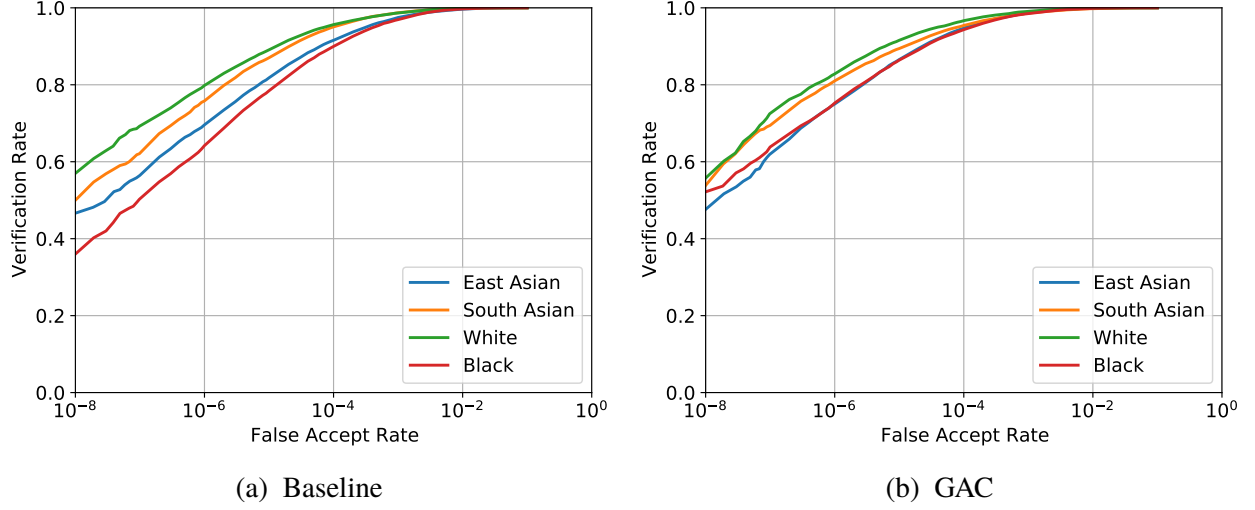


Figure 4.13 ROC of (a) baseline and (b) GAC evaluated on all pairs of RFW.

suggest smaller gaps in TAR at every FAR, which demonstrates the de-biasing capability of GAC. Fig. 4.14 shows pairs of false positives (two faces falsely verified as the same identity) and false negatives in RFW dataset.

**Ablation on Adaptive Strategies.** To investigate the efficacy of our network design, we conduct three ablation studies: adaptive mechanisms, number of convolutional layers, and demographic information. For adaptive mechanisms, since deep feature maps contain both spatial and channel-wise information, we study the relationship among adaptive kernels, spatial and channel-wise attentions, and their impact to bias mitigation. We also study the impact of  $\tau$  in our automation module. Apart from the baseline and GAC, we ablate eight variants: (1) GAC-Channel: channel-wise attention for race-differential feature; (2) GAC-Kernel: adaptive convolution with race-specific kernels; (3) GAC-Spatial: only spatial attention is added to baseline; (4) GAC-CS: both channel-wise and spatial attention; (5) GAC-CSK: combine adaptive convolution with spatial and channel-wise attention; (6,7,8) GAC- $(\tau = *)$ : set  $\tau$  to  $*$ .

From Tab. 4.7, we make several observations: (1) the baseline model is the most biased across race groups. (2) spatial attention mitigates the race bias at the cost of verification accuracy, and is less effective on learning fair features than other adaptive techniques. This is probably because spatial contents, especially local layout information, only reside at earlier CNN layers, where the



Figure 4.14 8 false positive and false negative pairs on RFW given by the baseline but successfully verified by GAC.

spatial dimensions are gradually decreased by the later convolutions and poolings. Thus, semantic details like demographic attributes are hardly encoded spatially. (3) Compared to GAC, combining adaptive kernels with both spatial and channel-wise attention increases the number of parameters, lowering the performance. (4) As  $\tau$  determines the number of adaptive layers in GAC, it has a great impact on the performance. A small  $\tau$  may increase redundant adaptive layers, while the adaptation layers may lack in capacity if too large.

**Ablation on Depths and Demographic Labels.** Both the adaptive layers and de-biasing loss in GAC can be applied to CNN in any depth. In this ablation, we train both the baseline and GAC ( $\lambda = 0.1$ ,  $\tau = -0.2$ ) in ArcFace architecture with three different numbers of layers: 34, 50, and 100. As the training of GAC relies on demographic information, the error and bias in demographic labels might impact the bias reduction of GAC. Thus, we also ablate with different demographic

Table 4.7 Ablation of adaptive strategies on the RFW protocol [4].

Method	White	Black	East Asian	South Asian	Avg ( $\uparrow$ )	STD ( $\downarrow$ )
Baseline	96.18	93.98	93.72	94.67	94.64	1.11
GAC-Channel	95.95	93.67	94.33	94.78	94.68	0.83
GAC-Kernel	96.23	94.40	94.27	94.80	94.93	0.78
GAC-Spatial	95.97	93.20	93.67	93.93	94.19	1.06
GAC-CS	96.22	93.95	94.32	<b>95.12</b>	94.65	0.87
GAC-CSK	96.18	93.58	94.28	94.83	94.72	0.95
GAC-( $\tau = 0$ )	96.18	93.97	93.88	94.77	94.70	0.92
GAC-( $\tau = -0.1$ )	<b>96.25</b>	94.25	94.83	94.72	95.01	0.75
GAC-( $\tau = -0.2$ )	96.20	<b>94.77</b>	<b>94.87</b>	94.98	<b>95.21</b>	<b>0.58</b>

Table 4.8 Ablation of CNN depths and demographics on RFW protocol [4].

Method	White	Black	East Asian	South Asian	Avg ( $\uparrow$ )	STD ( $\downarrow$ )
Number of Layers						
ArcFace-34	96.13	93.15	92.85	93.03	93.78	1.36
GAC-ArcFace-34	96.02	94.12	94.10	94.22	94.62	0.81
ArcFace-50	96.18	93.98	93.72	94.67	94.64	1.11
GAC-ArcFace-50	96.20	94.77	94.87	94.98	95.21	0.58
ArcFace-100	96.23	93.83	94.27	94.80	94.78	0.91
GAC-ArcFace-100	96.43	94.53	94.90	95.03	95.22	0.72
Race/Ethnicity Labels						
Ground-truth	96.20	94.77	94.87	94.98	95.21	0.58
Estimated	96.27	94.40	94.32	94.77	94.94	0.79
Random	95.95	93.10	94.18	94.82	94.50	1.03

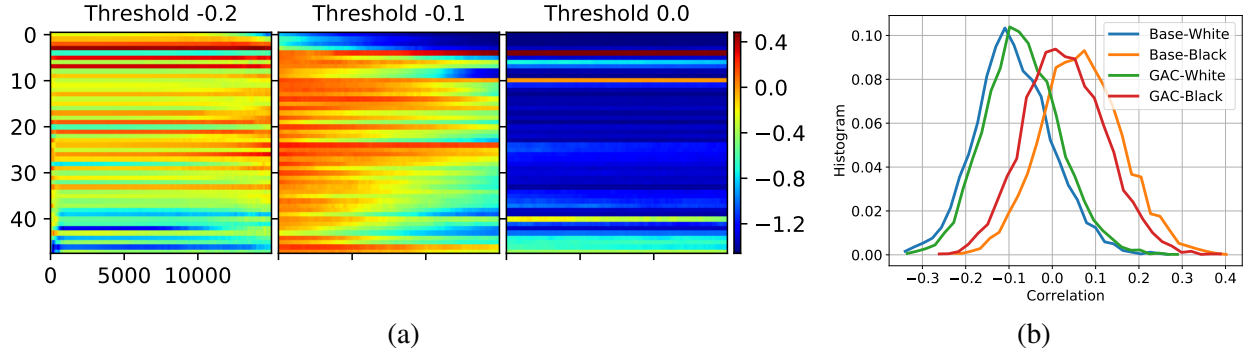


Figure 4.15 (a) For each of the three  $\tau$  in automatic adaptation, we show the average similarities of pair-wise demographic kernel masks, *i.e.*,  $\bar{\theta}$ , at 1-48 layers (y-axis), and 1-15K training steps (x-axis). The number of adaptive layers in three cases, *i.e.*,  $\sum_1^{48}(\bar{\theta} > \tau)$  at 15K<sup>th</sup> step, are 12, 8, and 2, respectively. (b) With two race groups (White, Black in PCSO [14]) and two models (baseline, GAC), for each of the four combinations, we compute pair-wise correlation of face representations using any two of 1K subjects in the same race, and plot the histogram of correlations. GAC reduces the difference/bias of two distributions.

Table 4.9 Ablations on  $\lambda$  on RFW protocol (%).

$\lambda$	White	Black	East Asian	South Asian	Avg ( $\uparrow$ )	STD ( $\downarrow$ )
0	96.23	94.65	94.93	95.12	95.23	0.60
0.1	96.20	94.77	94.87	94.98	95.21	0.58
0.5	94.89	94.00	93.67	94.55	94.28	0.47

Table 4.10 Verification Accuracy (%) of 5-fold cross-validation on 8 groups of RFW [4].

Method	Gender	White	Black	East Asian	South Asian	Avg ( $\uparrow$ )	STD ( $\downarrow$ )
Baseline	Male	97.49 $\pm$ 0.08	96.94 $\pm$ 0.26	97.29 $\pm$ 0.09	97.03 $\pm$ 0.13	96.96 $\pm$ 0.03	0.69 $\pm$ 0.04
	Female	97.19 $\pm$ 0.10	97.93 $\pm$ 0.11	95.71 $\pm$ 0.11	96.01 $\pm$ 0.08		
AL+Manual	Male	98.57 $\pm$ 0.10	98.05 $\pm$ 0.17	98.50 $\pm$ 0.12	<b>98.36 <math>\pm</math> 0.02</b>	98.09 $\pm$ 0.05	0.66 $\pm$ 0.07
	Female	98.12 $\pm$ 0.18	<b>98.97 <math>\pm</math> 0.13</b>	96.83 $\pm$ 0.19	97.33 $\pm$ 0.13		
GAC	Male	<b>98.75 <math>\pm</math> 0.04</b>	<b>98.18 <math>\pm</math> 0.20</b>	<b>98.55 <math>\pm</math> 0.07</b>	98.31 $\pm$ 0.12	<b>98.19 <math>\pm</math> 0.06</b>	<b>0.56 <math>\pm</math> 0.05</b>
	Female	<b>98.26 <math>\pm</math> 0.16</b>	98.80 $\pm$ 0.15	<b>97.09 <math>\pm</math> 0.12</b>	<b>97.56 <math>\pm</math> 0.10</b>		

information, (1) ground-truth: the race/ethnicity labels provided by RFW; (2) estimated: the labels predicted by a pre-trained race estimation model; (3) random: the demographic label randomly assigned to each face.

As shown in Tab. 4.8, compared to the baselines, GAC successfully reduces the STD at different number of layers. We see that the model with least number of layers presents the most bias, and the bias reduction by GAC is the most as well. The noise and bias in demographic labels do, however, impair the performance of GAC. With estimated demographics, the biasness is higher than that of the model with ground-truth supervision. Meanwhile, the model trained with random demographics has the highest biasness. Even so, using estimated attributes during testing still improves fairness in face recognition compared to baseline. This indicates the efficacy of GAC even in the absence of ground-truth labels.

**Ablation on  $\lambda$ .** We use  $\lambda$  to control the weight of de-biasing loss. Tab. 4.9 reports the results of GAC trained with different values of  $\lambda$ . When  $\lambda = 0$ , de-biasing loss is removed in training. The results indicate a larger  $\lambda$  leads to lower biasness at the cost of overall accuracy.

#### Ablation on Automation Module

Here, we also ablate GAC with two variants to show the efficiency of its automation module: i) *Ada-All*, *i.e.*, all the convolutional layers are adaptive and ii) *Ada-8*, *i.e.*, the same 8 layers as GAC are set to be adaptive starting from the beginning of the training process, with no automation



Table 4.11 Ablations on the automation module on RFW protocol (%).

Method	White	Black	East Asian	South Asian	Avg ( $\uparrow$ )	STD ( $\downarrow$ )
Ada-All	93.22	90.95	91.32	92.12	91.90	0.87
Ada-8	96.25	94.40	94.35	95.12	95.03	0.77
GAC	96.20	94.77	94.87	94.98	95.21	0.58

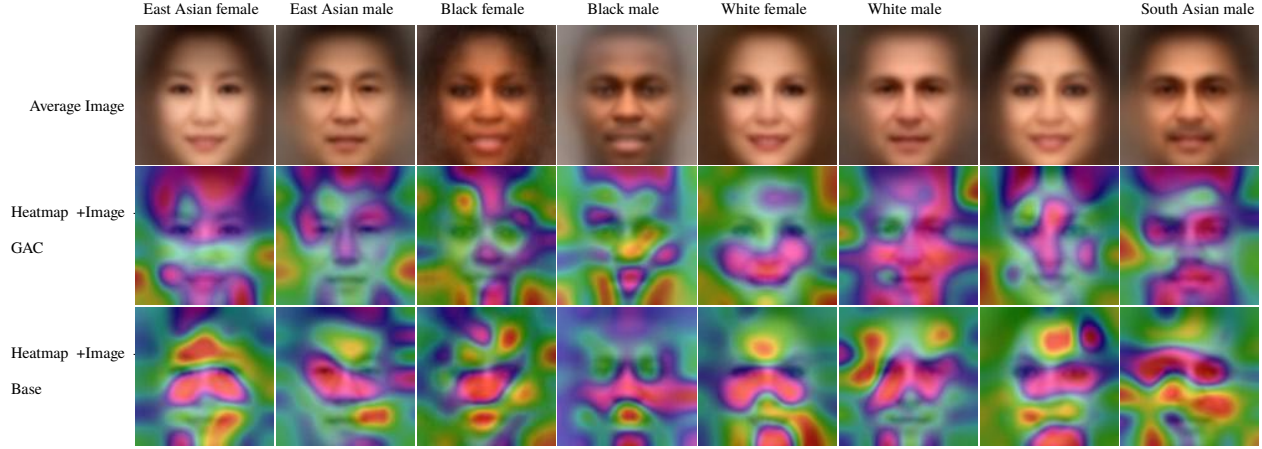


Figure 4.16 The first row shows the average faces of different groups in RFW. The next two rows show gradient-weighted class activation heatmaps [15] at the 43<sup>th</sup> convolutional layer of the GAC and baseline. The higher diversity of heatmaps in GAC shows the variability of parameters in GAC across groups.

module (our best GAC model has 8 adaptive layers). As in Tab. 4.11, with automation module, GAC achieves higher average accuracy and lower biasness than the other two models.

#### 4.4.3.2 Results on Gender and Race Groups

Table 4.12 Statistics of dataset folds in the cross-validation experiment.

Fold	White (#)		Black (#)		East Asian (#)		South Asian (#)	
	Subjects	Images	Subjects	Images	Subjects	Images	Subjects	Images
1	1,991	68,159	1,999	67,880	1,898	67,104	1,996	57,628
2	1,991	67,499	1,999	65,736	1,898	66,258	1,996	57,159
3	1,991	66,091	1,999	65,670	1,898	67,696	1,996	56,247
4	1,991	66,333	1,999	67,757	1,898	65,341	1,996	57,665
5	1,994	68,597	1,999	67,747	1,898	68,763	2,000	56,703

We now extend demographic attributes to both gender and race. First, we train two classifiers that predict gender and race/ethnicity of a face image. The classification accuracy of gender and

Table 4.13 Verification (%) on gender groups of IJB-C (TAR @ 0.1% FAR).

Model	Male	Female	Avg ( $\uparrow$ )	STD ( $\downarrow$ )
Baseline	89.72	79.57	84.64	5.08
GAC	88.25	83.74	86.00	2.26

race/ethnicity is 85% and 81%<sup>5</sup>, respectively. Then, these fixed classifiers are affiliated with GAC to provide demographic information for learning adaptive kernels and attention maps. We merge BUPT-Balancedface and RFW, and split the subjects into 5 sets for each of 8 demographic groups. In 5-fold cross-validation, each time a model is trained on 4 sets and tested on the remaining set. Tab. 4.12 reports the statistics of each data fold for the cross-validation experiment on BUPT-Balancedface and RFW datasets.

Here we demonstrate the efficacy of the automation module for GAC. We compare to the scheme of manually design (AL+Manual) that adds adaptive kernels and attention maps to a subset of layers. Specifically, the first block in every residual unit is chosen to be the adaptive convolution layer, and channel-wise attentions are applied to the feature map output by the last block in each residual unit. As we use 4 residual units and each block has 2 convolutional layers, the manual scheme involves 8 adaptive convolutional layers and 4 groups of channel-wise attention maps. As in Tab. 4.10, automatic adaptation is more effective in enhancing the discriminability and fairness of face representations. Figure 4.15a shows the dissimilarity of kernel masks in the convolutional layers changes during training epochs under three thresholds  $\tau$ . A lower  $\tau$  results in more adaptive layers. We see the layers that are determined to be adaptive do vary across both layers (vertically) and training time (horizontally), which shows the importance of our automatic mechanism.

Since IJB-C also provides gender labels, we evaluate our GAC-gender model (see Sec. 4.2 of the main paper) on IJB-C as well. Specifically, we compute the verification TAR at 0.1% FAR on the pairs of female faces and male faces, respectively. Tab. 4.13 reports the TAR @ 0.1% FAR on

<sup>5</sup>This seemingly low accuracy is mainly due to the large dataset we assembled for training and testing gender/race classifiers. Our demographic classifier has been shown to perform comparably as SOTA on common benchmarks. While demographic estimation errors impact the training, testing, and evaluation of bias mitigation algorithms, the evaluation is of the most concern as demographic label errors may greatly impact the biasness calculation. Thus, future development may include either manually cleaning the labels, or designing a biasness metric robust to label errors.

Table 4.14 Verification accuracy (%) on the RFW protocol [4] with varying race/ethnicity distribution in the training set.

Training Ratio	White	Black	East Asian	South Asian	Avg ( $\uparrow$ )	STD ( $\downarrow$ )
7 : 7 : 7 : 7	96.20	94.77	94.87	94.98	95.21	0.58
5 : 7 : 7 : 7	96.53	94.67	94.55	95.40	95.29	0.79
3.5 : 7 : 7 : 7	96.48	94.52	94.45	95.32	95.19	0.82
1 : 7 : 7 : 7	95.45	94.28	94.47	95.13	94.83	0.48
0 : 7 : 7 : 7	92.63	92.27	92.32	93.37	92.65	0.44

gender groups of IJB-C. The biasness of GAC is still lower than the baseline for different gender groups of IJB-C.

#### 4.4.3.3 Analysis on Intrinsic Bias and Data Bias

For all the algorithms listed in Tab. 1 of the main paper, the performance is higher in White group than those in the other three groups, even though all the models are trained on a demographic balanced dataset, BUPT-Balancedface [5]. In this section, we further investigate the intrinsic bias of face recognition between demographic groups and the impact of the data bias in the training set. *Are non-White faces inherently difficult to be recognized for existing algorithms? Or, are face images in BUPT-Balancedface (the training set) and RFW [4] (testing set) biased towards the White group?*

To this end, we train our GAC network using training sets with different race/ethnicity distributions and evaluate them on RFW. In total, we conduct four experiments, in which we gradually reduce the total number of subjects in the White group from the BUPT-Balancedface dataset. To construct a new training set, subjects from the non-White groups in BUPT-Balancedface remain the same, while a subset of subjects is randomly picked from the White group. As a result, the ratios between non-White groups are consistently the same, and the ratios of White, Black, East Asian, South Asian are  $\{5 : 7 : 7 : 7\}$ ,  $\{3.5 : 7 : 7 : 7\}$ ,  $\{1 : 7 : 7 : 7\}$ ,  $\{0 : 7 : 7 : 7\}$  in the four experiments, respectively. In the last setting, we completely remove White from the training set.

Tab. 4.14 reports the face verification accuracy of models trained with different race/ethnicity distributions on RFW. For comparison, we also put our results on the balanced dataset here (with ratio  $\{7 : 7 : 7 : 7\}$ ), where all images in BUPT-Balancedface are used for training. From the results,

Table 4.15 Verification performance on LFW, IJB-A, and IJB-C. [Key: **Best**, *Second*, Third Best]

Method	LFW (%)	Method	IJB-A (%)	IJB-C @ FAR (%)		
			0.1% FAR	0.001%	0.01%	0.1%
DeepFace+ [17]	97.35	Yin <i>et al.</i> [182]	$73.9 \pm 4.2$	-	-	69.3
CosFace [19]	99.73	Cao <i>et al.</i> [59]	$90.4 \pm 1.4$	74.7	84.0	91.0
ArcFace [6]	<b>99.83</b>	Multicolumn [183]	$92.0 \pm 1.3$	77.1	86.2	92.7
PFE [184]	99.82	PFE [184]	<b><math>95.3 \pm 0.9</math></b>	<b>89.6</b>	<b>93.3</b>	<b>95.5</b>
Baseline	99.75	Baseline	$90.2 \pm 1.1$	80.2	88.0	92.9
GAC	<u>99.78</u>	GAC	<u><math>91.3 \pm 1.2</math></u>	83.5	89.2	93.7

we see several observations: (1) It shows that the White group still outperforms the non-White groups for all the first three experiments. Even without any White subjects in the training set, the accuracy on the White testing set is still higher than those on the testing images in Black and East Asian groups. This suggests that White faces are either intrinsically easier to be verified or face images in the White group of RFW are less challenging. (2) With the decline in the total number of White subjects, the average performance declines as well. In fact, for all these groups, the performance suffers from the decrease in the number of White faces. This indicates that face images in the White groups are helpful to boost the face recognition performance for both White and non-White faces. In other words, faces from the White group benefit the representation learning of global patterns for face recognition in general. (3) Opposite to our intuition, the biasness is lower with less number of White faces, while the data bias is actually increased by adding the unbalancedness to the training set.

#### 4.4.3.4 Results on Standard Benchmark Datasets

While our GAC mitigates bias, we also hope it can perform well on standard benchmarks. Therefore, we evaluate GAC on standard benchmarks without considering demographic impacts, including LFW [27], IJB-A [29], and IJB-C [9]. These datasets exhibit imbalanced distribution in demographics. For a fair comparison with SOTA, instead of using ground truth demographics, we train GAC on Ms-Celeb-1M [23] with the demographic attributes estimated by the classifier pre-trained in Sec. 4.4.3.2. As in Tab. 4.15, GAC outperforms the baseline and performs comparable to SOTA.

Table 4.16 Distribution of ratios between minimum inter-class distance and maximum intra-class distance of face features in 4 race groups of RFW. GAC exhibits higher ratios, and more similar distributions to the reference.

Race	Mean		StaD		Relative Entropy	
	Baseline	GAC	Baseline	GAC	Baseline	GAC
White	1.15	1.17	0.30	0.31	0.0	0.0
Black	1.07	1.10	0.27	0.28	0.61	0.43
East Asian	1.08	1.10	0.31	0.32	0.65	0.58
South Asian	1.15	1.18	0.31	0.32	0.19	0.13

#### 4.4.3.5 Visualization and Analysis on Bias of FR

**Visualization** To understand the adaptive kernels in GAC, we visualize the feature maps at an adaptive layer for faces of various demographics, via a Pytorch visualization tool [214]. We visualize important face regions pertaining to the FR decision by using a gradient-weighted class activation mapping (Grad-CAM) [15]. Grad-CAM uses the gradients back from the final layer corresponding to an input identity, and guides the target feature map to highlight import regions for identity predicting. Figure 4.16 shows that, compared to the baseline, the salient regions of GAC demonstrate more diversity on faces from different groups. This illustrates the variability of network parameters in GAC across different groups.

**Bias via Local Geometry** In addition to STD, we explain the bias phenomenon via the local geometry of a given face representation in each demographic group. We assume that the statistics of neighbors of a given point (representation) reflects certain properties of its manifold (local geometry). Thus, we illustrate the pair-wise correlation of face representations. To minimize variations caused by other variables, we use constrained frontal faces of a mug shot dataset, PCSO [14]. We randomly select 1K White and 1K Black subjects from PCSO, and compute their pair-wise correlation within each race. In Fig. 4.15b, Base-White representations show lower inter-class correlation than Base-Black, *i.e.*, faces in the White group are over-represented by the baseline than the Black group. In contrast, GAC-White and GAC-Black shows more similarity in their correlation histograms.

As PCSO has few Asian subjects, we use RFW for another examination of the local geometry in 4 groups. That is, after normalizing the representations, we compute the pair-wise Euclidean

Table 4.17 Network complexity and inference time.

Model	Input Resolution	# Parameters (M)	MACs (G)	Inference (ms)
Baseline	$112 \times 112$	43.58	5.96	1.1
GAC	$112 \times 112$	44.00	9.82	1.4

distance and measure the ratio between the minimum distance of inter-subjects pairs and the maximum distance of intra-subject pairs. We compute the mean and standard deviation (Std) of ratio distributions in 4 groups, by two models. Also, we gauge the relative entropy to measure the deviation of distributions from each other. For simplicity, we choose White group as the reference distribution. As shown in Tab. 4.16, while GAC has minor improvement over baseline in the mean, it gives smaller relative entropy in the other 3 groups, indicating that the ratio distributions of other races in GAC are more similar, *i.e.*, less biased, to the reference distribution. These results demonstrate the capability of GAC to increase fairness of face representations.

#### 4.4.3.6 Network Complexity and FLOPs

Tab. 4.17 summarizes the network complexity of GAC and the baseline in terms of the number of parameters, multiplier–accumulator, and inference times. While we agree the number of parameters will increase with the number of demographic categories, it will not necessarily increase the inference time, which is more important for real-time applications.

## 4.5 Demographic Estimation

We train three demographic estimation models to annotate age, gender, and race information of the face images in BUPT-Balancedface and MS-Celeb-1M for training GAC and DebFace. For all three models, we randomly sample equal number of images from each class and set the batch size to 300. The training process finishes at  $35K^{th}$  iteration. All hyper-parameters are chosen by testing on a separate validation set. Below gives the details of model learning and estimation performance of each demographic.

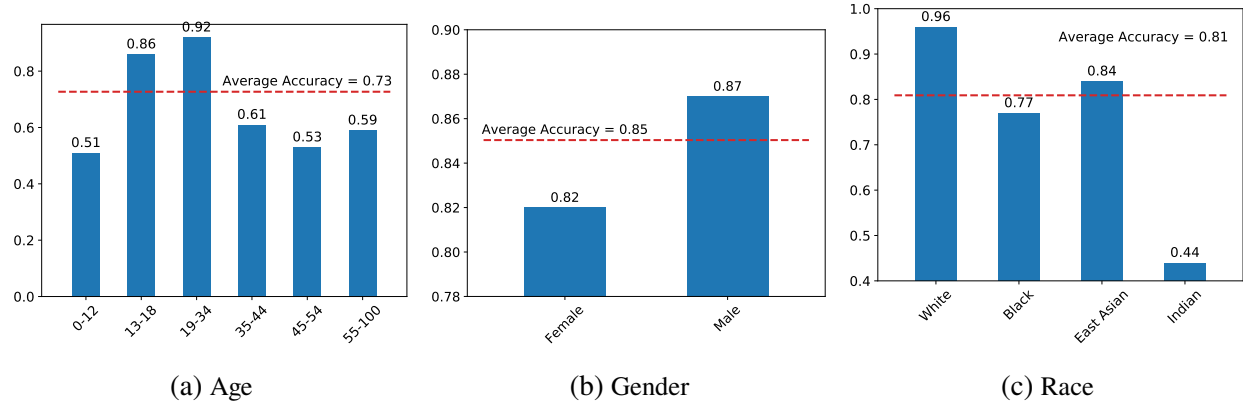


Figure 4.17 Demographic Attribute Classification Accuracy on each group. The red dashed line refers to the average accuracy on all images in the testing set.

Table 4.18 Gender distribution of the datasets for gender estimation.

Dataset	# of Images	
	Male	Female
Training	321,590	229,000
Testing	15,715	10,835

**Gender:** We combine IMDB, UTKFace, AgeDB, AFAD, and AAF datasets for learning the gender estimation model. Similar to age, 90% of the images in the combined datasets are used for training, and the remaining 10% are used for validation. Table 4.18 reports the total number of female and male face images in the training and testing set. More images belong to male faces in both training and testing set. Figure 4.17b shows the gender estimation performance on the validation set. The performance on male images is slightly better than that on female images.

**Race:** We combine AFAD, RFW, IMFDB-CVIT, and PCSO datasets for training the race estimation model. UTKFace is used as validation set. Table 4.19 reports the total number of images in each race category of the training and testing set. Similar to age and gender, the performance of race estimation is highly correlated to the race distribution in the training set. Most of the images are within the White group, while the Indian group has the least number of images. Therefore, the performance on White faces is much higher than that on Indian faces.

**Age:** We combine CACD, IMDB, UTKFace, AgeDB, AFAD, and AAF datasets for learning the age estimation model. 90% of the images in the combined datasets are used for training, and the

Table 4.19 Race distribution of the datasets for race estimation.

Dataset	# of Images			
	White	Black	East Asian	Indian
Training	468,139	150,585	162,075	78,260
Testing	9,469	4,115	3,336	3,748

Table 4.20 Age distribution of the datasets for age estimation

Dataset	# of Images in the Age Group					
	0-12	13-18	19-34	35-44	45-54	55-100
Training	9,539	29,135	353,901	171,328	93,506	59,599
Testing	1,085	2,681	13,848	8,414	5,479	4,690

remaining 10% are used for validation. Table 4.20 reports the total number of images in each age group of the training and testing set, respectively. Figure 4.17a shows the age estimation performance on the validation set. The majority of the images come from the age 19 to 34 group. Therefore, the age estimation performs the best on this group. The performance on the young children and middle to old age group is significantly worse than the majority group.

It is clear that all the demographic models present biased performance with respect to different cohorts. These demographic models are used to label the BUPT-Balancedface and MS-Celeb-1M for training GAC and DebFace. Thus, in addition to the bias from the dataset itself, we also add label bias to it. Since DebFace employs supervised feature disentanglement, we only strive to reduce the data bias instead of the label bias.

## 4.6 Conclusion

This chapter tackles the issue of demographic bias in FR by learning fair face representations. We present two de-biasing FR networks, GAC and DebFace, to mitigate demographic bias in FR. In particular, GAC is proposed to improve robustness of representations for every demographic group considered here. Both adaptive convolution kernels and channel-wise attention maps are introduced to GAC. We further add an automatic adaptation module to determine whether to use adaptations in a given layer. Our findings suggest that faces can be better represented by using layers adaptive to



different demographic groups, leading to more balanced performance gain for all groups. Unlike GAC, DebFace mitigate mutual bias across identities and demographic attributes recognition by adversarially learning the disentangled representation for gender, race, and age estimation, and face recognition simultaneously. We empirically demonstrate that DebFace can not only reduce bias in face recognition but in demographic attribute estimation as well.

## **Chapter 5**

# **Adversarial Face Representation Learning via Graph Classification**

Face representation learning is one of the key steps in FR to overcome challenges caused by variations in face images. In this chapter, we propose a representation learning method that utilizes graph classification via adversarial training. Each face image can be viewed as a node in the graph, and the edges between nodes stand for the connectivity of face samples. A graph classifier is trained to distinguish graphs between those generated by extracted feature vectors and those defined by a practical assumption. Meanwhile, the face representation model attempts to fool the graph classifier so that it can gradually acquire the feature distribution of the ideal oracle graph. In this way, the feature points of the same identity will come closer while maintain a reasonable distance from points of other identities. Experiments on benchmark face datasets (LFW, CPLFW, CFP-FP, IJB-A, IJB-B, IJB-C) show that our framework achieves state-of-the-art performance for both verification and identification tasks.

### **5.1 Adversarial Learning and Graph Classification with GNN**

Adversarial learning [153] has proven to be a useful approach to learning data distribution, and has been applied to many computer vision applications. (Refer to Sec. 4.3.1 for more details on resent

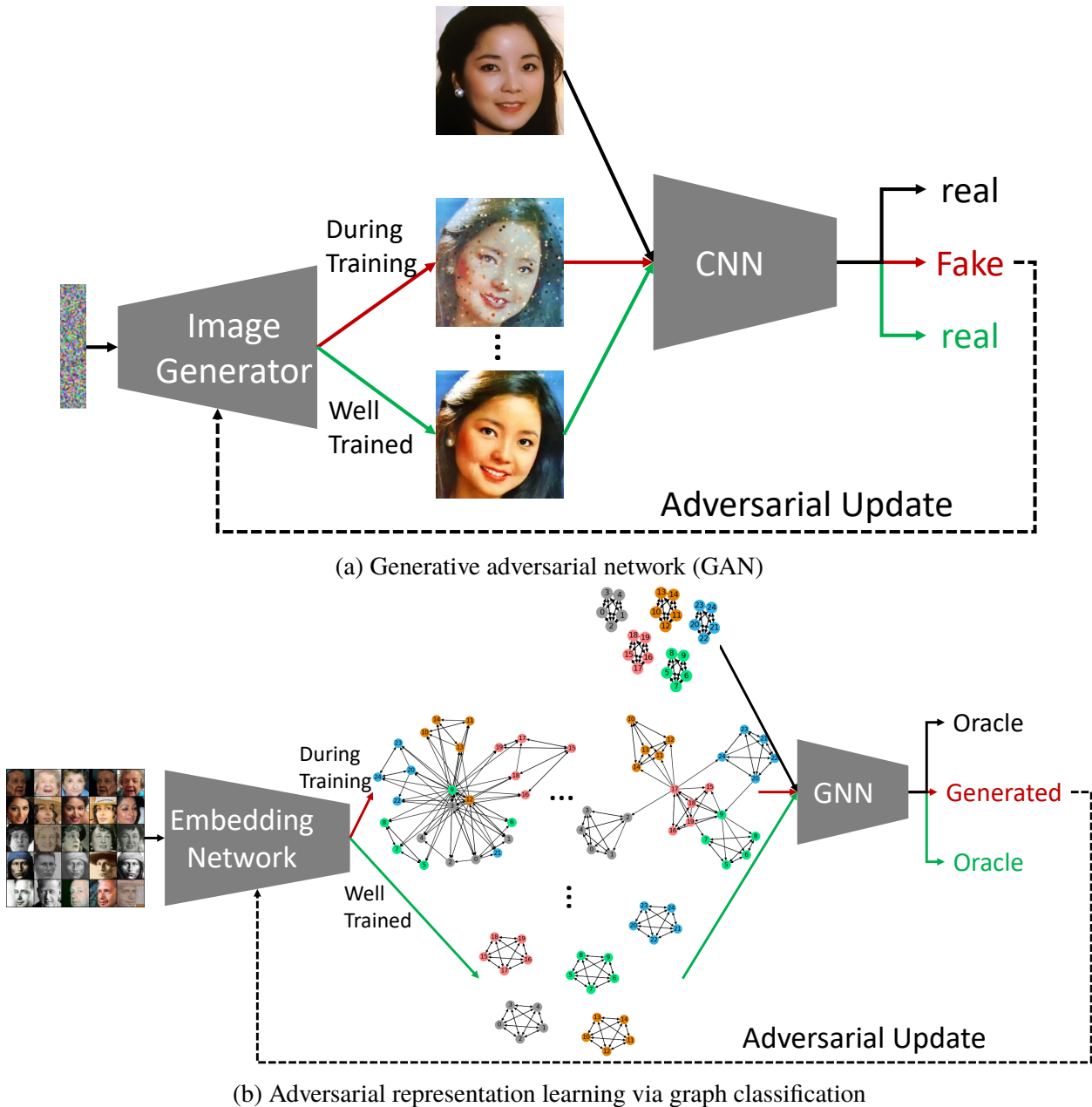


Figure 5.1 (a) In GANs, during training an image generator gradually produces higher quality faces so that a CNN-based discriminator could not distinguish fake from real faces. (b) Analogously, given input faces, our embedding network for face recognition learns to extract discriminative features and connect features as a graph, with the goal that a graph neural network (GNN)-based discriminator could not distinguish generated graphs from oracle graphs — the graph of ideal face representations. During inference, our embedding network can extract more discriminative features that form oracle-like graph, just like GAN’s generator synthesizes photo-realistic faces.

advances in adversarial learning.)

The task of graph classification is to predict the category a graph belongs to. Unlike node label prediction, a full graph structure is considered as a single input component, and the corresponding output is either a single representation vector or a class label. Two main techniques are involved in recent DNN-based graph network: spatial computation and spectral operation. In Spectral approaches [215, 216], the graph convolutions are based on the convolution theorem from signal processing technology, where the point-wise multiplications are performed in the Fourier domain of the graph. In contrast, spatial methods [217–219] operate convolutions directly on the graph structure. Before the labeling procedure, the algorithm in [218] first applies a normalization on the neighborhood graphs created by determining the sequence of nodes. This normalization step moves graphs with similar structural roles in the same neighborhood, and benefits the final classification. Antoine *et al.* [220] modifies the conventional 1D graph convolution to a vanilla 2D CNN architecture for 2D graph classification.

## 5.2 Our Approach

### 5.2.1 Overall Framework

The proposed adversarial training framework for face representation is composed of: a face embedding network  $E(\cdot)$ , a feature graph constructor  $G(\cdot)$ , and a GNN-based graph discriminator  $D(\cdot)$ . First, given a set of  $N$  labeled training images  $\{(\mathbf{x}_i, y_i)\}_i^N$ , the embedding network takes the  $i^{th}$  image  $\mathbf{x}_i$  as the input and transforms  $\mathbf{x}_i$  into a  $m$ -dimensional feature vector:  $\mathbf{f}_i = E(\mathbf{x}_i)$ , where  $\mathbf{f}_i \in \mathcal{R}^m$ . The pair of the feature vector and its corresponding label  $(\mathbf{f}_i, y_i)$  is then sent to the graph constructor  $G(\cdot)$  to be added as a new node in the graph. When  $G(\cdot)$  collects a specified number of nodes, it starts to build graphs based on labels of the nodes and their similarities. For the graphs in the oracle space, two vertices  $(v_i, v_j)$  are connected by a bidirectional edge if they are from the same subject; and for the graphs in the generated space, one vertex  $v_i$  is linked to another  $v_j$  if  $v_i$  is one of the  $k$  nearest neighbors of  $v_j$ . More details of graph construction are discussed in Sec. 5.2.2.

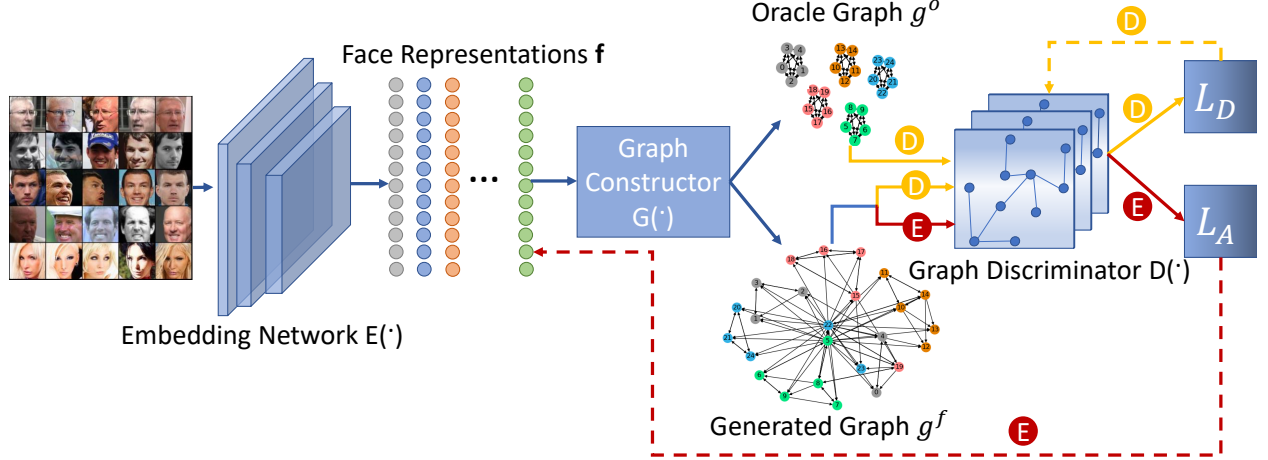


Figure 5.2 Overview of the proposed adversarial face representation learning via graph classification. Solid arrows present forward pass, and dashed arrows denote backward propagation. The training alternates between  $E(\cdot)$  and  $D(\cdot)$ . For the shared inference (solid blue arrows), a set of face images are first taken by the embedding network  $E(\cdot)$  to extract feature representations. These feature vectors are then converted into graph structure by the graph constructor  $G(\cdot)$  in which an oracle graph and a generated graph are constructed. During the training of  $D(\cdot)$  (yellow arrows), the two types of graphs are received by the graph discriminator  $D(\cdot)$  that is required to make predictions on the category of the graphs.  $D(\cdot)$  is then updated based on the gradient sent back from the loss function  $\mathcal{L}_D$ . In the course of training on  $E(\cdot)$  (red arrows), only generated graphs are delivered to  $D(\cdot)$ , and  $E(\cdot)$  receives feedbacks from  $\mathcal{L}_A$  whose goal is to drive  $D(\cdot)$  to make errors on generated graphs.

Next, the two types of graphs produced by  $G(\cdot)$  are fed into the GNN discriminator  $D(\cdot)$  as the training samples. The parameters of  $D(\cdot)$  is updated with the objective to correctly classify the graph category. In the mean time, the adversarial goal of  $E(\cdot)$  is to mislead  $D(\cdot)$  by making the feature distribution in the two spaces from which these graphs are created *indistinguishable*.

As shown in Fig. 5.2, with the cooperation from  $G(\cdot)$ ,  $E(\cdot)$  endeavors to extract “ideal” feature representations by competing against  $D(\cdot)$  using a graph classification network [217]. Such a graph structure can not only allow for distance metric between nodes, but attend to both local and global distributions of feature representations, and hence benefits the discriminative power of face embeddings.

### 5.2.2 Graph Construction

A graph is composed of vertices  $\mathbf{V}$  and edges  $\mathbf{E}$ . In our framework, each vertex is represented by a  $m$ -dimensional feature representation, and an adjacency matrix with binary (0/1) elements is used to describe edge information. The adjacent element equals to one when the two vertices are connected, otherwise it is zero. Given a set of data pairs  $\{(\mathbf{f}_i, y_i)\}_i^N$ , the graph constructor  $G(\cdot)$  produces graphs of two types: (1) an oracle graph  $g^o(\mathbf{V}^o, \mathbf{E}^o)$ , and (2) a generated graph  $g^f(\mathbf{V}^f, \mathbf{E}^f)$ . We introduce the construction of both types of graphs in terms of the definition of their vertices and edges, respectively.

**Oracle Graph** For edges in an oracle graph, one vertex is connected to all the other vertices of the same subject. The resulting adjacency matrix is symmetric:  $\mathbf{A}_{ij}^o = \begin{cases} 1, & y_i = y_j \\ 0, & \text{otherwise} \end{cases}$ . For vertices in an oracle graph, we need to redefine the representation vectors based on both the existing feature pairs  $\{(\mathbf{f}_i, y_i)\}_i^N$  and the ultimate learning objective. For example, in the most ideal situation each face identity would be represented by a single vector, which means all image samples of the same subject would be mapped to the same feature vector, regardless of the intra-subject variations. The corresponding node information matrix is:

$$\mathbf{P}^o = \{\mathbf{f}_{y_i}^c\}_i^N \in \mathcal{R}^{N \times m}, \quad (5.1)$$

where  $\mathbf{f}_j^c = \frac{1}{T_j} \sum_{y_i=j} \mathbf{f}_i$ , and  $T_j$  denotes the total number of face images of the  $j^{th}$  subject (See Fig. 5.3b). However, when such node representations are used as the training target, it is far beyond the realistic data distribution of face images. As a result, it may either be hard to train or lead to trivial solutions.

To make it more practical, we introduce a ratio hyper-parameter,  $r$ , to adaptively adjust the training difficulty in terms of the node representation. Specifically,  $r$  is a fraction number between 0 and 1. It controls the maximum distance of any feature vector to its centroid. For those vectors that violate this constraint, they are forced to move their coordinates towards the direction of the

centroids in an effort to reach the maximum distance. Now we can easily manipulate the training target by changing the value of  $r$ . The final matrix of node information in the oracle graph is:

$$\mathbf{P}_i^o = \begin{cases} \mathbf{f}_i, \text{Dist}(\mathbf{f}_i, \mathbf{f}_{y_i}^c) \leq r \cdot \text{Dist}_{min} \\ \frac{\mathbf{f}_{y_i}^c + r(\mathbf{f}_i - \mathbf{f}_{y_i}^c)\text{Dist}_{min}}{\text{Dist}(\mathbf{f}_i, \mathbf{f}_{y_i}^c)}, \text{otherwise} \end{cases} \quad (5.2)$$

where  $\text{Dist}_{min} = \min \left( \{\text{Dist}(\mathbf{f}_{y_i}^c, \mathbf{f}_j^c)\}_{j=1, j \neq y_i}^n \right)$ ,  $n$  is the total number of subjects, and  $\text{Dist}(\cdot)$  is a distance metric function. Fig. 5.3c and 5.3d show a 2D example of this process.

**Generated Graph** A vertex in a generated graph is simply represented by its corresponding representation vector  $\mathbf{f}_i$  extracted from  $E(\cdot)$ . And the entire node information in the graph is denoted by a feature matrix  $\mathbf{P}^f = \{\mathbf{f}_i\}_i^N \in \mathcal{R}^{N \times m}$  with each row corresponds to a vertex. For edges in a generated graph, we assign the adjacent value based on the similarity between two vertices. A vertex  $v_i$  is associated with another vertex  $v_j$  if  $v_j$  is one of  $v_i$ 's top  $k$  nearest neighbors, denoted by  $\text{Neib}_k(\mathbf{f}_j)$ , which is based on the distance of their feature vectors in the Euclidean space. Unlike  $\mathbf{A}^o$ , the final adjacency matrix of a generated graph may not be symmetric:  $\mathbf{A}_{ij}^f = \begin{cases} 1, \mathbf{f}_i \in \text{Neib}_k(\mathbf{f}_j) \\ 0, \text{otherwise} \end{cases}$ .

In the end, a pair of graphs  $(g^o(\mathbf{V}^o, \mathbf{E}^o), g^f(\mathbf{V}^f, \mathbf{E}^f))$  associated with the binary labels  $(y_i^g = 1, y_j^g = 0)$  are yielded by the graph constructor.

### 5.2.3 Discriminator and Adversarial Learning

For the graph discriminator  $D(\cdot)$ , we consider a whole graph with all its vertices and edges as a single instance, and the goal is to predict the category it belongs to. Since there is a variety of applications on graph classification, it has raised attentions to develop more useful and practical architectures for graph classification and representation learning. Here, we employ a fast approximate convolutions on graphs, proposed by [215]. In particular, the discriminator network consists of multiple graph convolutional layers connected by non-linear activation functions:

$$\mathbf{H}^{(l+1)} = \sigma \left( \tilde{\mathbf{D}}^{-\frac{1}{2}} \tilde{\mathbf{A}} \tilde{\mathbf{D}}^{-\frac{1}{2}} \mathbf{H}^{(l)} \mathbf{W}^{(l)} \right), \quad (5.3)$$

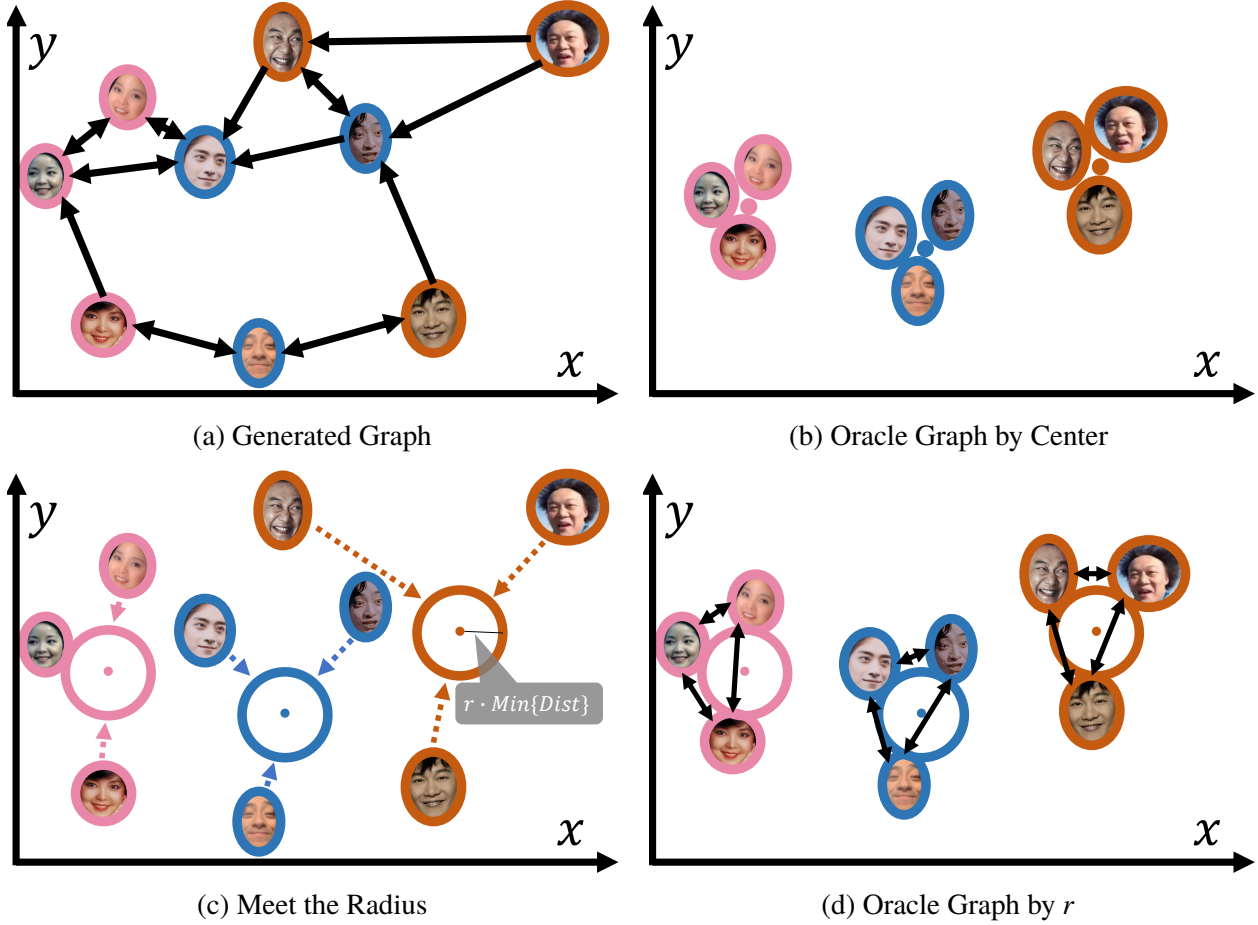


Figure 5.3 *Construction of generated graph and oracle graph.* In this example, the input image set comprises 9 images of 3 subjects, 3 images per subject. The image of each subject is surrounded by a circle with a unique color, indicating its identity. Each image is projected to a point in the 2D Euclidean feature space. The following graphs are constructed: (a) a generated graph, where each vertex  $v_i$  is represented by its feature vector, with a directed edge from  $v_i$  to  $v_j$  if  $v_j$  is one of the top 2 nearest neighbors of  $v_i$ ; (b) an oracle graph created by center points, where each vertex is represented by the mean vector of its identity with a bidirectional edge connecting two vertices of the same subject; (c) a radius constraint is used to allow tolerable intra-subject variations, where vertices move towards center directions (denoted by dashed arrows) to meet the radius requirement. For the vertex within the radius, the left most one in this example, it stays the same. (d) an oracle graph controlled by  $r$ , where the distance of each vertex to its center is reduced by the ratio of  $r$ , with a bidirectional edge connecting two vertices of the same subject.



where  $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}_N$  is the summation of the adjacency matrix and the identity matrix  $\mathbf{I}_N$ ,  $\tilde{\mathbf{D}}_{ii} = \sum_j \tilde{\mathbf{A}}_{ij}$ , and  $\mathbf{W}^{(l)}$  is the convolutional kernel matrix of the  $l^{th}$  layer.  $\sigma(\cdot)$  denotes the activation function.  $\mathbf{H}^{(l)}$  is the input feature map for layer  $l$ , and the input for the initial layer is the node information matrix  $\mathbf{P}$  of the graph. Finally, the classification network ends with a fully-connected layer followed by a sigmoid operation. As a binary classification task, the discriminator is trained by minimizing the standard binary cross entropy loss function:

$$\mathcal{L}_D = -\frac{1}{N_g} \sum_{i=1}^{N_g} (y_i^g \log(\mathbf{z}_i) + (1 - y_i^g) \log(1 - \mathbf{z}_i)), \quad (5.4)$$

where  $N_g$  is the total number of graphs, and  $D(g_i) = \mathbf{z}_i$  is the probability prediction by Sigmoid.

As a distribution detector,  $D(\cdot)$  is also in charge of guiding  $E(\cdot)$  to imitate features sampled from the target distribution. The parameters of  $E(\cdot)$  is adversarially trained by making  $D(g^f)$  to yield wrong predictions when  $g^f$  is the input, such that the Sigmoid output for  $g^f$  is as high as  $g^o$ . This adversarial loss is formulated as:

$$\mathcal{L}_a = -\frac{1}{N_g} \sum_{i=1}^{N_g} (\log D(g_i^f)), \quad (5.5)$$

where  $g_i^f = G(E(\{\mathbf{x}\}_i))$ . The gradients derived from  $\mathcal{L}_a$  will not propagate back to  $D(\cdot)$ , but only to  $E(\cdot)$  to update its parameters. By optimizing  $\mathcal{L}_a$ , we assume that if  $E(\cdot)$  successfully acquires the oracle distribution,  $D(\cdot)$  is supposed to consistently give high probability estimates for the category of oracle graphs, no matter what kind of graph is actually taken as the input. As a result, the face embeddings output by  $E(\cdot)$  are likely to present similar global and local node dependencies and similarities, as well as the recognition ability to the target features.

## 5.2.4 Network Training

As both oracle graphs and generated graphs are constructed based on features extracted from images, our framework requires a pre-trained face representation model to initialize the node

information matrix for both graphs. Apart from the above adversarial loss, a conventional loss for face recognition,  $\mathcal{L}_f$ , is also included to constrain the feature distribution in a reasonable physical metric where vector similarities can be applied, and to prevent trivial solutions too. Further, as stated in Sec. 5.2.2 that the maximum distance of an arbitrary point to its class centroid is based on the minimum distance between centroids, the distribution of centroids is also a key factor in creating discriminative node embeddings for oracle graphs. Thus, we introduce another objective to constrain the distance between centroids:

$$\mathcal{L}_c = \frac{2}{n(n-1)} \sum_{i \neq j}^{\frac{n(n-1)}{2}} [\overline{Dist}_c - Dist(\hat{\mathbf{f}}_i^c, \hat{\mathbf{f}}_j^c) + m]_+, \quad (5.6)$$

where  $\hat{\mathbf{f}}^c$  is the batch-estimated class centroid that is smoothly updated during training, and  $\overline{Dist}_c$  is the average distance of all pair-wise centroids among the entire training set.  $[x]_+ = \max(0, x)$ , and  $m$  is the margin parameter. In total, the training on the face embedding network is updated by minimizing the combined loss function:

$$\mathcal{L}_E = \mathcal{L}_f + \lambda \mathcal{L}_a + \mu \mathcal{L}_c, \quad (5.7)$$

where  $\lambda$  and  $\mu$  are used to adjust the contribution of the adversarial loss and the centroid distance loss to  $E(\cdot)$ .

**Training strategy** The proposed framework can be trained in an end-to-end manner, or a two-step strategy. If trained as one piece, the training procedure is similar to that of GAN, where  $E(\cdot)$  and  $D(\cdot)$  are updated alternately. On the other hand, to obtain a more stable information of the global data structure in the feature space, *e.g.*, the global class centroid vector  $\mathbf{f}^c$  and the average centroid distance  $\overline{Dist}_c$ , we adopt a stage-wise learning process to train the two networks.

Specifically, the discriminator  $D(\cdot)$  is first trained on the features pre-extracted by  $E(\cdot)$ . Meanwhile, all the parameters related to the global distribution is also pre-computed, including  $\mathbf{f}^c$  and  $\overline{Dist}_c$ . When  $D(\cdot)$  achieves a decent performance, we stop the training and fix its parameters. Next,

the training enters the second stage, where the pairwise centroid distance and the vertex features in the generated graph are updated with  $E(\cdot)$ . A training loop is completed after the second stage comes to an end. We can start another loop if necessary. Similar to triplet loss, we select hard samples during training. Those generated graphs with vertices or edges different from oracle graphs are considered as valid training samples. In addition, the three adjustable hyper-parameters, maximum distance ratio  $r$ , loss contributions  $\lambda$  and  $\mu$  can be updated during different training loops. For instance, we can gradually raise the difficulty of the target graph manipulation while the number of loops increases.

## 5.3 Experiments

In this section, we first conduct ablation experiments to study various design choices of our algorithm, and then compare our performance with the state of the art methods on public benchmark datasets. Finally, we analyze the distribution of our face representations via graph visualization.

### 5.3.1 Datasets and Implementation Details

Our training dataset is MS-Celeb-1M (MS1M) [23] cleaned by ArcFace [6], referred to as *MS1MV2*, containing about 5.8M images of 85K subjects. We evaluated our method on six public benchmark datasets for face recognition: LFW [27], CPLFW [71], CFP-FP [72], IJB-A [29], IJB-B [30], and IJB-C [9]. The face area is first cropped from each image based on five facial landmarks detected by RetinaFace [34], and then resized to  $112 \times 112$  pixels.

The architecture of  $E(\cdot)$  is a 100-layer ResNet used in [6]. For graph discriminator  $D(\cdot)$ , we adopt the DGCNN architecture proposed by [217], consisting of four layers of graph convolution [215] and Tanh activation, a sort pooling layer, MaxPooling and 1D convolution layers, a fully-connected layer with ReLU activation, and a Softmax classification layer in the end. For each graph, 5 subjects with 5 images per subject are randomly selected to form the set of nodes. During the training of  $D(\cdot)$ , the graph batch size is set to 90, of which half are oracle and half are generated. The parameters of  $D(\cdot)$

Table 5.1 Verification performance (%) of different vertex feature matrices of oracle graphs. A bigger  $r$  tolerates more intra-class variations, while a small  $r$ ,  $\mathbf{f}_i^c$ , or  $\mathbf{f}_i^p$  strive for minimal intra-class variation. A balance between the learning capability and ideal representations performs the best ( $r = 0.7$ ).

$\mathbf{P}^o$	CFP-FP	IJB-A
		TAR @ 0.1% FAR
$r = 1.0$	97.90	$94.42 \pm 1.49$
$r = 0.9$	98.27	$96.58 \pm 0.45$
$r = 0.7$	<b>98.34</b>	<b><math>97.31 \pm 0.38</math></b>
$r = 0.5$	96.68	$94.04 \pm 1.87$
$r = 0.3$	90.04	$78.13 \pm 3.98$
$\{\mathbf{f}_i^c\}_1^N$	92.71	$81.30 \pm 3.61$
$\{\mathbf{f}_i^p\}_1^N$	90.05	$77.42 \pm 4.19$

are updated using Adam with a learning rate of  $1 \times 10^{-3}$ . For training  $E(\cdot)$ , each batch contains 275 images from 55 subjects, also 5 images per subject. With the same graph size, 22 graphs are constructed by  $G(\cdot)$  in every training step.  $E(\cdot)$  is optimized by SGD with a momentum of 0.9 and a weight decay of  $5e - 4$ . The learning rate starts from 0.05 and drops at epoch 8, 15, 20. The margin  $m$  in loss function  $\mathcal{L}_c$  is set to 0.3. We utilize the loss function introduced in CurricularFace [221] as  $\mathcal{L}_f$ , and their ResNet100 model trained on MS1MV2 as the pre-trained model to initialize the vertex features in graphs. The entire training process takes two loops, and the three hyper-parameters,  $r = \{0.9, 0.7\}$ ,  $\lambda = \{1.0, 0.5\}$ ,  $\mu = \{0.5, 0.1\}$  in the two loops, respectively.

### 5.3.2 Ablation Study

**Vertex Feature Matrix of Oracle Graphs** The design for vertex feature matrix,  $\mathbf{P}^o$ , directly influences the feature distribution in oracle space, and also determines the complexity and feasibility for learning the embedding network. Here, we explore different ways to define oracle representations and analyze their impact on the learned feature space. Seven variants are considered: (1-5): Move each feature vector towards class centers under the control of different ratios  $r$ , in which  $r = 1.0, 0.9, 0.7, 0.5$ , and  $0.3$ , respectively. (6) Each identity is represented by a single feature vector  $\mathbf{f}_i^c$ , the mean vector of all samples from the subject  $i$ . This is equivalent to  $r = 0$ ; (7) Each identity is represented by a prototype feature vector  $\mathbf{f}_i^p$ , the corresponding column vector in the weight matrix

of the last classification layer; All these seven ablation models are trained for one loop, with  $\lambda = 1.0$  and  $\mu = 0.5$ .

Tab. 5.1 reports the face verification results on CFP-FP and IJB-A using different vertex feature matrices for oracle graphs. Our results suggest that there is a limit of the intra-class distance of each identity in oracle graphs. If the predefined intra-class distance is smaller than the limit, it is beyond the network learning capacity and thus its recognition performance will significantly degrade. For example, when we appoint center representation  $\{\mathbf{f}_i^c\}_1^N$ , or prototype representation  $\{\mathbf{f}_i^p\}_1^N$  as the vertex feature matrix for oracle graph, the average intra-class distance is zero, since every identity is a perfect 512-dimensional feature representation. The verification accuracy yet drops 8.29% on CFP-FP compared to the best model in this ablation, and the True Acceptance Rate (TAR) falls from  $97.31\% \pm 0.38\%$  to  $77.42\% \pm 4.19\%$  on IJB-A. A similar performance is also observed when  $r$  is set to 0.3.

These results are even worse than the initial pre-trained network. This indicates that graph discriminator would deliver a jumble of information that may misguide the embedding network, when an unreasonably ideal distribution is assumed for oracle representations. On the other hand, the performance is relatively insensitive to  $r$  when the minimum intra-class variation is within a reasonable range. By definition, a bigger  $r$  tolerates more intra-class variations, but leaving less room for improving the representation. Thus,  $r$  should be small enough to achieve higher discriminability of the representation, and meanwhile, be bigger enough to prevent capacity overflow. In our experiments,  $r = 0.7$  appears to be a good trade-off and consistently performs the best on both CFP and IJB-A.

**Adjacency Matrix of Generated Graphs** In Sec. 5.2.2, we mention that the adjacency matrix of a generated graph is created by the information of nearest neighbors. The goal is for  $E(\cdot)$  to learn how to map the oracle dependencies between vertices in the Euclidean space. To show its efficacy, we ablate by replacing it with the adjacency matrix of the oracle graph, which is established based on identity labels. Both the ablation model and the proposed model are trained for one loop, with  $r = 0.7$ ,  $\lambda = 1.0$ , and  $\mu = 0.5$ .

Table 5.2 Verification performance (%) of different adjacency matrices of generated graphs.

$A^f$	CFP-FP	IJB-A TAR @ 0.1% FAR
Nearest Neighbors	<b>98.34</b>	<b>97.31 <math>\pm</math> 0.38</b>
Identity Labels	97.16	94.44 $\pm$ 1.09

Table 5.3 Verification performance (%) of different  $\lambda$  and  $\mu$ .

$\lambda$	$\mu$	CFP-FP	IJB-A TAR @ 0.1% FAR
1.0	0.5	<b>98.34</b>	<b>97.31 <math>\pm</math> 0.38</b>
1.0	0.1	98.27	96.20 $\pm$ 0.40
0.5	0.5	98.14	95.82 $\pm$ 0.47

Tab. 5.2 compares the results of training using different ways to define adjacency matrices for generated graphs. Clearly it is more important to allow adjacency matrices depending on the feature representations, which helps gradient flows through the embedding network, and the adjacency matrices will update along with the embeddings. Otherwise, if defined via identity labels, constant adjacency matrices will be utilized during training iterations.

**Contribution of Adversarial Loss and Centroid Loss** Here we show the effects of adversarial loss and centroid loss by training the network using different  $\lambda$  and  $\mu$ . The remaining settings are the same for all the models trained in this ablation: one loop, with  $r = 0.7$ . Tab. 5.3 reports the results for different hyper-parameters,  $\lambda$  and  $\mu$  to show the contributions of both objective functions. We keep one of them unchanged, and decrease the value of the other to see how it affects the performance when less contribution is made by  $\mathcal{L}_c$  or  $\mathcal{L}_a$ . As shown in Tab. 5.3, a smaller  $\lambda$  and  $\mu$  both lead to worse performance on CFP-FP and IJB-A, while increasing either of them boosts the performance. This indicates that the proposed adversarial learning with the centroid distance loss makes non-negligible contribution to the discriminability of face presentations.

### 5.3.3 Comparisons with SOTA Methods

We choose six benchmark datasets widely used for face recognition, to thoroughly evaluate our approach and compare it with other SOTA methods as well. Among the six datasets, three of them

Table 5.4 Verification accuracy (%) of our model and SOTA methods on LFW, CPLFW, and CFP-FP. The results marked by (\*) are re-implemented by ArcFace [6]. All other baseline results are reported by their respective papers. [Keys: Red: Best, Blue: Second best]

Method	Training Data	LFW	CPLFW	CFP-FP
DeepID2+ [222]	0.3M	99.47	–	–
Center Loss [47]	0.7M	99.28	–	–
SphereFace [13]	CASIA [11] (0.5M)	99.42	–	–
DeepFace [17]	4M	97.35	–	–
FaceNet [12]	200M	99.63	–	–
TPE [223]	CASIA	–	–	89.17
DRGAN [224]	1M	–	–	93.41
Yin <i>et al.</i> [11]	CASIA	–	–	94.39
UVGAN [225]	MS1M (10M)	99.60	–	94.05
CosFace [19]	CASIA	99.51*	–	95.44*
PFE [184]	MS1M (4.4M)	99.82	–	93.34
ArcFace [6]	MS1MV2 (5.8M)	99.83	–	98.37
CurricularFace [221]	MS1MV2	99.80	93.13	98.37
DDL [226]	VGGFace2 (3.3M)	99.68	93.43	98.53
Ours	MS1MV2	99.78	93.40	98.41

are tested under the instance-based image-to-image verification protocol, verifying whether a pair of face images belong to the same person; and the verification protocol of the other three datasets is based on image templates. A template of images is referred as a collection of face images sampled from the same identity. The template-based verification task requires us to decide whether two templates are from the same person or not.

**Instance-based Face Verification** Tab. 5.4 reports the verification accuracy on the three instance-based benchmark datasets, LFW, CPLFW, and CFP-FP. LFW is a dataset collected before the era of deep face representations. Its 6,000 pairs of face images are considered as semi-constrained, with limited intra-class variations and relatively high image quality. The SOTA performance is already saturated on LFW. Almost all DNN-based methods listed in Tab. 5.4 achieve over 99.00% accuracy. Even so, LFW is still used as a standard validation benchmark given its prevalence in FR research communities and efficient pairwise assessment. Our model obtains a similar accuracy (99.78%) to other methods, though slightly worse than the two models both trained on MS1MV2 (ArcFace [6] and CurricularFace [221]).

The other two datasets, CPLFW, and CFP-FP, are created to address the challenge of large facial

Table 5.5 Comparisons of verification performance with SOTA methods on IJB-A, IJB-B, and IJB-C. The evaluation is measured by TAR (%), True Acceptance Rate, at a certain FAR, False Acceptance Rate. For IJB-A, FAR = 0.1%; for IJB-B and IJB-C, FAR = 0.01%. The decimal precision of TAR varies among those reported by SOTA methods. Results reported in this table are unified to one decimal place (0.1). All baseline results are reported by their respective papers. [Keys: Red: Best, Blue: Second best]

Method	Training Data	IJB-A	IJB-B	IJB-C
DRGAN [224]	1M	53.9 $\pm$ 4.3	–	–
Yin <i>et al.</i> [11]	CASIA	73.9 $\pm$ 4.2	–	–
NAN [227]	3M	88.1 $\pm$ 1.1	–	–
QAN [228]	5M	89.3 $\pm$ 3.9	–	–
TPE [223]	CASIA	90.0 $\pm$ 1.0	–	–
VGGFace2 [59]	VGGFace2	90.4 $\pm$ 2.0	80.0	84.0
Multicolumn [183]	VGGFace2	92.0 $\pm$ 1.3	83.1	88.7
DCN [125]	VGGFace2	–	84.9	88.5
PFE [184]	MS1M (4.4M)	95.3 $\pm$ 0.9	–	93.3
Adacos [229]	2.8M	–	–	92.4
P2sgad [230]	2.8M	–	–	92.3
ArcFace [6]	MS1MV2	–	94.2	95.6
CurricularFace [221]	MS1MV2	–	94.8	96.1
DDL [226]	VGGFace2	–	90.7	93.1
Ours	MS1MV2	97.3 $\pm$ 0.4	94.6	96.2

pose variations. The verification is conducted between a frontal face and a profile face, or two faces with variant yaw angles. Despite more challenging than LFW, images in CFP-FP are of high resolution. And for CPLFW, it contains the same images as LFW, but with re-designated face pairs with pose difference. Thus, the average SOTA performance on these two datasets is over 90.00% accuracy. Without a particular policy tailored for large pose variations, our approach still achieves top performance, being better than all of the SOTA methods except DDL [226], which is trained on VGGFace2 [59], a dataset with large pose variations that exhibits less domain gap with the two testing sets compared to MS1MV2.

**Template-based Face Verification** Tab. 5.5 reports the TAR performance on the three template-based benchmark datasets, IJB-A, IJB-B, and IJB-C. To evaluate FR models in more challenging scenes, NIST released a series of datasets that contain a mix of high/low quality images and low quality video frames, presenting large variations in pose, illumination, occlusion, resolution, *etc.* IJB-A is among the first to be published and has the smallest number of subjects and images. Our model outperforms the other methods on IJB-A using no specific scheme for template-based face



Table 5.6 Comparisons of face identification performance (%) on the IJB-C dataset (close-set).

Method	IJB-C	
	Rank-1	Rank-5
VGGFace2 [59]	91.4	95.1
CurricularFace [221]	94.4	96.1
Ours	<b>95.3</b>	<b>96.7</b>

verification. It should be noted that since the evaluation on IJB-A is a ten-fold cross-validation protocol, fine-tuning can be done on the split folds before evaluation. No fine-tuning has yet been conducted on our model.

Both IJB-B and IJB-C are the extended versions of IJB-A by adding more subjects and images. The  $n$ -fold cross-validation protocol is not provided for these two extensions and only evaluation is allowed. In Tab. 5.5, we see that our approach performs comparably to the SOTA methods on IJB-B and IJB-C, which demonstrates that our graph based adversarial framework indeed benefits the discriminability and generalizability for face representation.

Apart from verification tasks, we also report the face identification results on IJB-C in Tab. 5.6. The close-set identification protocol of IJB-C contains two sets of face templates with no overlapping images, referred to as probe templates, and gallery templates, respectively. In particular, the set of gallery templates include all identities in probe templates, one template per identity. Each time, one template in the probe set is compared with all the gallery templates to search for the nearest matches. The final result is presented by a *Rank-k* accuracy, the correct matching rate among the top  $k$  nearest neighbors. We report Rank-1 and Rank-5 accuracies in Tab. 5.6. Compared with CurricularFace and the algorithm in [59], our model boosts the performance in both Rank-1 and Rank-5 accuracies, which shows that the proposed method is also effective in improving face representations for identification tasks.

### 5.3.4 Analysis on Feature Distribution

We further investigate the effects of our graph-based adversarial learning mechanism on face representations via visualization. We discuss the enhancement of feature discriminativeness from

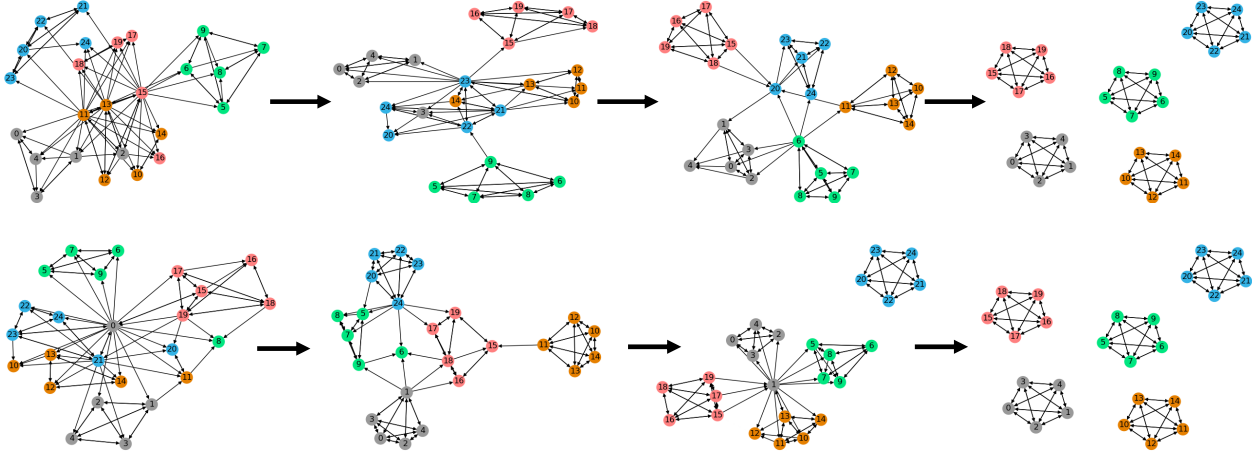


Figure 5.4 Two examples of generated graphs being updated by adversarial learning at 4 instances during the training process.

the graph perspective. Since the vertex features of our graph are initialized by a well-trained model, it lays a good foundation of further improvement. In fact, many feature points are within the radius constraint and are well connected with their same-identity neighbors. For these *good points*, they need no attentions from training. On the contrary, these points may confuse the graph discriminator  $D(\cdot)$  as their features and structures are the same as oracle graphs, which leads to bad results. Hence, our adversarial training only focuses on those *hard samples* that present differently from oracle graphs, *i.e.*, either violating the radius requirement or not connecting to the same identity. We ask the network to learn a good mapping for these samples to boost the general performance. For example, Fig. 5.4 shows the training progress of two sets of initially hard samples made by the proposed adversarial learning.

Fig. 5.5 shows the t-SNE visualization of the face embeddings extracted from CurricularFace as well as the proposed method. First, We randomly select 15 hard sample subjects with all of their face images available in MS1MV2, our training dataset. Next, a 512-dim feature vector is extracted from each image by our model and the initial CurricularFace model. We then use t-SNE to project each 512-dim vector into a 2D space, in which the geodesic distances between points in the high dimensional space are maximally maintained. The color of each 2D point stands for the subject's identity. From Fig. 5.5, we see that these samples images are hard to be recognized by CurricularFace because many of the nearest neighbors are points of other identities. In comparison, by learning

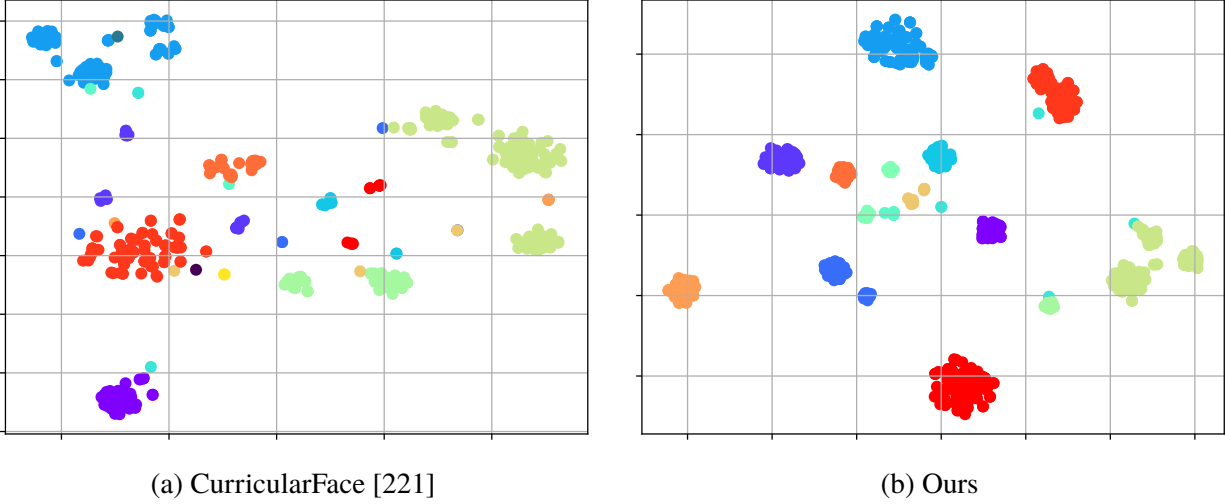


Figure 5.5 t-SNE visualization of the face representations in a 2D space. Each identity is represented by a unique color. The initial face representations extract from CurricularFace, and the updated representations learned via adversarial graph classification are shown in (a) and (b), respectively.

from oracle graphs, the proposed method learns a better feature space where the face embeddings of the same identity form a more compact clustering, leading to a higher discriminability.

## 5.4 Concluding Remarks

Deep network based face recognition has witnessed rapid development in recent years, especially owing to the innovation of a series of loss functions designed to enhance the discriminativeness of the embedding space. However, most loss functions examine each individual sample, a sample pair, or at most a triplet of samples in the physical distance metric, but ignore the general *distribution* derived from correlations between samples of within-class and cross-class. In contrast, motivated by the design of generative adversarial network, our proposed approach oversees the distribution of feature vectors, represented by a graph, and push the generated graph to be similar to the “idealized” oracle graph, by updating the embedding network.

We believe this work is a meaningful and exciting exploration along the direction of loss function design in the face recognition community. Experimental results demonstrate the promising of our proposed approach. Since our main idea is not face specific, one of the future works is to extend this study to ImageNet-like general classification problems.

# Chapter 6

## Summary and Future Work

This dissertation addresses four face recognition problems, which are essential to both fundamental analysis and practical applications. Solutions proposed in this dissertation employ a variety of techniques in deep learning to advance research within the FR community.

**Intrinsic Dimensionality.** One of the main goals is to *estimate the limit of representation compactness with no loss in recognition performance*. This is denoted as the intrinsic dimensionality (IND) of a face representation. Meanwhile, given the intrinsic dimensionality, we also aim at finding a projection method to obtain a representation towards its limit of compactness.

The density variation in a representation is the principal consideration when estimating reliable IND of a given face representation. However, it is difficult to obtain an accurate estimate of such probability distribution since face images often lie on a topologically complex curved manifold, and estimating its distribution requires data points at very small length-scales (distances), which are hard to access when data is limited, especially in high-dimensional spaces where deep face representations are usually embedded. Another challenging task is to verify whether a given estimate of IND truly represents the dimensionality of the complex high-dimensional representation space.

Our contribution to this problem is that we overcome the above challenges by offering a topological dimensionality estimation technique for high-dimensional face representations and proposing a mapping approach that enables validation of the IND estimates through image matching

experiments on the corresponding low-dimensional intrinsic representation of feature vectors. In this study, we define the notion of intrinsic dimension through the classical concept of topological dimension of the support of a distribution. We adopt an elegant solution to addressing the issue of *curse of dimensionality* by utilizing the geodesic distance between points that is computed as graph induced shortest path between points instead of the Euclidean distance. And the difficulty of estimating the data distribution is conquered based on the observation that different topological geometries are similar to each other as long as the intrinsic dimensionality is the same, or in other words the distribution depends only on the intrinsic dimensionality and not on the geometric support of the manifolds. Thus, the intrinsic dimensionality of face manifold can be estimated by comparing the empirical distribution of the pairwise distances on the manifold to that of a known distribution, such as the  $m$ -hypersphere. To validate the estimated IND, we propose a method, DeepMDS, that relies on the ability of DNNs to approximate the complex mapping function from the ambient space to the intrinsic space. The DeepMDS model is optimized to preserve the interpoint geodesic distances between the feature vectors in the ambient and intrinsic space, and is trained in a stage-wise manner that progressively reduces the dimensionality of the representation. Our new dimensionality reduction method addresses the scalability and out-of-sample-extension problems suffered by traditional spectral methods like Isomap. A well-trained DeepMDS model is not limited to map observed data, but can be easily applied to new test data since it provides a mapping function in the form of a feed-forward network that maps the ambient feature vector to its corresponding intrinsic feature vector. It is shown that DeepMDS mapping is significantly better than other dimensionality reduction approaches in terms of its discriminative capability.

**Capacity.** *Given a face representation, how many identities can it resolve?* Addressing this question is the primary goal of this work. We also refer to it as the capacity of a given face representation. We define the maximal number of users at which the face representation reaches its limit as the capacity of the representation. By our definition, the capacity is determined in an objective manner without the need for empirical evaluation.

Our solution relies on the notion of capacity that has been well studied in the information

theory community in the context of wireless communication. The setting, commonly referred to as the Gaussian channel, consists of a source signal that is additively corrupted by Gaussian noise to generate observations. The capacity of this Gaussian channel is defined as the number of distinct source signals in the signal representations. Despite the rich theoretical understanding of the capacity of a Gaussian channel, there has been limited practical application of this theory in the context of estimating the capacity of learned embeddings like face representations. For example, estimating the distribution of the source and the noise for a high-dimensional embedding, such as a face representation, is an open problem. A variety of sources of noise need to be taken into account when it comes to reliably inferring the probability distributions in high-dimensional spaces. It is also challenging to obtain reliable estimates of the volume of arbitrarily shaped high-dimensional manifolds (for capacity bound).

We address the aforementioned challenges to obtain reliable estimates of the capacity of any face representation by leveraging the dimensionality reduction method, DeepMDS, we proposed in the previous study. With the assistance of DeepMDS, we first model the face representation as a low-dimensional Euclidean manifold embedded within a high-dimensional space, and then project and unfold the manifold to a low-dimensional space. In our solution, two kinds of manifolds need to be approximated: (1) a population manifold that is approximated by a multivariate Gaussian distribution (equivalently, hyper-ellipsoidal support) in the unfolded low-dimensional space; (2) identity-specific manifolds that are approximated by the corresponding multi-variate Gaussian distributions whose supports are estimated as a function of the specified FAR. The final capacity value is estimated as a ratio of the volumes of the population and identity-specific hyper-ellipsoids. We estimate the distribution of both kinds of manifolds from an observed face representation by leveraging the recent advances in DNNs. In particular, given an embedding function (*teacher network*) that maps normalized high-dimensional face images to a low-dimensional vector, we train a DNN (*student network*) to model two sources of uncertainty that contribute to the noise in the embeddings: (i) uncertainty in the data, and (ii) uncertainty in the embedding function. These uncertainty estimates are then used to determine the volumes of their manifolds and also directly

enable us to compute the representation capacity as a function of the desired operating point as determined by its corresponding FAR. Experimental results suggest that our capacity estimates are an upper bound on the actual performance of face recognition systems in practice, especially under unconstrained scenarios. The relative order of the capacity estimates mimics the relative order of the verification accuracy on the benchmark datasets.

**Bias.** Demographic bias in face recognition systems can potentially cause ethical issues when deployed. A thorough analysis on the discriminatory behavior of FR systems against certain demographic groups and a solution to mitigating such bias are the central aims of this study. We define FR bias as the uneven recognition performance w.r.t. demographic groups. And, the ultimate goal of unbiased face recognition is that, given a face recognition system, there should be no statistically significant difference among the performance in different demographic groups of face images.

Bias can be derived from various sources, such as the balance degree of the demographic samples, variations in capture conditions and image noise among different demographic groups. Therefore, naively training on a dataset containing uniform samples over the group space may still lead to bias. In fact, the demographic distribution of a dataset is often imbalanced with underrepresented and overrepresented groups. Similarly, simply re-sampling an imbalanced training dataset may not solve the problem either, given the fact that the diversity of latent variables is different across groups and the instances cannot be treated fairly during training. For these reasons, bias mitigation requires special attentions on both data sampling and algorithm design.

This thesis focuses on developing de-biasing algorithms for face recognition. Our contribution to this problem is that we provide two approaches to addressing the issue of demographic bias. The first framework, DebFace, is to diminish the influence of bias on both face recognition and demographic attribute estimation. In fact, during our investigation on demographic bias in face recognition, we also observe the biased performance of demographic attribute estimation, which may cause additional bias since it acts differently when applied to de-bias face representations in different groups. To this end, we propose to jointly learn unbiased representations for both

the identity and demographic attributes. This solution is based on the assumption that if the face representation does not carry discriminative information of demographic attributes, it would be unbiased in terms of demographics. And the same hypothesis is applied to demographic attribute estimation as well. Starting from a multi-task learning framework that learns disentangled feature representations of gender, age, race, and identity, respectively, we request the classifier of each task to act as adversarial supervision for the other tasks. These four classifiers help each other to achieve better feature disentanglement, resulting in unbiased feature representations for both the identity and demographic attributes.

Although DebFace shows noticeable effect in mitigating demographic bias, we observe that the recognition performance declines as well. Thus, in our second solution, GAC, we mainly focus on racial bias and strive to enhance the discriminativeness of face representations in every race/ethnicity group. The key idea of GAC is to give the network more capacity to broaden its scope for multiple face patterns from different groups since an unbiased FR model shall rely on both unique patterns for recognition of different groups, and general patterns of all faces for improved generalizability. GAC explicitly learns these different feature patterns by leveraging two modules: the adaptive layer and automation module. The adaptive layer comprises adaptive convolution kernels and channel-wise attention maps where each kernel and map tackle faces in *one* demographic group. We also introduce a new objective function to GAC, which diminishes the variation of average intra-class distance between demographic groups. To dynamically apply these adaptive modules, we also propose an automation scheme that can choose which layers to apply the adaptations. As a result, our experiments demonstrate the efficacy of GAC in bias mitigation and SOTA performance preservation.

**Representation Learning.** We focused on the fairness of pre-defined groups (demographic groups) in the previous study. Yet, it is equally important to address the individual fairness and performance in face recognition. In this study, we aim at developing a face representation method in consideration of individual performance so that face images of each identity is fairly treated, and in the mean time the overall performance is improved.



By investigating the average intra-subject and inter-subject distance of each identity, we start by constructing a  $k$ -NN graph to describe the general face distribution in the embedding space. As a further endeavor, we propose a representation learning method that utilizes graph classification via adversarial training. In contrast to using an equational metric as the constraint in the feature space, our idea engages a creation of an ideal feature space that ensures individual performance, referred to as *oracle space*, where the cluster of feature points in each class is clearly separated from other classes. A deep neural network (DNN) is then trained to generate face features that follow the data distribution in the *oracle space*. Since the relationships and inter-dependencies between feature points are not as simple as the data structure of fixed-size grid images, we can no longer use the conventional CNN *discriminator* as the form of our adversarial supervision. In particular, the data structure in the feature space can be represented by a directed graph, where each vertex corresponds to an image sample and the edges between vertices represent their dependencies. For the oracle space, feature points are connected if they belong to the subject; while in the actual feature space, nodes are linked to their  $k$  nearest neighbors. The discrimination task here is to distinguish between graphs from the oracle space and the generated space. Hence, we employ a graph classifier trained with Graph Neural Network (GNN), as the discriminator that guides the representation model to output features that follow the oracle distribution. It is demonstrated that our framework is capable of learning a generic feature space with enhanced discriminative power for face images, based on a predesigned feature distribution defined on a graph structure.

**Future Work.** While this dissertation has explored fundamental problems in face recognition and has developed useful tools and algorithms that provide excellent performance, there is always room for additional improvement. Most importantly, the research contributions in this dissertation is not limited to face recognition. There are a number of other areas in computer vision, for example, general image classification and representation learning, that can benefit greatly from the research conducted in this dissertation.

## APPENDIX

## PUBLICATIONS

- [1] S. Gong, V. N. Boddeti, and A. K. Jain, “On the intrinsic dimensionality of image representations,” in *CVPR*, 2019.
- [2] S. Gong, X. Liu, and A. K. Jain, “Jointly de-biasing face recognition and demographic attribute estimation,” *ECCV*, 2020.
- [3] D. Deb, S. Wiper, S. Gong, Y. Shi, C. Tymoszek, A. Fletcher, and A. K. Jain, “Face recognition: Primates in the wild,” in *BTAS*, IEEE, 2018.
- [4] S. Gong, Y. Shi, N. D. Kalka, and A. K. Jain, “Video face recognition: Component-wise feature aggregation network (c-fan),” in *ICB*, IEEE, 2019.
- [5] S. Gong, Y. Shi, and A. Jain, “Low quality video face recognition: Multi-mode aggregation recurrent network (marn),” in *ICCVW*, 2019.
- [6] S. Gong, X. Liu, and A. K. Jain, “Mitigating face recognition bias via group adaptive classifier,” in *CVPR*, 2021.
- [7] S. Gong, V. N. Boddeti, and A. K. Jain, “On the capacity of face representation,” *arXiv preprint arXiv:1709.10433*, 2017.

## BIBLIOGRAPHY

## BIBLIOGRAPHY

- [1] D. Granata and V. Carnevale, “Accurate estimation of the intrinsic dimension using graph distances: Unraveling the geometric complexity of datasets,” *Scientific Reports*, vol. 6, p. 31377, 2016.
- [2] K. W. Pettis, T. A. Bailey, A. K. Jain, and R. C. Dubes, “An intrinsic dimensionality estimator from near-neighbor information,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 25–37, 1979.
- [3] A. Rozza, G. Lombardi, C. Ceruti, E. Casiraghi, and P. Campadelli, “Novel high intrinsic dimensionality estimators,” *Machine Learning*, vol. 89, no. 1-2, pp. 37–65, 2012.
- [4] M. Wang, W. Deng, J. Hu, X. Tao, and Y. Huang, “Racial faces in the wild: Reducing racial bias by information maximization adaptation network,” in *ICCV*, 2019.
- [5] M. Wang and W. Deng, “Mitigating bias in face recognition using skewness-aware reinforcement learning,” in *CVPR*, 2020.
- [6] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, “Arcface: Additive angular margin loss for deep face recognition,” in *CVPR*, 2019.
- [7] P. Grother, M. Ngan, and K. Hanaoka, “Face recognition vendor test (FRVT) part 3: Demographic effects,” in *Technical Report, National Institute of Standards and Technology*, 2019.
- [8] S. Gong, V. N. Boddeti, and A. K. Jain, “On the intrinsic dimensionality of image representations,” in *CVPR*, pp. 3987–3996, 2019.
- [9] B. Maze, J. Adams, J. A. Duncan, N. Kalka, T. Miller, C. Otto, A. K. Jain, W. T. Niggel, J. Anderson, J. Cheney, *et al.*, “Iarpa janus benchmark-c: Face dataset and protocol,” in *ICB*, 2018.
- [10] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, “Inception-v4, inception-resnet and the impact of residual connections on learning,” in *AAAI*, 2017.
- [11] D. Yi, Z. Lei, S. Liao, and S. Z. Li, “Learning face representation from scratch,” *arXiv preprint arXiv:1411.7923*, 2014.
- [12] F. Schroff, D. Kalenichenko, and J. Philbin, “Facenet: A unified embedding for face recognition and clustering,” in *CVPR*, 2015.
- [13] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, and L. Song, “Sphereface: Deep hypersphere embedding for face recognition,” in *CVPR*, 2017.
- [14] B. F. Klare, M. J. Burge, J. C. Klontz, W. V. Bruegge, Richard, and A. K. Jain, “Face recognition performance: Role of demographic information,” *IEEE Trans. Information Forensics and Security*, vol. 7, no. 6, pp. 1789–1801, 2012.

- [15] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, “Grad-cam: Visual explanations from deep networks via gradient-based localization,” in *ICCV*, 2017.
- [16] H.-T. Nguyen, *Contributions to facial feature extraction for face recognition*. PhD thesis, Grenoble, 2014.
- [17] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, “Deepface: Closing the gap to human-level performance in face verification,” in *CVPR*, 2014.
- [18] Y. Sun, Y. Chen, X. Wang, and X. Tang, “Deep learning face representation by joint identification-verification,” in *NeurIPS*, 2014.
- [19] H. Wang, Y. Wang, Z. Zhou, X. Ji, D. Gong, J. Zhou, Z. Li, and W. Liu, “Cosface: Large margin cosine loss for deep face recognition,” in *CVPR*, 2018.
- [20] C. Lu and X. Tang, “Surpassing human-level face verification performance on lfw with gaussianface,” *arXiv preprint arXiv:1404.3840*, 2014.
- [21] J. Howard, Y. Sirotnin, and A. Vemury, “The effect of broad and specific demographic homogeneity on the imposter distributions and false match rates in face recognition algorithm performance,” in *IEEE BTAS*, 2019.
- [22] E. Creager, D. Madras, J.-H. Jacobsen, M. Weis, K. Swersky, T. Pitassi, and R. Zemel, “Flexibly fair representation learning by disentanglement,” in *ICML*, 2019.
- [23] Y. Guo, L. Zhang, Y. Hu, X. He, and J. Gao, “Ms-celeb-1m: A dataset and benchmark for large-scale face recognition,” in *ECCV*, Springer, 2016.
- [24] I. Masi, Y. Wu, T. Hassner, and P. Natarajan, “Deep face recognition: A survey,” in *SIBGRAPI*, 2018.
- [25] C. Jin, R. Jin, K. Chen, and Y. Dou, “A community detection approach to cleaning extremely large face database,” *Computational Intelligence and Neuroscience*, 2018.
- [26] “[https://github.com/inlmouse/ms-celeb-1m\\_washlist](https://github.com/inlmouse/ms-celeb-1m_washlist),”
- [27] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller, “Labeled faces in the wild: A database for studying face recognition in unconstrained environments,” Tech. Rep. 07-49, University of Massachusetts, Amherst, October 2007.
- [28] P. Viola and M. J. Jones, “Robust real-time face detection,” *International journal of computer vision*, 2004.
- [29] B. F. Klare, B. Klein, E. Taborisky, A. Blanton, J. Cheney, K. Allen, P. Grother, A. Mah, and A. K. Jain, “Pushing the frontiers of unconstrained face detection and recognition: Iarpa janus benchmark a,” in *CVPR*, 2015.
- [30] C. Whitelam, E. Taborisky, A. Blanton, B. Maze, J. Adams, T. Miller, N. Kalka, A. K. Jain, J. A. Duncan, K. Allen, *et al.*, “Iarpa janus benchmark-b face dataset,” in *CVPRW*, 2017.

- [31] S. A. Rizvi, P. J. Phillips, and H. Moon, “The feret verification testing protocol for face recognition algorithms,” in *AFGR*, pp. 48–53, IEEE, 1998.
- [32] M. Gunther, S. Cruz, E. M. Rudd, and T. E. Boult, “Toward open-set face recognition,” in *CVPRW*, pp. 71–80, 2017.
- [33] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, “Joint face detection and alignment using multitask cascaded convolutional networks,” *IEEE Signal Processing Letters*, 2016.
- [34] J. Deng, J. Guo, Y. Zhou, J. Yu, I. Kotsia, and S. Zafeiriou, “Retinaface: Single-stage dense face localisation in the wild,” *arXiv preprint arXiv:1905.00641*, 2019.
- [35] A. K. Jain, K. Nandakumar, and A. Ross, “50 years of biometric research: Accomplishments, challenges, and opportunities,” *Pattern Recognition Letters*, vol. 79, pp. 80–105, 2016.
- [36] M. A. Turk and A. P. Pentland, “Face recognition using eigenfaces,” in *CVPR*, 1991.
- [37] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, “Eigenfaces vs. fisherfaces: Recognition using class specific linear projection,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 711–720, 1997.
- [38] D. Gabor, “Theory of communication. part 1: The analysis of information,” *Journal of the Institution of Electrical Engineers-Part III: Radio and Communication Engineering*, vol. 93, no. 26, pp. 429–441, 1946.
- [39] L. Wiskott, N. Krüger, N. Kuiger, and C. Von Der Malsburg, “Face recognition by elastic bunch graph matching,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 775–779, 1997.
- [40] T. Ahonen, A. Hadid, and M. Pietikäinen, “Face recognition with local binary patterns,” in *ECCV*, 2004.
- [41] T. Ahonen, E. Rahtu, V. Ojansivu, and J. Heikkilä, “Recognition of blurred faces using local phase quantization,” in *ICPR*, IEEE, 2008.
- [42] A. Albiol, D. Monzo, A. Martin, J. Sastre, and A. Albiol, “Face recognition using hog–ebgm,” *Pattern Recognition Letters*, vol. 29, no. 10, pp. 1537–1543, 2008.
- [43] M. Bicego, A. Lagorio, E. Grosso, and M. Tistarelli, “On the use of sift features for face authentication,” in *CVPRW*, IEEE, 2006.
- [44] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *NeurIPS*, 2012.
- [45] P. Grother, P. Grother, M. Ngan, and K. Hanaoka, *Face Recognition Vendor Test (FRVT) Part 2: Identification*. US Department of Commerce, National Institute of Standards and Technology, 2019.
- [46] Y. Sun, X. Wang, and X. Tang, “Deep learning face representation from predicting 10,000 classes,” in *CVPR*, pp. 1891–1898, 2014.

- [47] Y. Wen, K. Zhang, Z. Li, and Y. Qiao, “A discriminative feature learning approach for deep face recognition,” in *ECCV*, 2016.
- [48] R. Ranjan, C. D. Castillo, and R. Chellappa, “L2-constrained softmax loss for discriminative face verification,” *arXiv preprint arXiv:1703.09507*, 2017.
- [49] W. Liu, Y. Wen, Z. Yu, and M. Yang, “Large-margin softmax loss for convolutional neural networks,” in *ICML*, p. 7, 2016.
- [50] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *CVPR*, 2016.
- [51] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *arXiv preprint arXiv:1409.1556*, 2014.
- [52] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, “Web-scale training for face identification,” in *CVPR*, pp. 2746–2754, 2015.
- [53] R. S. Bennett, “Representation and analysis of signals part xxi. the intrinsic dimensionality of signal collections,” tech. rep., Johns Hopkins University Baltimore MD, Department of Electrical Engineering and Computer Science, 1965.
- [54] J. Theiler, “Estimating fractal dimension,” *JOSA A*, vol. 7, no. 6, pp. 1055–1073, 1990.
- [55] J. A. Costa and A. O. Hero, “Geodesic entropic graphs for dimension and entropy estimation in manifold learning,” *IEEE Transactions on Signal Processing*, vol. 52, no. 8, pp. 2210–2221, 2004.
- [56] V. N. Boddeti, “Secure face matching using fully homomorphic encryption,” in *IEEE International Conference on Biometrics: Theory, Applications, and Systems (BTAS)*, 2018.
- [57] A. Talwalkar, S. Kumar, and H. Rowley, “Large-scale manifold learning,” in *CVPR*, 2008.
- [58] T. G. Dietterich and E. B. Kong, “Machine learning bias, statistical bias, and statistical variance of decision tree algorithms,” tech. rep., Department of Computer Science, Oregon State University, 1995.
- [59] Q. Cao, L. Shen, W. Xie, O. M. Parkhi, and A. Zisserman, “VGGface2: A dataset for recognising faces across pose and age,” in *FRGC*, IEEE, 2018.
- [60] T. Bolukbasi, K.-W. Chang, J. Y. Zou, V. Saligrama, and A. T. Kalai, “Man is to computer programmer as woman is to homemaker? debiasing word embeddings,” in *NeurIPS*, 2016.
- [61] A. Torralba, A. A. Efros, *et al.*, “Unbiased look at dataset bias,” in *CVPR*, 2011.
- [62] C. Drummond, R. C. Holte, *et al.*, “C4. 5, class imbalance, and cost sensitivity: why under-sampling beats over-sampling,” in *Workshop on Learning from Imbalanced Datasets II*, Citeseer, 2003.



- [63] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, “Smote: synthetic minority over-sampling technique,” *Journal of Artificial Intelligence research*, vol. 16, pp. 321–357, 2002.
- [64] S. S. Mullick, S. Datta, and S. Das, “Generative adversarial minority oversampling,” *arXiv preprint arXiv:1903.09730*, 2019.
- [65] K. Cao, C. Wei, A. Gaidon, N. Arechiga, and T. Ma, “Learning imbalanced datasets with label-distribution-aware margin loss,” *arXiv preprint arXiv:1906.07413*, 2019.
- [66] Y. Cui, M. Jia, T.-Y. Lin, Y. Song, and S. Belongie, “Class-balanced loss based on effective number of samples,” in *CVPR*, 2019.
- [67] Q. Dong, S. Gong, and X. Zhu, “Imbalanced deep learning by minority class incremental rectification,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018.
- [68] C. Huang, Y. Li, C. L. Chen, and X. Tang, “Deep imbalanced learning for face recognition and attribute prediction,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.
- [69] S. Khan, M. Hayat, S. W. Zamir, J. Shen, and L. Shao, “Striking the right balance with uncertainty,” in *CVPR*, 2019.
- [70] K. He, X. Zhang, S. Ren, and J. Sun, “Identity mappings in deep residual networks,” in *ECCV*, Springer, 2016.
- [71] T. Zheng and W. Deng, “Cross-pose lfw: A database for studying cross-pose face recognition in unconstrained environments,” *Beijing University of Posts and Telecommunications, Tech. Rep.*, vol. 5, 2018.
- [72] S. Sengupta, J.-C. Chen, C. Castillo, V. M. Patel, R. Chellappa, and D. W. Jacobs, “Frontal to profile face verification in the wild,” in *WACV*, IEEE, 2016.
- [73] S. Gong, V. N. Boddeti, and A. K. Jain, “On the capacity of face representation,” *arXiv preprint arXiv:1709.10433*, 2017.
- [74] S. Gong, X. Liu, and A. K. Jain, “Jointly de-biasing face recognition and demographic attribute estimation,” *ECCV*, 2020.
- [75] S. Gong, X. Liu, and A. K. Jain, “Mitigating face recognition bias via group adaptive classifier,” in *CVPR*, 2021.
- [76] J. B. Tenenbaum, V. De Silva, and J. C. Langford, “A global geometric framework for nonlinear dimensionality reduction,” *Science*, vol. 290, no. 5500, pp. 2319–2323, 2000.
- [77] K. Fukunaga and D. R. Olsen, “An algorithm for finding intrinsic dimensionality of data,” *IEEE Transactions on Computers*, vol. 100, no. 2, pp. 176–183, 1971.
- [78] J. Bruske and G. Sommer, “Intrinsic dimensionality estimation with optimally topology preserving maps,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 5, pp. 572–575, 1998.

- [79] P. J. Verveer and R. P. W. Duin, “An evaluation of intrinsic dimensionality estimators,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, no. 1, pp. 81–86, 1995.
- [80] A. S. Georghiades, P. N. Belhumeur, and D. J. Kriegman, “From few to many: Illumination cone models for face recognition under variable lighting and pose,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 6, pp. 643–660, 2001.
- [81] H. Murase and S. K. Nayar, “Visual learning and recognition of 3-d objects from appearance,” *International Journal of Computer Vision*, vol. 14, no. 1, pp. 5–24, 1995.
- [82] P. Grassberger and I. Procaccia, “Measuring the strangeness of strange attractors,” in *The Theory of Chaotic Attractors*, pp. 170–189, Springer, 2004.
- [83] F. Camastra and A. Vinciarelli, “Estimating the intrinsic dimension of data with a fractal-based method,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 10, pp. 1404–1407, 2002.
- [84] B. Kégl, “Intrinsic dimension estimation using packing numbers,” in *Advances in Neural Information Processing Systems*, 2003.
- [85] M. Hein and J.-Y. Audibert, “Intrinsic dimensionality estimation of submanifolds in  $\mathbb{R}^d$ ,” in *International Conference on Machine Learning*, 2005.
- [86] E. Levina and P. J. Bickel, “Maximum likelihood estimation of intrinsic dimension,” in *Advances in Neural Information Processing Systems*, 2005.
- [87] I. T. Jolliffe, “Principal component analysis and factor analysis,” in *Principal Component Analysis*, pp. 115–128, Springer, 1986.
- [88] J. B. Kruskal, “Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis,” *Psychometrika*, vol. 29, no. 1, pp. 1–27, 1964.
- [89] M. Belkin and P. Niyogi, “Laplacian eigenmaps for dimensionality reduction and data representation,” *Neural Computation*, vol. 15, no. 6, pp. 1373–1396, 2003.
- [90] S. T. Roweis and L. K. Saul, “Nonlinear dimensionality reduction by locally linear embedding,” *Science*, vol. 290, no. 5500, pp. 2323–2326, 2000.
- [91] R. R. Coifman and S. Lafon, “Diffusion maps,” *Applied and Computational Harmonic Analysis*, vol. 21, no. 1, pp. 5–30, 2006.
- [92] G. E. Hinton and R. R. Salakhutdinov, “Reducing the dimensionality of data with neural networks,” *Science*, vol. 313, no. 5786, pp. 504–507, 2006.
- [93] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, “Extracting and composing robust features with denoising autoencoders,” in *International Conference on Machine Learning*, pp. 1096–1103, ACM, 2008.

- [94] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, “Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion,” *Journal of Machine Learning Research*, vol. 11, no. Dec, pp. 3371–3408, 2010.
- [95] R. Hadsell, S. Chopra, and Y. LeCun, “Dimensionality reduction by learning an invariant mapping,” in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1735–1742, 2006.
- [96] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, “Curriculum learning,” in *International Conference on Machine Learning*, pp. 41–48, ACM, 2009.
- [97] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.
- [98] I. Loshchilov and F. Hutter, “SGDR: stochastic gradient descent with warm restarts,” *arXiv preprint arXiv:1608.03983*, 2016.
- [99] S. Liao, Z. Lei, D. Yi, and S. Z. Li, “A benchmark study of large-scale unconstrained face recognition,” in *IEEE International Joint Conference on Biometrics (IJCB)*, 2014.
- [100] N. A. Schmid and J. A. O’Sullivan, “Performance prediction methodology for biometric systems using a large deviations approach,” *IEEE Transactions on Signal Processing*, vol. 52, no. 10, pp. 3036–3045, 2004.
- [101] N. A. Schmid, M. V. Ketkar, H. Singh, and B. Cukic, “Performance analysis of iris-based identification system at the matching score level,” *IEEE Transactions on Information Forensics and Security*, vol. 1, no. 2, pp. 154–168, 2006.
- [102] J. Bhatnagar and A. Kumar, “On estimating performance indices for biometric identification,” *Pattern Recognition*, vol. 42, no. 9, pp. 1803–1815, 2009.
- [103] P. Wang, Q. Ji, and J. L. Wayman, “Modeling and predicting face recognition system performance based on analysis of similarity scores,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 4, pp. 665–670, 2007.
- [104] C. E. Rasmussen and C. K. Williams, *Gaussian Processes for Machine Learning*, vol. 1. MIT Press Cambridge, 2006.
- [105] Y. Gal and Z. Ghahramani, “Dropout as a bayesian approximation: Representing model uncertainty in deep learning,” *arXiv:1506.02142*, vol. 2, 2015.
- [106] Y. Gal and Z. Ghahramani, “A theoretically grounded application of dropout in recurrent neural networks,” in *NIPS*, 2016.
- [107] Y. Gal and Z. Ghahramani, “Bayesian convolutional neural networks with bernoulli approximate variational inference,” *arXiv:1506.02158*, 2015.
- [108] A. Kendall and Y. Gal, “What uncertainties do we need in bayesian deep learning for computer vision?,” in *NIPS*, 2017.

- [109] A. Der Kiureghian and O. Ditlevsen, “Aleatory or epistemic? does it matter?,” *Structural Safety*, vol. 31, no. 2, pp. 105–112, 2009.
- [110] S. Pankanti, S. Prabhakar, and A. K. Jain, “On the individuality of fingerprints,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 8, pp. 1010–1025, 2002.
- [111] Y. Zhu, S. C. Dass, and A. K. Jain, “Statistical models for assessing the individuality of fingerprints,” *IEEE Transactions on Information Forensics and Security*, vol. 2, no. 3, pp. 391–401, 2007.
- [112] J. Daugman, “Information theory and the iriscode,” *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 2, pp. 400–409, 2016.
- [113] A. Adler, R. Youmaran, and S. Loyka, “Towards a measure of biometric feature information,” *Pattern Analysis and Applications*, vol. 12, no. 3, pp. 261–270, 2009.
- [114] J. Ba and R. Caruana, “Do deep nets really need to be deep?,” in *NIPS*, 2014.
- [115] D. P. Kingma, T. Salimans, and M. Welling, “Variational dropout and the local reparameterization trick,” in *NIPS*, 2015.
- [116] C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra, “Weight uncertainty in neural networks,” *arXiv:1505.05424*, 2015.
- [117] L. Bottou, “Large-scale machine learning with stochastic gradient descent,” in *Proceedings of COMPSTAT’2010*, pp. 177–186, Springer, 2010.
- [118] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning representations by back-propagating errors,” *Cognitive Modeling*, vol. 5, no. 3, p. 1, 1988.
- [119] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*. John Wiley & Sons, 2012.
- [120] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. John Wiley & Sons, 2012.
- [121] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions,” in *CVPR*, 2015.
- [122] D. Wang, C. Otto, and A. K. Jain, “Face search at scale: 80 million gallery,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1122 – 1136, 2017.
- [123] X. Wu, R. He, Z. Sun, and T. Tan, “A light cnn for deep face representation with noisy labels,” *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 11, pp. 2884–2896, 2018.
- [124] R. Ranjan, A. Bansal, H. Xu, S. Sankaranarayanan, J.-C. Chen, C. D. Castillo, and R. Chellappa, “Crystal loss and quality pooling for unconstrained face verification and recognition,” *arXiv preprint arXiv:1804.01159*, 2018.
- [125] W. Xie, L. Shen, and A. Zisserman, “Comparator networks,” *arXiv preprint arXiv:1807.11440*, 2018.

- [126] I. Kemelmacher-Shlizerman, S. M. Seitz, D. Miller, and E. Brossard, “The megaface benchmark: 1 million faces for recognition at scale,” in *CVPR*, 2016.
- [127] L. Best-Rowden and A. K. Jain, “Longitudinal study of automatic face recognition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 1, pp. 148–162, 2018.
- [128] M. Alvi, A. Zisserman, and C. Nellåker, “Turning a blind eye: Explicit removal of biases and variation from deep neural network embeddings,” in *ECCV*, 2018.
- [129] L. A. Hendricks, K. Burns, K. Saenko, T. Darrell, and A. Rohrbach, “Women also snowboard: Overcoming bias in captioning models,” in *ECCV*, 2018.
- [130] T. Wang, J. Zhao, M. Yatskar, K.-W. Chang, and V. Ordonez, “Balanced datasets are not enough: Estimating and mitigating gender bias in deep image representations,” in *CVPR*, 2019.
- [131] D. Madras, E. Creager, T. Pitassi, and R. Zemel, “Learning adversarially fair and transferable representations,” in *ICML*, 2018.
- [132] M. Kearns, S. Neel, A. Roth, and Z. S. Wu, “An empirical study of rich subgroup fairness for machine learning,” in *Proceedings of the Conference on Fairness, Accountability, and Transparency*, 2019.
- [133] J. Zhao, T. Wang, M. Yatskar, V. Ordonez, and K.-W. Chang, “Men also like shopping: Reducing gender bias amplification using corpus-level constraints,” in *EMNLP*, 2017.
- [134] Z. Wang, K. Qinami, Y. Karakozis, K. Genova, P. Nair, K. Hata, and O. Russakovsky, “Towards fairness in visual recognition: Effective strategies for bias mitigation,” *arXiv preprint arXiv:1911.11834*, 2019.
- [135] A. Grover, J. Song, A. Kapoor, K. Tran, A. Agarwal, E. J. Horvitz, and S. Ermon, “Bias correction of learned generative models using likelihood-free importance weighting,” in *NeurIPS*, 2019.
- [136] F. Calmon, D. Wei, B. Vinzamuri, K. N. Ramamurthy, and K. R. Varshney, “Optimized pre-processing for discrimination prevention,” in *NeurIPS*, 2017.
- [137] M. P. Kim, A. Ghorbani, and J. Zou, “Multiaccuracy: Black-box post-processing for fairness in classification,” in *AAAI/ACM*, 2019.
- [138] G. Pleiss, M. Raghavan, F. Wu, J. Kleinberg, and K. Q. Weinberger, “On fairness and calibration,” in *NeurIPS*, 2017.
- [139] Y. Zhang and Z.-H. Zhou, “Cost-sensitive face recognition,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 32, no. 10, pp. 1758–1769, 2009.
- [140] Y.-H. Liu and Y.-T. Chen, “Face recognition using total margin-based adaptive fuzzy support vector machines,” *IEEE Transactions on Neural Networks*, vol. 18, no. 1, pp. 178–192, 2007.
- [141] X. Yin, X. Yu, K. Sohn, X. Liu, and M. Chandraker, “Feature transfer learning for face recognition with under-represented data,” in *CVPR*, 2019.

- [142] X. Zhang, Z. Fang, Y. Wen, Z. Li, and Y. Qiao, “Range loss for deep face recognition with long-tailed training data,” in *CVPR*, 2017.
- [143] A. Amini, A. Soleimany, W. Schwarting, S. Bhatia, and D. Rus, “Uncovering and mitigating algorithmic bias through learned latent structure,” in *AAAI/ACM Conference on AI, Ethics, and Society*, 2019.
- [144] P. Wang, F. Su, Z. Zhao, Y. Guo, Y. Zhao, and B. Zhuang, “Deep class-skewed learning for face recognition,” *Neurocomputing*, 2019.
- [145] H. Qin, “Asymmetric rejection loss for fairer face recognition,” *arXiv preprint arXiv:2002.03276*, 2020.
- [146] D. Moyer, S. Gao, R. Brekelmans, A. Galstyan, and G. Ver Steeg, “Invariant representations without adversarial training,” in *NeurIPS*, 2018.
- [147] J. Song, P. Kalluri, A. Grover, S. Zhao, and S. Ermon, “Learning controllable fair representations,” in *ICAIS*, 2019.
- [148] R. Zemel, Y. Wu, K. Swersky, T. Pitassi, and C. Dwork, “Learning fair representations,” in *ICML*, 2013.
- [149] M. Hardt, E. Price, and N. Srebro, “Equality of opportunity in supervised learning,” in *NeurIPS*, 2016.
- [150] F. Locatello, G. Abbati, T. Rainforth, S. Bauer, B. Schölkopf, and O. Bachem, “On the fairness of disentangled representations,” in *NeurIPS*, 2019.
- [151] X. Yin, X. Yu, K. Sohn, X. Liu, and M. Chandraker, “Towards large-pose face frontalization in the wild,” in *ICCV*, 2017.
- [152] L. Tran, X. Yin, and X. Liu, “Representation learning by rotating your faces,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 41, no. 12, pp. 3007–3021, 2019.
- [153] J. Schmidhuber, “Learning factorial codes by predictability minimization,” *Neural Computation*, vol. 4, no. 6, pp. 863–879, 1992.
- [154] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” in *NIPS*, 2014.
- [155] E. Tzeng, J. Hoffman, T. Darrell, and K. Saenko, “Simultaneous deep transfer across domains and tasks,” in *CVPR*, 2015.
- [156] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, “Adversarial discriminative domain adaptation,” in *CVPR*, 2017.
- [157] M. Long, Z. Cao, J. Wang, and M. I. Jordan, “Conditional adversarial domain adaptation,” in *NIPS*, 2018.

- [158] C. Tao, F. Lv, L. Duan, and M. Wu, “Minimax entropy network: Learning category-invariant features for domain adaptation,” *arXiv preprint arXiv:1904.09601*, 2019.
- [159] B. Yin, L. Tran, H. Li, X. Shen, and X. Liu, “Towards interpretable face recognition,” in *ICCV*, 2019.
- [160] Y. Liu, Z. Wang, H. Jin, and I. Wassell, “Multi-task adversarial network for disentangled feature learning,” in *CVPR*, 2018.
- [161] Y. Liu, F. Wei, J. Shao, L. Sheng, J. Yan, and X. Wang, “Exploring disentangled feature representation beyond face identification,” in *CVPR*, 2018.
- [162] L. Tran, X. Yin, and X. Liu, “Disentangled representation learning GAN for pose-invariant face recognition,” in *CVPR*, 2017.
- [163] F. Liu, D. Zeng, Q. Zhao, and X. Liu, “Disentangling features in 3D face shapes for joint face reconstruction and recognition,” in *CVPR*, 2018.
- [164] H. Kim and A. Mnih, “Disentangling by factorising,” *arXiv preprint arXiv:1802.05983*, 2018.
- [165] S. Narayanaswamy, T. B. Paige, J.-W. Van de Meent, A. Desmaison, N. Goodman, P. Kohli, F. Wood, and P. Torr, “Learning disentangled representations with semi-supervised deep generative models,” in *NIPS*, 2017.
- [166] F. Locatello, S. Bauer, M. Lucic, S. Gelly, B. Schölkopf, and O. Bachem, “Challenging common assumptions in the unsupervised learning of disentangled representations,” *arXiv preprint arXiv:1811.12359*, 2018.
- [167] Z. Zhang, L. Tran, X. Yin, Y. Atoum, J. Wan, N. Wang, and X. Liu, “Gait recognition via disentangled representation learning,” in *CVPR*, 2019.
- [168] H. Han, K. J. Anil, S. Shan, and X. Chen, “Heterogeneous face attribute estimation: A deep multi-task learning approach,” *IEEE Trans. Pattern Analysis Machine Intelligence*, vol. PP, no. 99, pp. 1–1, 2017.
- [169] C. M. Cook, J. J. Howard, Y. B. Sirotin, J. L. Tipton, and A. R. Vemury, “Demographic effects in facial recognition and their dependence on image acquisition: An evaluation of eleven commercial systems,” *IEEE Transactions on Biometrics, Behavior, and Identity Science*, 2019.
- [170] F. Wang, J. Cheng, W. Liu, and H. Liu, “Additive margin softmax for face verification,” *IEEE Signal Processing Letters*, vol. 25, no. 7, pp. 926–930, 2018.
- [171] A. Jourabloo, X. Yin, and X. Liu, “Attribute preserved face de-identification,” in *ICB*, 2015.
- [172] B.-C. Chen, C.-S. Chen, and W. H. Hsu, “Cross-age reference coding for age-invariant face recognition and retrieval,” in *ECCV*, 2014.
- [173] R. Rothe, R. Timofte, and L. Van Gool, “Deep expectation of real and apparent age from a single image without facial landmarks,” *IJCV*, 2018.

- [174] Z. Zhang, Y. Song, and H. Qi, “Age progression/regression by conditional adversarial autoencoder,” in *CVPR*, 2017.
- [175] S. Moschoglou, A. Papaioannou, C. Sagonas, J. Deng, I. Kotsia, and S. Zafeiriou, “Agedb: the first manually collected, in-the-wild age database,” in *CVPRW*, 2017.
- [176] Z. Niu, M. Zhou, L. Wang, X. Gao, and G. Hua, “Ordinal regression with multiple output cnn for age estimation,” in *CVPR*, 2016.
- [177] J. Cheng, Y. Li, J. Wang, L. Yu, and S. Wang, “Exploiting effective facial patches for robust gender recognition,” *Tsinghua Science and Technology*, vol. 24, no. 3, pp. 333–345, 2019.
- [178] S. Setty, M. Husain, P. Beham, J. Gudavalli, M. Kandasamy, R. Vaddi, V. Hemadri, J. C. Karure, R. Raju, Rajan, V. Kumar, and C. V. Jawahar, “Indian Movie Face Database: A Benchmark for Face Recognition Under Wide Variations,” in *NCVPRIPG*, 2013.
- [179] “<http://trillionpairs.deepglint.com/overview>,”
- [180] D. Deb, L. Best-Rowden, and A. K. Jain, “Face recognition performance under aging,” in *CVPRW*, 2017.
- [181] [https://yanweifu.github.io/FG\\_NET\\_data](https://yanweifu.github.io/FG_NET_data).
- [182] X. Yin and X. Liu, “Multi-task convolutional neural network for pose-invariant face recognition,” *IEEE Trans. Image Processing*, vol. 27, no. 2, pp. 964–975, 2017.
- [183] W. Xie and A. Zisserman, “Multicolumn networks for face recognition,” *arXiv preprint arXiv:1807.09192*, 2018.
- [184] Y. Shi and A. K. Jain, “Probabilistic face embeddings,” in *ICCV*, 2019.
- [185] D. Kang, D. Dhar, and A. Chan, “Incorporating side information by adaptive convolution,” in *NeurIPS*, 2017.
- [186] J. Yang, Z. Ren, C. Gan, H. Zhu, and D. Parikh, “Cross-channel communication networks,” in *NeurIPS*, 2019.
- [187] Q. Wang, B. Wu, P. Zhu, P. Li, W. Zuo, and Q. Hu, “Eca-net: Efficient channel attention for deep convolutional neural networks,” *arXiv preprint arXiv:1910.03151*, 2019.
- [188] X. Lu, W. Wang, C. Ma, J. Shen, L. Shao, and F. Porikli, “See more, know more: Unsupervised video object segmentation with co-attention siamese networks,” in *CVPR*, 2019.
- [189] R. Hou, H. Chang, M. Bingpeng, S. Shan, and X. Chen, “Cross attention network for few-shot classification,” in *NeurIPS*, 2019.
- [190] K. Su, D. Yu, Z. Xu, X. Geng, and C. Wang, “Multi-person pose estimation with enhanced channel-wise and spatial information,” in *CVPR*, 2019.
- [191] T.-K. Hu, Y.-Y. Lin, and P.-C. Hsiu, “Learning adaptive hidden layers for mobile gesture recognition,” in *AAAI*, 2018.



- [192] Y. Zhang, D. Zhao, J. Sun, G. Zou, and W. Li, “Adaptive convolutional neural network and its application in face recognition,” *Neural Processing Letters*, 2016.
- [193] S. Li, J. Xing, Z. Niu, S. Shan, and S. Yan, “Shape driven kernel adaptation in convolutional neural network for robust facial traits recognition,” in *CVPR*, 2015.
- [194] J. Du, S. Zhang, G. Wu, J. M. Moura, and S. Kar, “Topology adaptive graph convolutional networks,” *arXiv preprint arXiv:1710.10370*, 2017.
- [195] C. Ding, Y. Li, Y. Xia, L. Zhang, and Y. Zhang, “Automatic kernel size determination for deep neural networks based hyperspectral image classification,” *Remote Sensing*, 2018.
- [196] X. Li, W. Wang, X. Hu, and J. Yang, “Selective kernel networks,” in *CVPR*, 2019.
- [197] C. Ding, Y. Li, Y. Xia, W. Wei, L. Zhang, and Y. Zhang, “Convolutional neural networks based hyperspectral image classification method with adaptive kernels,” *Remote Sensing*, 2017.
- [198] X. Li, M. Ye, Y. Liu, and C. Zhu, “Adaptive deep convolutional neural networks for scene-specific object detection,” *IEEE Transactions on Circuits and Systems for Video Technology*, 2017.
- [199] H. Su, V. Jampani, D. Sun, O. Gallo, E. Learned-Miller, and J. Kautz, “Pixel-adaptive convolutional neural networks,” in *CVPR*, 2019.
- [200] J. Zamora Esquivel, A. Cruz Vargas, P. Lopez Meyer, and O. Tickoo, “Adaptive convolutional kernels,” in *ICCV Workshops*, 2019.
- [201] B. Klein, L. Wolf, and Y. Afek, “A dynamic convolutional layer for short range weather prediction,” in *CVPR*, 2015.
- [202] X. Jia, B. De Brabandere, T. Tuytelaars, and L. V. Gool, “Dynamic filter networks,” in *NeurIPS*, 2016.
- [203] X. Zhang, T. Wang, J. Qi, H. Lu, and G. Wang, “Progressive attention guided recurrent network for salient object detection,” in *CVPR*, 2018.
- [204] Z. Chen, L. Liu, I. Sa, Z. Ge, and M. Chli, “Learning context flexible attention model for long-term visual place recognition,” *IEEE Robotics and Automation Letters*, 2018.
- [205] L. Chen, H. Zhang, J. Xiao, L. Nie, J. Shao, W. Liu, and T.-S. Chua, “SCA-CNN: Spatial and channel-wise attention in convolutional networks for image captioning,” in *CVPR*, 2017.
- [206] B. Chen, P. Li, C. Sun, D. Wang, G. Yang, and H. Lu, “Multi attention module for visual tracking,” *Pattern Recognition*, 2019.
- [207] A. A. Bastidas and H. Tang, “Channel attention networks,” in *CVPR Workshops*, 2019.
- [208] H. Ling, J. Wu, J. Huang, J. Chen, and P. Li, “Attention-based convolutional neural network for deep face recognition,” *Multimedia Tools and Applications*, 2020.

- [209] V. A. Sindagi and V. M. Patel, “Ha-ccn: Hierarchical attention-based crowd counting network,” *IEEE Transactions on Image Processing*, 2019.
- [210] J. Hu, L. Shen, and G. Sun, “Squeeze-and-excitation networks,” in *CVPR*, 2018.
- [211] S. Woo, J. Park, J.-Y. Lee, and I. So Kweon, “Cbam: Convolutional block attention module,” in *ECCV*, 2018.
- [212] M. Sadiq, D. Shi, M. Guo, and X. Cheng, “Facial landmark detection via attention-adaptive deep network,” *IEEE Access*, 2019.
- [213] D. Linsley, D. Schiebler, S. Eberhardt, and T. Serre, “Learning what and where to attend,” in *ICLR*, 2019.
- [214] U. Ozbulak, “Pytorch cnn visualizations.” <https://github.com/utkuozbulak/pytorch-cnn-visualizations>, 2019.
- [215] T. N. Kipf and M. Welling, “Semi-supervised classification with graph convolutional networks,” in *ICLR*, 2017.
- [216] M. Defferrard, X. Bresson, and P. Vandergheynst, “Convolutional neural networks on graphs with fast localized spectral filtering,” in *NeurIPS*, 2016.
- [217] M. Zhang, Z. Cui, M. Neumann, and Y. Chen, “An end-to-end deep learning architecture for graph classification,” in *AAAI*, 2018.
- [218] M. Niepert, M. Ahmed, and K. Kutzkov, “Learning convolutional neural networks for graphs,” in *ICML*, PMLR, 2016.
- [219] J.-C. Vialatte, V. Gripon, and G. Mercier, “Generalizing the convolution operator to extend cnns to irregular domains,” *arXiv preprint arXiv:1606.01166*, 2016.
- [220] A. J.-P. Tixier, G. Nikolentzos, P. Meladianos, and M. Vazirgiannis, “Graph classification with 2d convolutional neural networks,” in *ICANN*, Springer, 2019.
- [221] Y. Huang, Y. Wang, Y. Tai, X. Liu, P. Shen, S. Li, J. Li, and F. Huang, “Curricularface: adaptive curriculum learning loss for deep face recognition,” in *CVPR*, 2020.
- [222] Y. Sun, X. Wang, and X. Tang, “Deeply learned face representations are sparse, selective, and robust,” in *CVPR*, 2015.
- [223] S. Sankaranarayanan, A. Alavi, C. D. Castillo, and R. Chellappa, “Triplet probabilistic embedding for face verification and clustering,” in *BTAS*, IEEE, 2016.
- [224] L. Tran, X. Yin, and X. Liu, “Disentangled representation learning gan for pose-invariant face recognition,” in *CVPR*, 2017.
- [225] J. Deng, S. Cheng, N. Xue, Y. Zhou, and S. Zafeiriou, “Uv-gan: Adversarial facial uv map completion for pose-invariant face recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7093–7102, 2018.

- [226] Y. Huang, P. Shen, Y. Tai, S. Li, X. Liu, J. Li, F. Huang, and R. Ji, “Improving face recognition from hard samples via distribution distillation loss,” in *ECCV*, pp. 138–154, Springer, 2020.
- [227] J. Yang, P. Ren, D. Zhang, D. Chen, F. Wen, H. Li, and G. Hua, “Neural aggregation network for video face recognition,” in *CVPR*, 2017.
- [228] Y. Liu, J. Yan, and W. Ouyang, “Quality aware network for set to set recognition,” in *CVPR*, 2017.
- [229] X. Zhang, R. Zhao, Y. Qiao, X. Wang, and H. Li, “Adacos: Adaptively scaling cosine logits for effectively learning deep face representations,” in *CVPR*, 2019.
- [230] X. Zhang, R. Zhao, J. Yan, M. Gao, Y. Qiao, X. Wang, and H. Li, “P2sgard: Refined gradients for optimizing deep face models,” in *CVPR*, 2019.