

# PoseFace: Pose-Invariant Features and Pose-Adaptive Loss for Face Recognition

Qiang Meng\* Xiaqing Xu\* Xiaobo Wang<sup>†</sup> Yang Qian<sup>‡</sup> Yunxiao Qin<sup>§</sup>  
 Zezheng Wang\* Chenxu Zhao<sup>¶</sup> Feng Zhou\* Zhen Lei \*\*††

## Abstract

Despite the great success achieved by deep learning methods in face recognition, severe performance drops are observed for large pose variations in unconstrained environments (e.g., in cases of surveillance and photo-tagging). To address it, current methods either deploy pose-specific models or frontalize faces by additional modules. Still, they ignore the fact that identity information should be consistent across poses and are not realizing the data imbalance between frontal and profile face images during training. In this paper, we propose an efficient PoseFace framework which utilizes the facial landmarks to disentangle the pose-invariant features and exploits a pose-adaptive loss to handle the imbalance issue adaptively. Extensive experimental results on the benchmarks of Multi-PIE, CFP, CPLFW and IJB have demonstrated the superiority of our method over the state-of-the-arts.

## 1. Introduction

Recent years have witnessed the breakthrough of deep convolutional face recognition techniques [9, 29, 34, 8, 40, 47, 24]. However, face recognition with large pose variations is still a great challenge in computer vision. The performance of most face recognition models degrades by over 10% from frontal-frontal to frontal-profile recognition [30]. That is mainly because: (a) proportions of frontal and profile faces in most modern datasets are highly imbalanced [5] and that prevents data-driven deep models to capture robust

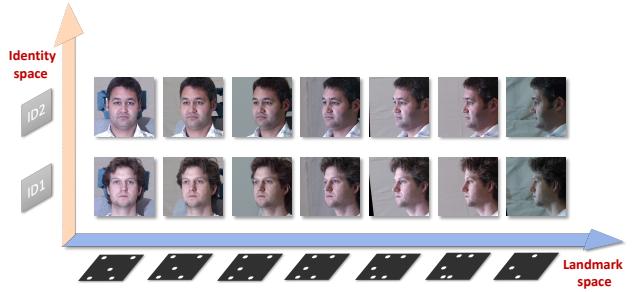


Figure 1: Face images are mainly composed of identity information and pose information. As pose information is encoded in the facial landmarks, we decompose the faces into identities and landmarks, which distribute in two orthogonal subspaces separately.

identity information; (b) profile faces are intrinsically hard for recognition. A face image mainly consists of identity information and pose information. How to efficiently disentangle them is still an open issue.

To recognize faces in large poses, existing methods can be typically categorized into two classes. The first one is to normalize profile faces into a frontal view [27, 52, 38, 15, 17, 50]. Besides the artifacts, synthesizing faces also suffers from extra computations as well as loss of identity information, limiting their usage for recognition. The second category aims at learning pose-invariant facial features. Early works [22, 49, 46] train multiple pose-specific models for faces under various poses. To simplify the process, DFN [13] and DREAM [5] unify multiple paths into a pose-aware transformation over features. However, their disentanglements are incomplete and not straightforward, as well as cannot be applied to in-the-wild recognition datasets where simultaneous existences of frontal and profile faces for each class cannot be guaranteed.

To address the aforementioned issues, we propose a novel and efficient framework called PoseFace which contributes to large-pose face recognition in two aspects. The first one is to achieve pose-invariant embeddings in an end-to-end and self-disentanglement manner, instead of naively

\*Algorithm Research, AiBee Inc.

<sup>†</sup>Institute of Automation, Chinese Academy of Sciences

<sup>‡</sup>The University of Sydney

<sup>§</sup>Northwestern Polytechnical University

<sup>¶</sup>Corresponding author

<sup>||</sup>Academy of Sciences, Mininglamp Technology

<sup>\*\*</sup>CBSR & NLPR, Institute of Automation, Chinese Academy of Sciences, Beijing, China

<sup>††</sup>School of Artificial Intelligence, University of Chinese Academy of Sciences

comparing features from frontal and profile faces. Specifically, we map each input face into pose and identity features simultaneously, and enforce orthogonality between subspaces of two features as shown in Figure 1. To achieve complete disentanglement, we (a) constrain mappings to two subspaces to be linear and orthogonal; (b) ensure the purity of pose features (*i.e.*, only relates to pose) and therefore avoid identity information to be disentangled into pose features. To this end, we design an AutoEncoder which is pre-trained and builds one-to-one mappings between facial landmarks and feature vectors. The feature vectors are thus only related to landmarks and treated as pseudo labels for pose features. With the supervision of such labels, identity information are kept for recognition to the most extent and that prevents disentanglement from decreasing recognition performances (indicated by our experiments in table 2). In a word, the proposed self-disentanglement method is straightforward, and filters pose information efficiently while avoids wastage of identity information.

The second part of PoseFace is the pose-adaptive loss which handles data imbalances between profile and frontal faces. Compared to frontal faces, profile faces are harder to be captured and processed (*e.g.*, face detection and face alignment). Therefore, profile faces (pitch/yaw/roll angles  $> 60^\circ$ ) are of small proportions in large and in-the-wild datasets (*e.g.*, 0.19% for MS1MV2 [8] according to our analysis). Inspired by the weight-adaptive methods such as focal loss [18], Adacos [51] and Adaptiveface [20], we down-weight losses for frontal faces by reformulating the prevailing ArcFace loss [8]. In that way, we focus training on hard examples (*i.e.*, large pose faces) and prevent the vast number of easy samples (*i.e.*, near-frontal faces) from overwhelming the network. To sum up, the main contributions of this paper are as follows:

1. We propose a self-disentanglement method which distributes identity and pose features into two orthogonal subspaces, and therefore obtain pose-invariant features. Our self-disentanglement doesn't require pairwise inputs and works in an end-to-end manner. Moreover, disentangled pose features are irrelevant to identities that avoids affecting recognitions.
2. A pose-adaptive loss is designed to deal with the data imbalance problem. The loss increases penalties assigned to profile faces, and therefore prevents frontal faces from overwhelming the network.
3. Extensive experiments are conducted on benchmarks for face recognition under large pose variations. The results have verified the superiority of our approach over current state-of-the-art methods. For example, PoseFace achieves human-level performances when recognizing the profile faces on the CFP benchmark. On the MultiPIE benchmark, our PoseFace achieves

a 0.85%, 4.87%, 6.51% boosts on the rank-1 recognition rates when identifying faces with yaw angle  $60^\circ, 75^\circ, 90^\circ$  respectively.

## 2. Related Works

In this section, we first briefly review face recognition methods under large pose variations, which are mainly categorized into pose-invariant feature representation (Sec. 2.1) and face frontalization (Sec. 2.3). Then, we list disentanglement methods used in face-related literatures and analyze differences between them and our self-disentanglement method in Sec. 2.3.

### 2.1. Pose-Invariant Feature Representation

Pose-invariant feature representation methods aim at exploring features that are robust to changes of poses. One possible solution is the divide and conquer, *i.e.*, using multiple pose-specific models to handle different-pose faces. For example, c-CNN [46] introduces dynamically activated kernels to form various network structures for different inputs. Kernels in each layer are sparsely activated conditioned on present feature representations. PAMs [22] fuses scores from multiple pose-specific models to tackle pose variation. p-CNN [49] proposes a multi-task convolutional neural network where identity classification is the main task and pose, illumination, and expression estimations are the side tasks. The efficiency issue of such multi-model frameworks limits their usage in many real-world applications.

Besides the divide and conquer, some works present a single model to learn pose-invariant features. For example, DFN [13] aligns features across different poses by exploiting deformable convolutional modules. Their method takes paired samples from the same identity and penalizes the differences to reduce the intra-class feature variation. DREAM [5] adaptively adds residuals to the input deep representation to transform a profile face representation to a canonical frontal pose. Conventional multi-view subspace approaches [31, 32, 55] learn complex non-linear transformations that respectively project images captured under different poses to the common space, where the intra-class variation is minimized. These methods either cannot achieve complete disentanglement or fail to handle data imbalance issue. In contrast, our proposed self-disentanglement method can be directly applied to datasets even with small proportions of profile faces. The profile faces are further increase-weighted by a pose-adaptive loss.

### 2.2. Face Frontalization

Face frontalization methods normalize faces to a canonical frontal view and then use synthesized faces for recognition. Recently, generative adversarial networks (GAN) has shown its great potential in synthesizing faces [27, 52, 38, 15, 17, 50, 54, 3]. TP-GAN [17] and PIM [52] normalize

faces by simultaneously perceiving global structures and local details. FNM [27] is an unsupervised face normalization method which includes a face expert network and face attention discriminators. In CAPG-GAN [15], head pose information not only guides the generator in the process, but also is used as a controllable condition during inference. Despite the improvements achieved by GAN, it suffers from unreal textures and loss of identity information in synthesized faces, as well as its high computational cost.

### 2.3. Disentanglement in Face Recognition

A great amount of works for robust face recognition share the same high-level idea of disentanglement. By disentangling factors such as pose, age and expressions, they finally achieve face representations which are robust to factor changes. Most works adopt generative adversarial networks or AutoEncoders, to either frontalize faces [17, 4, 19] or re-construct faces under different factors [2, 38, 37, 39]. Besides non-trivial training process and extra computations/modules, these methods usually require extra information such as facial texture maps [4], 3D Morphable Model [2, 19, 39] and faces from multiviews [38, 38]. Most of them are based on theory and rarely applied to actual scenarios, according to the recent survey [25].

Wang *et al.* [41] presents another way of disentanglement by regressing ages from feature magnitudes. Even though naturally orthogonal to vector direction, magnitude is a scalar and therefore unable to describe complex factors. Other disentanglement methods penalize differences between faces from same identity but with different factors [13, 26] or enforce consistent distributions across modalities [48, 14]. These methods require pair-wise inputs from same identities and that restrict their usage in many scenarios, as profile and frontal faces are highly imbalance in modern recognition datasets.

Compared to previous works, our PoseFace is an efficient and straightforward framework. The proposed self-disentanglement is achieved by mapping faces into identity and landmark spaces, which are irrelevant to each other by orthogonal constraints. The whole process requires no pair-wise inputs or extra labels, and therefore can be trained in common datasets. In addition, the training speed and GPU memory consumptions are all close to the state-of-the-art ArcFace [8] as indicated in table 3. During inference, PoseFace has no difference with ArcFace as only the identity module (in figure 2) is needed.

## 3. Methodology

Our work is mainly based on the following two observations:

**Observation 1.** *The appearance of a face is composed of identity information and variation information (e.g., light-*

*ing, poses, and expressions) [43, 35]. Identity information and variation information should be irrelevant.*

**Observation 2.** *Head pose information can be encoded by facial landmarks [15]. Other information such as lighting and textures from the original face images are not that necessary for pose estimation.*

Note that in observation 2, 3D facial landmarks are normally needed to fully express out-of-plane 3D head poses, 2D facial landmarks are mostly distributed in a same plane and therefore insufficient to represent 3D head poses. However, the key idea of PoseFace is to remove pose-related information out of 2D images by the self-disentanglement. Therefore, we can employ 2D facial landmarks to help filter pose information in images.

The overall pipeline of the proposed method is shown in figure 2. Our proposed PoseFace mainly consists of a landmark module, an identity module and a pose module. The landmark module is pre-trained and served as the pose feature extractor. During the training phase, the orthogonal constraint and pose loss are introduced to train the identity module and the pose module and achieve self-disentanglement. The adaptive recognition loss is adjusted from the prevailing ArcFace [8] to increase-weight profile faces. During testing, only the identity module is used for evaluation, same as ArcFace. Below we will introduce our PoseFace from the perspectives of network details, pose-adaptive ArcFace (PAA) loss and pose-invariant features.

### 3.1. Network Details

As shown in figure 2, faces are first fed into the backbone network and then the identity/pose module disentangles backbone features to identity/landmark subspace. The disentanglement is achieved by an orthogonal constraint over the linear layer in each module. The landmark module takes landmarks as inputs and generates pseudo labels of pose features to supervise the learning of pose module.

**Pose Module.** The pose module is a linear layer with weight  $\mathbf{W}_P$  which projects the backbone feature  $F_b$  to the pose feature  $F_p$ .

**Identity Module.** In the identity module,  $F_b$  is first projected to identity feature  $F_i$  by  $W_I$  and then transferred to  $F_o$  for recognition via an extra feature layer. The feature layer, which takes the form of a fully connected layer in our implementation, is designed for the following two reasons: (1) The projected feature  $F_i$  may not be suitable for recognition and the extra layer can refine it for the task. (2) The projection is defined on Euclidean space where the product of two vectors equals 0 (*i.e.*,  $v_1^T v_2 = 0$ ) indicates orthogonal property in the space. In contrast, features for recognition are measured by cosine similarity. Therefore,  $v_1^T v_2 = 0$  means the two vectors are not similar. The domain difference is relieved by the introduced feature layer.

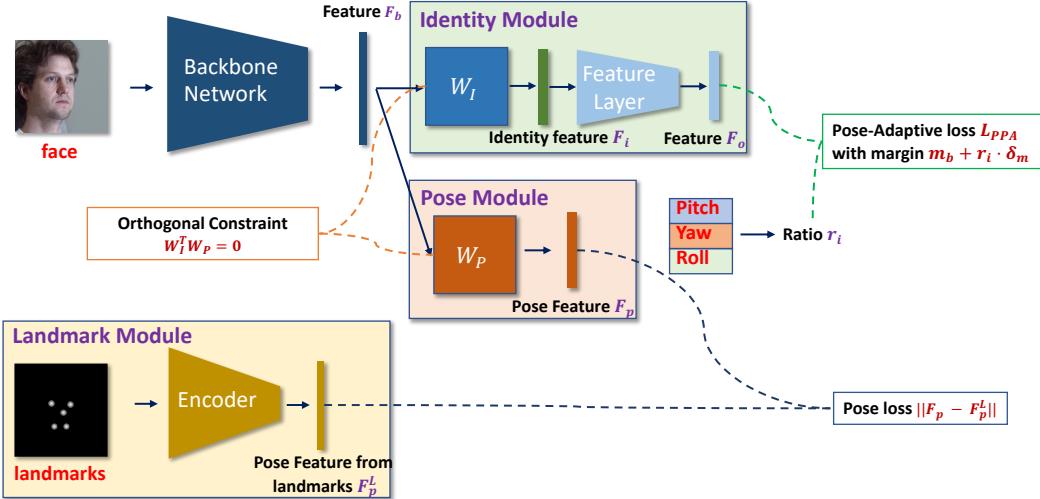


Figure 2: An overview of our proposed PoseFace framework. The pseudo label of pose feature  $F_p^L$  is generated by a pre-trained Encoder network from facial landmarks. The identity feature  $F_i$  and pose feature  $F_p$  are estimated by two fully connected layers with orthogonal weights  $\mathbf{W}_I$  and  $\mathbf{W}_P$  after the backbone network. The network is trained by (1) the pose-adaptive ArcFace loss with variable margins controlled by ratio  $r_i$ , which is estimated from facial angles, (2) the pose loss  $\|F_p - F_p^L\|$  and (3) the orthogonal constraint  $\mathbf{W}_I^T \mathbf{W}_P = 0$ .

**Landmark Module.** In PoseFace, pose feature  $F_p^L$  is directly estimated from facial landmarks and therefore is pose-related only. With the supervision of such  $F_p^L$  and an orthogonal constraint, pose information flows into the pose module while other information is well-kept in the identity module. That accomplishes the disentanglement without hurting recognition performance (ablation study in section 4.2 reveals the efficacy of such design).

To find the one-to-one mapping between pose feature and facial landmarks, we implement an Auto-Encoder network as shown in figure 3. The pre-trained Encoder is exploited to extract features in the landmark module. According to observation 2, we treat extracted features as pose features and use them as pseudo labels to train the pose module. Considering the sparsity of heatmaps, we adopt a weighted  $l_2$  loss to train the Auto-Encoder as shown in Eq. (1).

$$L_{ae} = \lambda_h \cdot \|\mathbf{H}_i \circ (\mathbf{H}_i - \mathbf{H}_o)\|_F + \|(\mathbf{1} - \mathbf{H}_i) \circ (\mathbf{H}_i - \mathbf{H}_o)\|_F \quad (1)$$

where  $\|\cdot\|_F$  means Frobenius norm and  $\circ$  means Hadamard product.  $\mathbf{H}_i, \mathbf{H}_o$  are input and output heatmaps, respectively.  $\lambda_h$  is the weight of the loss assigned to the area where  $\mathbf{H}_i = 1$ , i.e., the locations of the landmarks.

**Network Input.** Many works on pose-invariant feature learning take pairs of frontal and profile faces from the same identity as inputs. Differences between frontal and profile features are penalized to get rid of the pose information. In contrast, our approach takes one aligned face image, its corresponding landmarks and angles as one unit of input (marked in red in figure 2). The angles are estimated from facial landmarks. Pose-invariant features can be learned

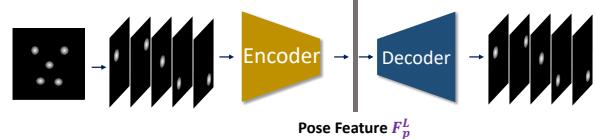


Figure 3: The AutoEncoder for extracting pose features. Heatmaps are synthesized from landmarks and we stack all heatmaps as inputs and outputs of the Auto-Encoder.

in an end-to-end manner by our self-disentanglement approach.

### 3.2. Pose-Adaptive ArcFace (PAA) Loss

Many existing models encounter large performance degradation when recognizing profile faces compared to frontal ones. Besides intrinsically hard for recognition, another reason is that numbers of frontal and profile samples are highly imbalanced in many training datasets [5]. In another word, the profile faces are hard and rare samples.

Inspired by the weight-adaptive methods such as focal loss [18], Adacos [51] and Adaptiveface [20], we modify the state-of-the-art ArcFace loss [8] by adapting the margin based on the facial angles as shown in figure 4. The margin for sample  $i$  is defined as  $m_i = m_b + r_i \cdot \delta_m$ . Here  $m_b$  is the base margin and  $\delta_m$  is the additional margin. The amount of added margin is controlled by ratio  $r_i \in [0, 1]$ , which is calculated according to the pitch/yaw/roll angles. Profile faces will be assigned with large ratios. In the end, we formulate the classification loss for identity features as pose-adaptive ArcFace (PAA) loss:

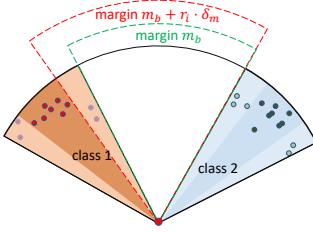


Figure 4: Geometrical interpretation of the adaptive margin. The dots with deeper colors represent profile faces. The harder the samples are, the larger margins will be used.

$$L_{PAA}(r_i) = -\frac{1}{n} \sum_{i=1}^N L_i, \quad \text{where}$$

$$L_i = \log \frac{e^{s \cdot (\cos(\theta_{y_i} + m_i))}}{e^{s \cdot (\cos(\theta_{y_i} + m_i))} + \sum_{j=1, j \neq y_i}^n e^{s \cdot \cos \theta_j}}. \quad (2)$$

where  $m_i = m_b + r_i \cdot \delta_m$  is the adaptive margin for different poses and other variables such as  $\theta_{y_j}, \theta_j, s$  are all consistent with those in ArcFace [8].

### 3.3. Pose-Invariant Features

In our implementation, identity features  $F_i$  and pose features  $F_p$  are enforced to be distributed in two orthogonal subspaces. By disentangling them, we ensure the generated identity features to be pose-invariant.

**Pose-Feature Constraint.** We treat features  $F_p^L$  from landmarks as the pseudo labels. The estimated pose features  $F_p = \mathbf{W}_p \cdot F_b$  from the pose module should be consistent with  $F_p^L$ , i.e.,  $\|F_p^L - F_p\|_2 = 0$ .

**Orthogonal Constraint.** Denote the features from backbone network as  $F_b$  and use matrix  $\mathbf{W}_I = [W_I^1, W_I^2, \dots, W_I^n], \mathbf{W}_P = [W_P^1, W_P^2, \dots, W_P^n]$  to represent the transformation of two orthogonal subspaces. Each base of  $\mathbf{W}_I$  and  $\mathbf{W}_P$  are normalized as the scales are not necessary for orthogonal requirement. We formulate the orthogonal constraint as

$$F_i = \mathbf{W}_I^T F_b, \quad F_p = \mathbf{W}_P^T F_b,$$

$$\widetilde{\mathbf{W}}_I = [\frac{W_I^1}{\|W_I^1\|_2}, \frac{W_I^2}{\|W_I^2\|_2}, \dots, \frac{W_I^n}{\|W_I^n\|_2}], \quad (3)$$

$$\widetilde{\mathbf{W}}_P = [\frac{W_P^1}{\|W_P^1\|_2}, \frac{W_P^2}{\|W_P^2\|_2}, \dots, \frac{W_P^n}{\|W_P^n\|_2}],$$

$$\|\widetilde{\mathbf{W}}_I^T \widetilde{\mathbf{W}}_P\|_F = 0.$$

Here  $\|\cdot\|_F$  denotes the Frobenius norm.

**PoseFace Loss.** Our problem can be formulated as

$$\begin{aligned} \min & \quad L_{PAA}(r_i), \\ \text{s.t.} & \quad \|F_p^L - F_p\|_2 = 0, \\ & \quad \|\widetilde{\mathbf{W}}_I^T \widetilde{\mathbf{W}}_P\|_F = 0. \end{aligned} \quad (4)$$

Introducing Lagrange multipliers  $\lambda_1, \lambda_2$ , we define the full version of PoseFace loss as

$$L_{PAA}(r_i) + \lambda_1 \cdot \|F_p^L - F_p\|_2 + \lambda_2 \cdot \|\widetilde{\mathbf{W}}_I^T \widetilde{\mathbf{W}}_P\|_F. \quad (5)$$

We call  $\lambda_1 \cdot \|F_p^L - F_p\|_2 + \lambda_2 \cdot \|\widetilde{\mathbf{W}}_I^T \widetilde{\mathbf{W}}_P\|_F$  the Orth loss, which is used to train pose-invariant features. Our PoseFace loss is a composition of PAA loss and Orth loss.

## 4. Experiments

In this section, we evaluate our PoseFace approach on various face recognition benchmarks with large pose variation. Specifically, we describe the implementation details in section 4.1, ablation study in section 4.2 and detailed results on benchmarks in section 4.3. In supplementary, we further present sensitive analysis on the hyperparameters  $\lambda_1, \lambda_2$  in section A and extra visualizations in sections C, D.

### 4.1. Implementation Details

We use Face Alignment Network (FAN) [1] to generate 68 landmarks and align face images into size  $108 \times 108$  by affine transformations. Pitch/yaw/roll angles are estimated based on the facial landmarks. In our implementation, we use a simple function  $\frac{|\text{yaw}|}{90^\circ}$  to estimate the adaptive ratio  $r_i$ . Roll and pitch angles are not considered as the roll's effect will be eliminated by face alignment while face images with large pitch angles are rare [5].

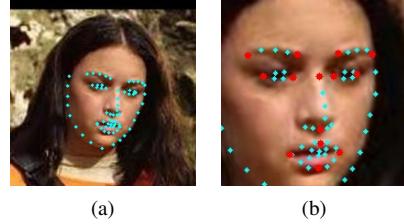


Figure 5: The procedure of data pre-processing. (a) A face image with the detected 68 facial landmarks. (b) The corresponding aligned  $108 \times 108$  face image with the 14 selected landmarks (marked in red) for pose feature extraction.

For the AutoEncoder network, we select 14 of 68 landmarks as the input because it is more efficient and some landmarks contain identity information (e.g., the shape of eyes and mouth). Using all 68 landmarks may degrade the integrity of the identity features. Figure 5(b) gives an example of our selected 14 landmarks. The AutoEncoder Network (see table A1 in supplementary for the detailed structure) is pre-trained by the facial landmarks detected from MS1MV2 [11, 8] (5.8M images, 85k unique identities).

MS1MV2 is also used as the training set of the PoseFace. Specifically, our PoseFace are trained by setting

$\lambda_1 = 200$ ,  $\lambda_2 = 100000$ ,  $m_b = 0.5$  with various  $\delta_m$ . Our baselines are trained with  $\lambda_1 = 0$ ,  $\lambda_2 = 0$ ,  $m_b = 0.5$  and  $\delta_m = 0$ , *i.e.*, ArcFace with margin 0.5 with same the network structure. In our experiments, we use ResNet [12] with 18, 34 and 50 layers as our backbone networks and rename our models as PoseFace18, PoseFace34 and PoseFace50, respectively. Next we describe our evaluation settings in the involved test databases.

**MultiPIE.** The CMU MultiPIE [10] dataset is the largest multi-view face recognition benchmark and contains 754,204 images of 337 identities from 15 viewpoints and 20 illumination conditions. We follow the evaluation protocol introduced in [46]. Images (15 poses, 20 illumination levels) of the first 150 identities are used for training. For testing, one frontal image with neutral illumination is chosen as the gallery image for each of the remaining 100 subjects. The remaining images are used as probes. Due to the limited number of training samples, PoseFace18 is pre-trained on the subset of MS1MV2 (0.68M images, 10K identities) and then fine-tuned on MultiPIE. The rank-1 accuracy is reported with regard to pose for comparison.

**CFP.** Celebrities in Frontal-Profile (CFP) dataset [30] is another challenging dataset with large and unconstrained pose variations. The dataset contains 500 celebrities, each individual has 10 frontal and 4 profile face images. We follow the standard protocol [30] for frontal-profile (FP) face verification. In particular, cross-validation is performed on 10 folders with 350 same and 350 not-same pairs. Means and standard deviations of Accuracy, Equal Error Rate (EER) and Area Under Curve (AUC) over the 10-fold experiments are also reported. PoseFace34 is warmed up by the subset of MS1MV2 for 20 epochs with a learning rate of 0.1. Then it is fine-tuned on the full MS1MV2 for 15 epochs with learning rate [0.1, 0.01, 0.001], 5 epochs for each stage.

**CPLFW.** CPLFW [53] dataset is a renovation of LFW dataset [16] with 5,749 identities and 11,652 images. It deliberately searches and selects 3,000 positive face pairs with large pose variations. Negative pairs with the same gender and race are also selected to reduce the influence of attribute difference between positive/negative pairs. PoseFace34 is trained the same way as for CFP. Mean and standard deviation of accuracy are reported under 10-fold cross-validation.

**IJB.** The IJB-B [42] dataset contains 1,845 subjects with 21.8K still images and 55K frames from 7,011 videos. The IJB-C [23] dataset is a further extension of IJB-B and contains about 3,500 identities with a total of 31,334 images and 117,542 unconstrained video frames. PoseFace50 is trained on MS1MV2 with batch size 512. The learning rate is initialized as 0.1 and divided by 10 at 5, 10, 15 epochs, and we finish the training at epoch 20. In the 1:1 verification, the numbers of positive/negative matches are 10k/8M in IJB-B and 19k/15M in IJB-C, and we report the TAR on FAR=1e-4, 1e-5 and 1e-6.

Method	$\pm 90^\circ$	$\pm 75^\circ$	$\pm 60^\circ$	$\pm 45^\circ$	$\pm 30^\circ$	$\pm 15^\circ$
baseline	81.68	88.03	92.75	96.99	98.77	99.37
PoseFace18(PAA)	85.37	91.70	95.40	98.52	99.36	99.77
PoseFace18(Orth)	87.87	94.86	97.75	<b>99.85</b>	<b>99.99</b>	99.99
PoseFace18(Orth+PAA)	<b>90.58</b>	<b>96.07</b>	<b>98.55</b>	99.62	99.97	<b>100</b>

Table 1: Rank-1 recognition rates (%) on MultiPIE.

## 4.2. Ablation Study

**Effect of our PAA and Orth.** To investigate the effect of our proposed PoseFace, we conduct several ablation experiments to show the superior performances on large-pose face recognition. The PoseFace models are trained by permutations of Orth loss and PAA loss. The baseline is trained by ArcFace with the same inference structure. Table 1 summarizes the rank-1 recognition rates. PoseFace18 achieves significant improvements across all poses, especially for profile poses (more than 60 degrees yaw variation defined as ‘profile’ [30]). The rank-1 recognition rates of PoseFace18 with PAA loss outperforms the baseline by 3.69%, 3.67% and 2.65% for  $\pm 90^\circ$ ,  $\pm 75^\circ$ ,  $\pm 60^\circ$  faces. If with Orth loss only, the increments are 6.19%, 6.83%, 5.00%. PoseFace18 achieves the best results with both losses used and the rank-1 recognition rates increase 8.89%, 8.04%, 5.80%. The experimental results demonstrate the effectiveness of our proposed Orth loss and PAA loss.

**Effect of the Landmark Module.** Our Orth Loss is a composition of an orthogonal constraint and a loss for pose features. The orthogonal constraint disentangles the original feature  $F_b$  into the identity module and the pose module. To ensure no other information such as identity flow into the pose module, we design a landmark module to extract features  $F_p^L$  which is only related to facial poses, and use  $F_p^L$  to supervise the learning of the pose module. An alternative way is to regress landmark points directly [26] instead of using the landmark module. We compare these two approaches with/without the orthogonal constraint used and present their performances on IJB benchmarks in table. 2. For fair comparisons, we pick 5 different hyperparameters (from 0.001 to 10, multiply 10 each time) when regressing the landmark points and report the best results.

Without the orthogonal constraint involved, the performances of using landmark module and using landmark points are similar (*e.g.*, TARs are the same when FAR=1e-4 on IJB-C, and with a tiny difference on IJB-B). However, when introducing the orthogonal constraint, the performance of using landmark points degrades a lot. Specifically, TAR reduces 0.94% on IJB-B and 1.03% on IJB-C when FAR=1e-4. The possible reason is that when regressing landmarks directly, the learned pose feature  $F_p$  cannot guarantee to be pose-related only. To be more specific, regressing landmark points means  $F_p$  can express landmarks, while cannot prevent other information such as iden-

ity to be embedded in  $F_p$ . This can hurt the recognition performance as valuable identity information is disentangled into the pose module. Compared to the direct regression, our landmark module builds a one-to-one mapping between pose features and facial landmarks. Experimentally, the model obtains 0.89% and 0.60% performance boosts at TAR@FAR=1e-4 on IJB-B and IJB-C and that demonstrates the validity of our design.

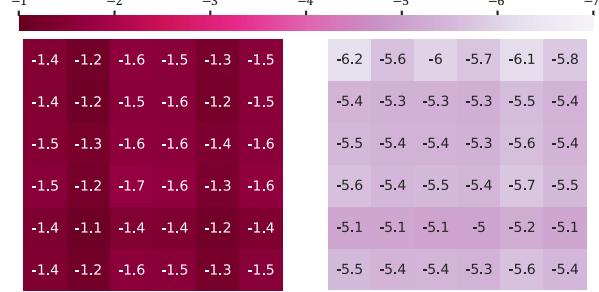
Method	IJB-B (TAR@FAR)			IJB-C (TAR@FAR)		
	1e-5	1e-4	1e-3	1e-5	1e-4	1e-3
Landmark Points only	85.74	92.65	95.25	90.95	94.45	96.51
Landmark Points + Orth	84.66	91.71	94.68	90.91	93.42	95.85
Landmark Module only	86.28	92.55	95.26	91.31	94.45	96.47
Landmark Module + Orth	<b>86.61</b>	<b>93.44</b>	<b>95.92</b>	<b>92.01</b>	<b>95.05</b>	<b>96.97</b>

Table 2: Verification evaluation (%) according to different FARs on IJB-B and IJB-C. Backbone: ResNet50.

**Visualization.** Denote the identity and pose feature for sample  $j$  as  $F_i^j$  and  $F_p^j$ . If distributed in two orthogonal subspaces, the inner product of an identity feature and a pose feature is always 0 (*i.e.*  $\forall(i, j), \langle F_i^j, F_p^k \rangle = 0$ ). We select the first 10 test images from MultiPIE and extract mapped identity as well as pose features. All the features are normalized and figure 6 shows the log10 of inner products of all pairs. With introduce of our orthogonal constraint, the inner products of identity and pose features are significantly decreased by over 4 orders of magnitude. The largest inner product of an identity and a pose feature is only 9.02e-6 in this example, which indicates the orthogonality between two feature spaces and the efficacy of our disentanglement.

We also visualize the distributions of identity features of frontal and profile faces in figure A2 (in supplementary). The visualization further reveals that PoseFace learns more discriminative and pose-invariant features than ArcFace.

**Is PoseFace scalable for real-world applications?** One important question of pose-invariant models is whether it can fit real-world applications. Recent related works usually need extra computations and labels to frontalize faces [17, 5, 52], introduce heavy network modules [49, 22, 46], or require pair-wise inputs [13, 27]. In contrast, our PoseFace works in an end-to-end manner and scalable for real-world applications. The introduced self-disentanglement method avoids pair-wise inputs and therefore can be applied to most existing training datasets. Besides, as introduced modules are all light-weight, PoseFace can be trained with similar configures as ArcFace as illustrated in table 3. Additional GPU memory consumptions are acceptable and the training speed doesn't decrease too much, especially when with large batch sizes. During inference, PoseFace has no difference with ArcFace as only the identity module is needed.



(a) Without the orthogonal constraint  
max product: 8.10e-2  
min product: 1.91e-2

(b) With the orthogonal constraint  
max product: 9.02e-6  
min product: 5.85e-7

Figure 6: We calculate inner products of the mapped identity features and pose features from 10 samples, (a) with and (b) without orthogonal constraints involved. Heatmaps show the **log10** of inner products. (See figure A1 in supplementary for an extended visualization with 100 samples.) **Best viewed in color.**

	Model	Batch Size			
		128	256	512	768
ThroughPut	ArcFace	802	949	986	1001
	PoseFace	729	921	956	963
GPU Memory (MB)	ArcFace	2599	4211	6123	9309
	PoseFace	2645	4329	6695	9798

Table 3: Training comparison of ArcFace and PoseFace on MS1Mv2 dataset with different batch sizes. The experiments are conducted with backbone ResNet50 on 8 1080Ti GPUs.

### 4.3. Results on Benchmarks

In this section, we present the recognition performances of PoseFace on various benchmarks including MultiPIE, CFP-FP, CPLFW and IJB. The “baseline” in all the following tables indicates a model trained by ArcFace with the same network structure with the corresponding PoseFace.

**Evaluations on MultiPIE.** We compare our method with the state-of-the-art alternatives on the MultiPIE benchmark in table 4. The c-CNN [46] uses dynamically activated kernels to form pose-specific models. DFN-L [13] aligns features across various poses by adding deformable convolutional modules. Other methods listed in the table are all based on GAN. From the results, we can conclude that our approach beats current state-of-the-art works in rank-1 recognition rates. For  $\pm 90^\circ$ , our recognition rate is 6.51% higher than the best results reported before (copied from DFN-L [13]). For  $\pm 75^\circ$  and  $\pm 60^\circ$ , the current best results are copied from PIM [52] with numbers 91.20% and 97.70%. Our results are 96.07% and 98.55%, respectively. The great performance improvements show the superiority of our proposed method.

**Evaluations on CFP-FP.** In table 5, we present the performance of the current state-of-the-art on the CFP benchmark. From the values, we can see that our PoseFace

Method	$\pm 90^\circ$	$\pm 75^\circ$	$\pm 60^\circ$	$\pm 45^\circ$	$\pm 30^\circ$	$\pm 15^\circ$
c-CNN [46]	47.26	60.66	74.38	89.02	94.05	96.97
FNM [27]	55.80	81.30	93.70	98.20	99.50	<b>99.90</b>
TP-GAN [17]	64.03	84.10	92.93	<b>98.58</b>	<b>99.85</b>	99.78
DR-GAN [38]	-	-	86.20	90.10	94.00	97.00
PIM [52]	75.00	<b>91.20</b>	<b>97.70</b>	98.30	99.40	99.80
p-CNN [49]	76.96	87.83	92.07	90.34	98.01	99.19
CAPG-GAN [15]	77.10	87.40	93.74	98.28	99.37	99.95
FF-GAN [50]	61.20	77.20	85.20	89.70	92.5	94.60
DFN-L [13]	<b>84.07</b>	88.97	95.16	98.05	99.23	99.58
baseline	81.68	88.03	92.75	96.99	98.77	99.37
PoseFace18(Ours)	<b>90.58</b>	<b>96.07</b>	<b>98.55</b>	<b>99.62</b>	<b>99.97</b>	<b>100</b>

Table 4: Comparative performances on MultiPIE benchmark. Rank-1 recognition rates (%) for different poses are reported. Symbol ‘-’ indicates that the metric is not available. Results of other methods are directly copied from the original papers.

achieves competitive results on the accuracy, EER and AUC. Among the listed methods, DREAM [5] also attempts to extract pose-invariant features by mapping profile features to frontal features. Compared to it, our PoseFace34 achieves better EER (5.09 compared to 6.02, smaller is better) with much fewer layers. Human performance on verifying the profile faces is listed in the last row of table 5. It can be seen that our PoseFace34 almost achieves a human-level ability to recognize profile faces.

Method	Accuracy	EER	AUC
Deep features [30]	84.91 (1.82)	14.97 (1.98)	93.00 (1.55)
TPE [28]	89.17 (2.35)	8.85 (0.99)	97.00 (0.53)
DR-GAN [38]	93.41 (1.17)	- (-)	- (-)
PIM [52]	93.10 (1.01)	7.69 (1.29)	97.65 (0.62)
p-CNN [49]	<b>94.39</b> (1.17)	<b>5.94</b> (0.11)	<b>98.36</b> (0.05)
DREAM18 [5]	- (-)	7.03 (-)	- (-)
DREAM50 [5]	- (-)	6.02 (-)	- (-)
baseline	93.09 (1.55)	6.83 (1.80)	97.78 (0.94)
PoseFace34(Ours)	<b>94.84</b> (1.18)	<b>5.09</b> (1.26)	<b>98.82</b> (0.40)
Human [30]	94.57 (1.10)	5.02 (1.07)	98.92 (0.46)

Table 5: Comparative performances on the CFP benchmark with the Frontal-Profile protocol designed in the original paper. Results (%) are the in the format *mean (standard deviation)* over 10-folds in the frontal-profile protocol. Results of other works are directly copied from original papers.

**Evaluations on CPLFW.** We further evaluate our method on the CPLFW benchmark. Specifically, we re-implement the ArcFace [8] as the baseline, and Softmax, SV-AM-Sofmax [40], SphereFace [21] as the competitors. The results are shown in table 6, from which we can see the similar trends that emerged on previous test datasets. Our PoseFace achieves the best results on the CPLFW benchmark and the mean accuracy increases more than 0.55% compared to the baseline ArcFace.

**Evaluations on IJB-B/IJB-C.** Besides benchmarks specifically designed for large-pose face recognition, we further

Method	Accuracy
Softmax	85.51 (1.81)
SphereFace	84.66 (2.01)
SV-AM-Softmax	81.84 (2.70)
ArcFace	88.10 (1.95)
PoseFace34(Ours)	<b>88.65</b> (1.97)

Table 6: Mean and standard deviation of the accuracy (%) on the CPLFW benchmark under the 10-fold protocol.

evaluate our PoseFace on IJB-B and IJB-C benchmarks, whose images are of large variations on poses, resolutions, occlusions and lighting. Table 7 presents our results. PoseFace surpasses the implemented baseline on all TAR criteria by 1.48% on FAR=1e-5, 0.81% on FAR=1e-4 and 0.56% on FAR=1e-3 on IJB-B, and by 1.04% on FAR=1e-5, 0.7% on FAR=1e-4, and 0.52% on FAR=1e-3 on IJB-C.

Besides our implemented baseline, we also compare our PoseFace with recent related works. We pick VGGFace2 [6], DCN [44], Multicolumn [45], PFE [33], DUL [7] and CircleLoss [36] as other baselines, and present their results in this table. Note that PFE [33] and DUL [7] are two methods specifically designed for low-quality face recognition by introducing the uncertainty learning. Compared to them, PoseFace50 still achieves better results on IJB-C with fewer layers used (50 compared to 64). Specifically, PoseFace50 surpasses best results of the two methods by 2.37% on TAR@FAR=1e-5, 0.44% on TAR@FAR=1e-4 and 0.27% on TAR@FAR=1e-3 respectively. That further shows the superiority of the proposed method.

Method	Backbone	IJB-B (TAR@FAR)			IJB-C (TAR@FAR)		
		1e-5	1e-4	1e-3	1e-5	1e-4	1e-3
VGGFace2 [6]	Res50	67.10	80.40	89.10	74.90	84.60	91.30
DCN [44]	Res50	<b>84.10</b>	<b>93.00</b>	-	88.00	94.40	-
Multicolumn [45]	Res50	70.80	83.10	<b>90.09</b>	77.10	86.20	92.70
PFE [33]	Res64	-	-	-	<b>89.64</b>	93.25	95.49
DUL [7]	Res64	-	-	-	88.18	<b>94.61</b>	<b>96.70</b>
CircleLoss [36]	Res100	-	-	-	89.60	93.95	96.29
baseline	Res50	85.13	92.63	95.36	90.97	94.35	96.45
PoseFace50	Res50	<b>86.61</b>	<b>93.44</b>	<b>95.92</b>	<b>92.01</b>	<b>95.05</b>	<b>96.97</b>

Table 7: Verification evaluation (%) according to different FARs on IJB-B and IJB-C. “-” indicates that the author did not report the performance on the corresponding protocols.

## 5. Conclusions

In this paper, we propose a novel PoseFace framework to handle the large pose variations. Specifically, to remove pose information from the identity features, we utilize the facial landmarks as well as an orthogonal constraint to generate the pose-invariant features. We further propose a pose-adaptive loss to focus training on hard and rare samples (*e.g.*, the profile faces) to address the data imbalance issues.

Extensive experiments on benchmarks present compelling results, which show the superiority of our PoseFace. As a general framework, PoseFace can be further extended with other classification losses besides ArcFace as well as metric learning losses (e.g., triplet loss [29] and N-pair loss [34]). Besides, the idea of combining AutoEncoder and orthogonal spaces can also be applied to other fields to disentangle irrelevant features for specific tasks.

## References

- [1] Adrian Bulat and Georgios Tzimiropoulos. How far are we from solving the 2d & 3d face alignment problem?(and a dataset of 230,000 3d facial landmarks). In *International Conference on Computer Vision*, pages 1021–1030, 2017. 5
- [2] Jie Cao, Yibo Hu, Bing Yu, Ran He, and Zhenan Sun. 3d aided duet gans for multi-view face image synthesis. *IEEE Transactions on Information Forensics and Security*, 14(8):2028–2042, 2019. 3
- [3] Jie Cao, Yibo Hu, Hongwen Zhang, Ran He, and Zhenan Sun. Learning a high fidelity pose invariant model for high-resolution face frontalization. *arXiv preprint arXiv:1806.08472*, 2018. 2
- [4] Jie Cao, Yibo Hu, Hongwen Zhang, Ran He, and Zhenan Sun. Towards high fidelity face frontalization in the wild. *International Journal of Computer Vision*, pages 1–20, 2019. 3
- [5] Kaidi Cao, Yu Rong, Cheng Li, Xiaoou Tang, and Chen Change Loy. Pose-robust face recognition via deep residual equivariant mapping. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 5187–5196, 2018. 1, 2, 4, 5, 7, 8
- [6] Qiong Cao, Li Shen, Weidi Xie, Omkar M Parkhi, and Andrew Zisserman. Vggface2: A dataset for recognising faces across pose and age. In *International Conference on Automatic Face and Gesture Recognition*, pages 67–74, 2018. 8
- [7] Jie Chang, Zhonghao Lan, Changmao Cheng, and Yichen Wei. Data uncertainty learning in face recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 5710–5719, 2020. 8
- [8] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4690–4699, 2019. 1, 2, 3, 4, 5, 8
- [9] Hang Du, Hailin Shi, Dan Zeng, and Tao Mei. The elements of end-to-end deep face recognition: A survey of recent advances. *arXiv preprint arXiv:2009.13290*, 2020. 1
- [10] Ralph Gross, Iain Matthews, Jeffrey Cohn, Takeo Kanade, and Simon Baker. Multi-pie. *Image and Vision Computing*, 28(5):807–813, 2010. 6
- [11] Yandong Guo, Lei Zhang, Yuxiao Hu, Xiaodong He, and Jianfeng Gao. Ms-celeb-1m: A dataset and benchmark for large-scale face recognition. In *European Conference on Computer Vision*, 2016. 5
- [12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. 6
- [13] Mingjie He, Jie Zhang, Shiguang Shan, Meina Kan, and Xilin Chen. Deformable face net: Learning pose invariant feature with pose aware feature alignment for face recognition. In *International Conference on Automatic Face and Gesture Recognition*, pages 1–8. IEEE, 2019. 1, 2, 3, 7, 8
- [14] Ran He, Xiang Wu, Zhenan Sun, and Tieniu Tan. Wasserstein cnn: Learning invariant features for nir-vis face recognition. *IEEE Trans on Pattern Analysis and Machine Intelligence*, 41(7):1761–1773, 2018. 3
- [15] Yibo Hu, Xiang Wu, Bing Yu, Ran He, and Zhenan Sun. Pose-guided photorealistic face rotation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 8398–8406, 2018. 1, 2, 3, 8
- [16] Gary B Huang, Marwan Mattar, Tamara Berg, and Eric Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. 2008. 6
- [17] Rui Huang, Shu Zhang, Tianyu Li, and Ran He. Beyond face rotation: Global and local perception gan for photorealistic and identity preserving frontal view synthesis. In *International Conference on Computer Vision*, pages 2439–2448, 2017. 1, 2, 3, 7, 8
- [18] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *International Conference on Computer Vision*, pages 2980–2988, 2017. 2, 4
- [19] Feng Liu, Ronghang Zhu, Dan Zeng, Qijun Zhao, and Xiaoming Liu. Disentangling features in 3d face shapes for joint face reconstruction and recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 5216–5225, 2018. 3
- [20] Hao Liu, Xiangyu Zhu, Zhen Lei, and Stan Z Li. Adaptive-face: Adaptive margin and sampling for face recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 11947–11956, 2019. 2, 4
- [21] Weiyang Liu, Yandong Wen, Zhiding Yu, Ming Li, Bhiksha Raj, and Le Song. Spheraface: Deep hypersphere embedding for face recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 8
- [22] Iacopo Masi, Stephen Rawls, Gérard Medioni, and Prem Natarajan. Pose-aware face recognition in the wild. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4838–4846, 2016. 1, 2, 7
- [23] Brianna Maze, Jocelyn Adams, James A Duncan, Nathan Kalka, Tim Miller, Charles Otto, Anil K Jain, W Tyler Niggel, Janet Anderson, Jordan Cheney, et al. Iarpa janus benchmark-c: Face dataset and protocol. In *International Conference on Biometrics*, pages 158–165, 2018. 6
- [24] Qiang Meng, Shichao Zhao, Zhida Huang, and Feng Zhou. MagFace: A universal representation for face recognition and quality assessment. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2021. 1
- [25] Xin Ning, Fangzhe Nan, Shaohui Xu, Lina Yu, and Liping Zhang. Multi-view frontal face image generation: A survey. *Concurrency and Computation: Practice and Experience*, page e6147, 2020. 3

- [26] Xi Peng, Xiang Yu, Kihyuk Sohn, Dimitris N Metaxas, and Manmohan Chandraker. Reconstruction-based disentanglement for pose-invariant face recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1623–1632, 2017. [3](#), [6](#)
- [27] Yichen Qian, Weihong Deng, and Jiani Hu. Unsupervised face normalization with extreme pose and expression in the wild. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 9851–9858, 2019. [1](#), [2](#), [3](#), [7](#), [8](#)
- [28] Swami Sankaranarayanan, Azadeh Alavi, Carlos D Castillo, and Rama Chellappa. Triplet probabilistic embedding for face verification and clustering. In *IEEE international conference on biometrics theory, applications and systems (BTAS)*, pages 1–8. IEEE, 2016. [8](#)
- [29] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 815–823, 2015. [1](#), [9](#)
- [30] Soumyadip Sengupta, Jun-Cheng Chen, Carlos Castillo, Vishal M Patel, Rama Chellappa, and David W Jacobs. Frontal to profile face verification in the wild. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1–9. IEEE, 2016. [1](#), [6](#), [8](#)
- [31] Abhishek Sharma, Murad Al Haj, Jonghyun Choi, Larry S Davis, and David W Jacobs. Robust pose invariant face recognition using coupled latent space discriminant analysis. *Computer Vision and Image Understanding*, 116(11):1095–1110, 2012. [2](#)
- [32] Abhishek Sharma, Abhishek Kumar, Hal Daume, and David W Jacobs. Generalized multiview analysis: A discriminative latent space. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2160–2167. IEEE, 2012. [2](#)
- [33] Yichun Shi and Anil K Jain. Probabilistic face embeddings. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 6902–6911, 2019. [8](#)
- [34] Kihyuk Sohn. Improved deep metric learning with multi-class n-pair loss objective. In *Annual Conference on Neural Information Processing Systems*, pages 1857–1865, 2016. [1](#), [9](#)
- [35] Lingxue Song, Dihong Gong, Zhifeng Li, Changsong Liu, and Wei Liu. Occlusion robust face recognition based on mask learning with pairwise differential siamese network. In *International Conference on Computer Vision*, pages 773–782, 2019. [3](#)
- [36] Yifan Sun, Changmao Cheng, Yuhan Zhang, Chi Zhang, Liang Zheng, Zhongdao Wang, and Yichen Wei. Circle loss: A unified perspective of pair similarity optimization. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 6398–6407, 2020. [8](#)
- [37] Linh Tran, Jean Kossaifi, Yannis Panagakis, and Maja Pantic. Disentangling geometry and appearance with regularised geometry-aware generative adversarial networks. *International Journal of Computer Vision*, 127(6):824–844, 2019. [3](#)
- [38] Luan Tran, Xi Yin, and Xiaoming Liu. Disentangled representation learning gan for pose-invariant face recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 10823–10832, 2019. [2](#), [4](#)
- [39] Mengjiao Wang, Zhixin Shu, Shiyang Cheng, Yannis Panagakis, Dimitris Samaras, and Stefanos Zafeiriou. An adversarial neuro-tensorial approach for learning disentangled representations. *International Journal of Computer Vision*, 127(6):743–762, 2019. [3](#)
- [40] Xiaobo Wang, Shuo Wang, Shifeng Zhang, Tianyu Fu, Hailin Shi, and Tao Mei. Support vector guided softmax loss for face recognition. *arXiv preprint arXiv:1812.11317*, 2018. [1](#), [8](#)
- [41] Yitong Wang, Dihong Gong, Zheng Zhou, Xing Ji, Hao Wang, Zhifeng Li, Wei Liu, and Tong Zhang. Orthogonal deep features decomposition for age-invariant face recognition. In *European Conference on Computer Vision*, pages 738–753, 2018. [3](#)
- [42] Cameron Whitelam, Emma Taborsky, Austin Blanton, Brianna Maze, Jocelyn Adams, Tim Miller, Nathan Kalka, Anil K Jain, James A Duncan, Kristen Allen, et al. Iarpa janus benchmark-b face dataset. In *IEEE Conference on Computer Vision and Pattern Recognition Workshop*, pages 90–98, 2017. [6](#)
- [43] Xiang Wu, Huaibo Huang, Vishal M Patel, Ran He, and Zhenan Sun. Disentangled variational representation for heterogeneous face recognition. In *AAAI Conference on Artificial Intelligence*, pages 9005–9012, 2019. [3](#)
- [44] Weidi Xie, Li Shen, and Andrew Zisserman. Comparator networks. In *European Conference on Computer Vision*, pages 782–797, 2018. [8](#)
- [45] Weidi Xie and Andrew Zisserman. Multicolumn networks for face recognition. *arXiv preprint arXiv:1807.09192*, 2018. [8](#)
- [46] Chao Xiong, Xiaowei Zhao, Danhang Tang, Karlekar Jayashree, Shuicheng Yan, and Tae-Kyun Kim. Conditional convolutional neural network for modality-aware face recognition. In *International Conference on Computer Vision*, pages 3667–3675, 2015. [1](#), [2](#), [6](#), [7](#), [8](#)
- [47] Xiaqing Xu, Qiang Meng, Yunxiao Qin, Jianzhu Guo, Chenxu Zhao, Feng Zhou, and Zhen Lei. Searching for alignment in face recognition. In *AAAI Conference on Artificial Intelligence*, 2020. [1](#)
- [48] Dong Yi, Zhen Lei, and Stan Z Li. Shared representation learning for heterogenous face recognition. In *International Conference on Automatic Face and Gesture Recognition*, 2015. [3](#)
- [49] Xi Yin and Xiaoming Liu. Multi-task convolutional neural network for pose-invariant face recognition. *IEEE Transactions on Image Processing*, 27(2):964–975, 2017. [1](#), [2](#), [7](#), [8](#)
- [50] Xi Yin, Xiang Yu, Kihyuk Sohn, Xiaoming Liu, and Manmohan Chandraker. Towards large-pose face frontalization in the wild. In *International Conference on Computer Vision*, pages 3990–3999, 2017. [1](#), [2](#), [8](#)
- [51] Xiao Zhang, Rui Zhao, Yu Qiao, Xiaogang Wang, and Hongsheng Li. Adacos: Adaptively scaling cosine logits for effectively learning deep face representations. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 10823–10832, 2019. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1415–1424, 2017. [1](#), [2](#), [3](#), [8](#)

- [52] Jian Zhao, Yu Cheng, Yan Xu, Lin Xiong, Jianshu Li, Fang Zhao, Karlekar Jayashree, Sugiri Pranata, Shengmei Shen, Junliang Xing, et al. Towards pose invariant face recognition in the wild. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2207–2216, 2018. [1](#), [2](#), [7](#), [8](#)
- [53] Tianyue Zheng and Weihong Deng. Cross-pose lfw: A database for studying cross-pose face recognition in unconstrained environments. *Beijing University of Posts and Telecommunications, Tech. Rep.*, 5, 2018. [6](#)
- [54] Hang Zhou, Jihao Liu, Ziwei Liu, Yu Liu, and Xiaogang Wang. Rotate-and-render: Unsupervised photorealistic face rotation from single-view images. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 5911–5920, 2020. [2](#)
- [55] Zhenyao Zhu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning identity-preserving face space. In *International Conference on Computer Vision*, pages 113–120, 2013. [2](#)

## A. Ablation Study on Parameters $\lambda_1, \lambda_2$

In our work,  $\lambda_1, \lambda_2$  in Eq.(5) are two hyperparameters for the Orth loss, where  $\lambda_1$  amplifies the errors of predicted pose features while  $\lambda_2$  controls the orthogonality. We first let the  $\lambda_1 = 100$  and report recognition results on MultiPIE with different  $\lambda_2$  as shown in table 8. When no orthogonal constraints added, the model performs poorly especially for large pose faces. For example, the recognition rate is only 80.87% and 88.98% for faces with yaws  $\pm 90^\circ, \pm 75^\circ$ . The performances increase significantly when enlarging  $\lambda_2$ . When  $\lambda_2 = 1e5$ , the model achieves overall the best results on the benchmark. Further increase the value exponentially does not hurt the performances too much. Specifically, setting  $\lambda_2 = 1e6$  still achieves a 6.1%, 5.24% boosts for  $\pm 90^\circ, \pm 75^\circ$ , compared to no orthogonal penalty added.

$\lambda_2$	$\pm 90^\circ$	$\pm 75^\circ$	$\pm 60^\circ$	$\pm 45^\circ$	$\pm 30^\circ$	$\pm 15^\circ$
0	80.87	88.98	93.51	97.08	98.43	99.48
1e3	82.04	90.77	94.72	97.56	99.27	99.52
1e4	83.49	92.22	95.85	98.79	88.69	99.87
1e5	<b>87.58</b>	<b>95.41</b>	<b>97.99</b>	99.43	<b>99.98</b>	<b>100.00</b>
1e6	86.97	94.22	97.87	<b>99.58</b>	99.92	99.99

Table 8: Rank-1 recognition rates (%) on MultiPIE for different poses.  $\lambda_1$  is fixed to be 100.

We further let  $\lambda_2 = 1e5$  and train models with different  $\lambda_1$ . Similar phenomenon observed as indicated in table 9. A small  $\lambda_1$  (e.g., 10 in the table) means the disentangled pose features may be incorrect and contain identity features. The performances are consistently high if when  $\lambda_1 \geq 100$  in the table, while over-emphasizing the loss term can also hurt recognition. The best  $\lambda_1$  are around 100 or 200.

According to the results on MultiPIE, we then empirically set  $\lambda_1 = 200, \lambda_2 = 1e5$  in all experiments presented

in the paper.

$\lambda_1$	$\pm 90^\circ$	$\pm 75^\circ$	$\pm 60^\circ$	$\pm 45^\circ$	$\pm 30^\circ$	$\pm 15^\circ$
10	81.24	88.94	92.64	97.31	99.00	99.62
100	87.58	<b>95.41</b>	<b>97.99</b>	99.43	99.98	<b>100.00</b>
200	<b>87.87</b>	94.86	97.75	<b>99.85</b>	<b>99.99</b>	99.99
500	86.93	95.06	97.89	99.61	99.88	99.99
1000	86.82	93.48	97.53	99.39	99.87	99.99

Table 9: Rank-1 recognition rates (%) on MultiPIE for different poses.  $\lambda_2$  is fixed to be 1e5.

## B. Auto-Encoder Structure

The structure of our Auto-Encoder network is summarized in Table 10. The encoder is a sequence of 5 convolution blocks, consisting of a convolution layer, batch normalization layer, ReLU and max pooling. A fully connected layer is applied to the end of the encoder to output the final feature vector. The decoder is exactly the inverse version of the encoder.

Encoder
<code>[conv, 3 × 3, 32], BN, ReLU, pooling</code>
<code>[conv, 3 × 3, 64], BN, ReLU, pooling</code>
<code>[conv, 3 × 3, 128], BN, ReLU, pooling</code>
<code>[conv, 3 × 3, 256], BN, ReLU, pooling</code>
<code>[conv, 3 × 3, 512], BN, ReLU, pooling</code>
<code>[fc, 512], l2 norm</code>

Decoder
<code>[fc, 512 × 4 × 4], reshape</code>
<code>[upsample, 7 × 7], [conv, 3 × 3, 256], BN, ReLU</code>
<code>[upsample, 14 × 14], [conv, 3 × 3, 128], BN, ReLU</code>
<code>[upsample, 28 × 28], [conv, 3 × 3, 64], BN, ReLU</code>
<code>[upsample, 56 × 56], [conv, 3 × 3, 32], BN, ReLU</code>
<code>[upsample, 108 × 108], [conv, 3 × 3, 14], sigmoid</code>

Table 10: Details of our Auto-Encoder structure

## C. Extended Visualization of Figure 6

Figure 7 presents a extended visualization of figure 6 with 100 samples used. The inner products of a pose feature and an identity feature is consistently of a small order of magnitude. That indicates the orthogonality between two feature spaces and demonstrates the efficacy of our disentanglement.

## D. Visualization of Feature Distributions

We visualize deep feature distributions of the baseline (ArcFace) and our PoseFace on MultiPIE in figure 8. Compared to ArcFace, features of our PoseFace are more discriminative in pose variations cases clearly indicated by

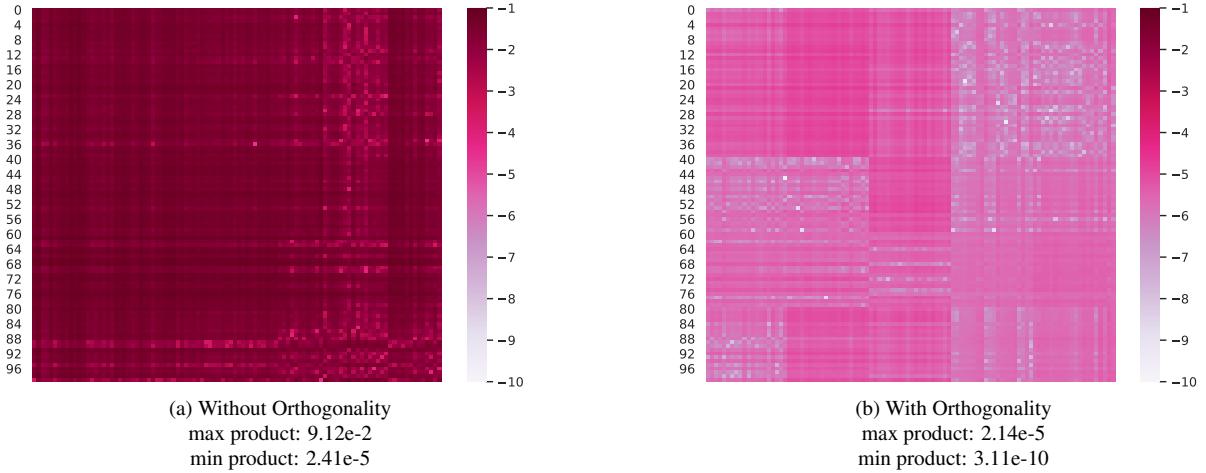


Figure 7: Extended visualization of figure 6. **Best viewed in color.**

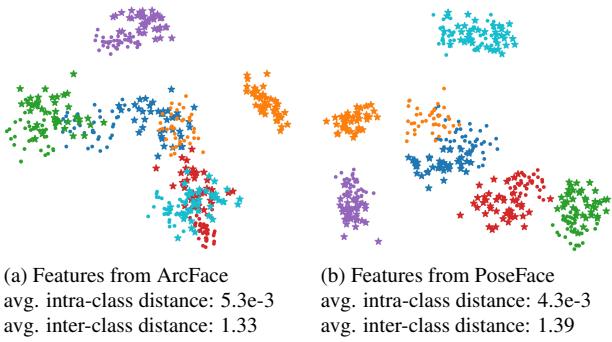


Figure 8: Visualization of the features from baseline (ArcFace) and PoseFace. Each color denotes an identity and each identity contains 40 frontal (yaw angle  $0^\circ$ , marked in dots) and 40 profile faces (yaw angle  $90^\circ$ , marked in stars). The features are projected to the top 2 dimensions via PCA. **Best viewed in color.**

the figure. We further calculate the average intra-class and inter-class distances. Compared to ArcFace, PoseFace decreases the average inter-class distance from 0.0053 to 0.0043 while increases the average intra-class distance from 1.33 to 1.39. In addition, for each identity of PoseFace, the profile face features (marked in stars) are clustered with the corresponding frontal face features (marked in dots) more tightly than that of in ArcFace. To sum up, the visualization reveals the proposed PoseFace learns more discriminative and pose-invariant features than ArcFace.