Supplementary Material for "Improving Face Recognition from Hard Samples via Distribution Distillation Loss"

Yuge Huang*[1], Pengcheng Shen*[1], Ying Tai♯[1], Shaoxin Li♯[1], Xiaoming Liu[2], Jilin Li[1], Feiyue Huang[1], and Rongrong Ji[3]

[1] Youtu Lab, Tencent
[2] Michigan State University
[3] Xiamen University
{yugehuang, quantshen, yingtai, darwinli, jerolinli, garyhuang}@tencent.com, liuxm@cse.msu.edu, rrji@xmu.seu.cn

This supplementary material provides additional details on the following:

- Gradient derivation of Eq. 3 (Paper).
- T-SNE illustrations on feature distribution in different iterations.
- Effects on Numbers of b and Bins.
- Experiments trained by CASIA and tested on CFP-FP with ResNet50 as the backbone.
- Comparisons on IJB-B and IJB-C between ArcFace and our DDL.
- A simple attempt of online DDL compared to the offline version in our submission.
- Time Complexity of the method.

1 Gradient Derivation of Eq. 3 (Paper)

First, we present the detailed derivation of Eq. 3 in our submission. As described in Sec. 3.1 (Paper), we have h_r^+ of the histogram H^+ at each bin as:

$$h_r^+ = \frac{1}{|S^+|} \sum_{(i,j):m_{ij=+1}} \delta_{i,j,r}, \tag{1}$$

where $\delta_{i,j,r}$ is the weight and is defined as:

$$\delta_{i,j,r} = \exp(-\gamma(s_{ij} - t_r)^2). \tag{2}$$

For any s_{ij}, $\frac{\partial h_r^+}{\partial s_{ij}}$ can be obtained as follows:

$$\frac{\partial h_r^+}{\partial s_{ij}} = \frac{-2 \cdot \gamma \cdot \delta_{i,j,r}(s_{ij} - t_r)}{|S^+|}. \tag{3}$$

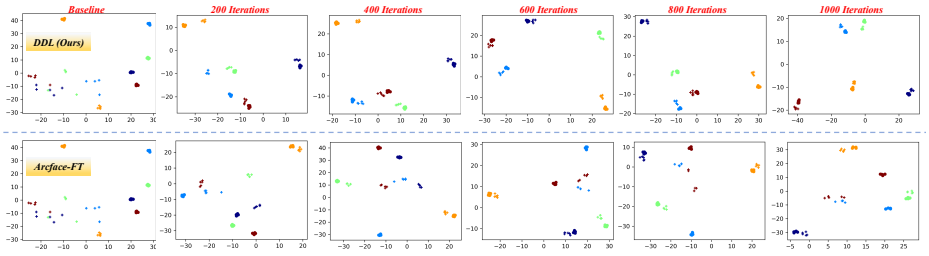


Fig. 1: **T-SNE illustration on feature distribution** in different training iterations on SCface. **Top**: Ours. **Bottom**: Arcface-FT. The same color indicates samples of the same subject. '•' indicates samples captured with the Distance3 (*i.e.*, d_3) setting, and '+' is samples with Distance1 (*i.e.*, d_1). d_1 and d_3 indicate images were captured at distances of 4.2 and 1.0m, respectively. Note that because of the characteristic of T-SNE algorithm, the axes of the same subject may keep changing. Please zoom in to see more details.

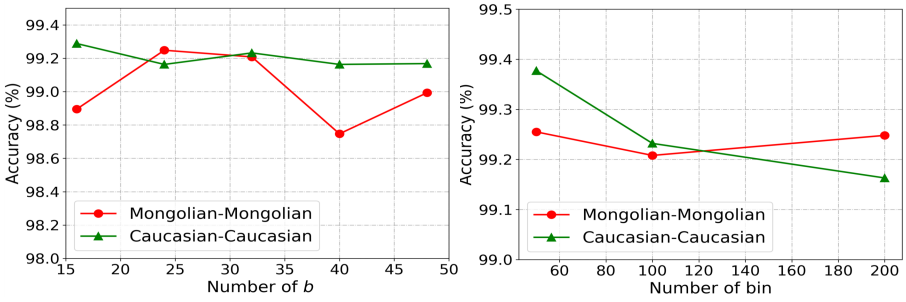


Fig. 2: **Effects on Numbers of b and bins** on COX datasets. Our loss is not sensitive to the batch size or the number of bins.

2 T-SNE Illustrations on Feature Distribution in Different Iterations

As shown in Fig. 1, we compare Arcface-FT and our method, and illustrate the T-SNE visualizations of feature distribution in different training iterations. We randomly select 5 subjects in the test set, and the baseline indicates the original Arcface model. It is shown that: 1) The original Arcface performs well on easy samples (*i.e.*, d_3 setting), but poorly on hard samples (*i.e.*, d_1 setting). 2) After finetuning, Arcface-FT significantly improves the performance on some hard samples, but still fails on some subjects. 3) Compared to Arcface-FT, our method better addresses the hard samples, which makes the easy/hard samples from the same subject more *compact* in the feature space.

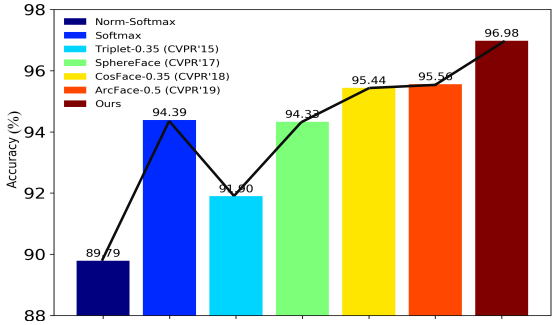
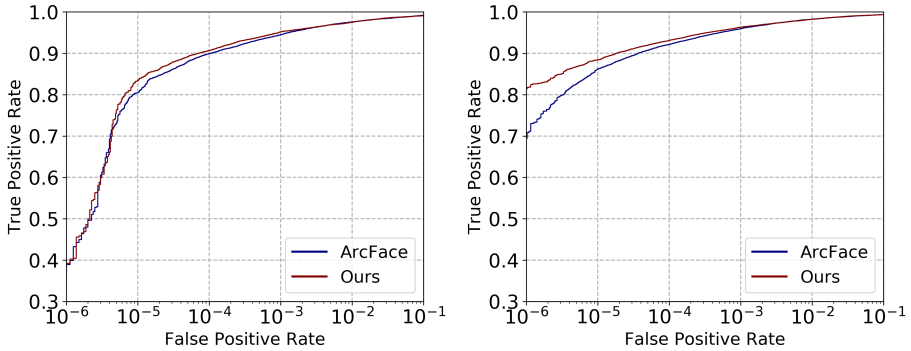


Fig. 3: **Verification results of different loss functions** on CFP-FP. All methods are trained by dataset CASIA with ResNet50 as the backbone.



(a) ROC for IJB-B (b) ROC for IJB-C

Fig. 4: ROC curves of 1:1 verification protocol on the IJB-B and IJB-C datasets. Both ArcFace and ours are trained by VGGFace2 dataset with backbone ResNet50.

3 Effects on Numbers of b and Bins

Since the similarity distributions of positive and negative pairs are estimated within a mini-batch, we conduct tests on COX to explore the effect of the number of b in Fig. 2, and observe that the results remain stable when b varies from 16 to 48 along with the number of bins equals to 100. In addition, we also investigate how the number of bins affects performance by fixing the number of b to be 32. As shown in Fig. 2, the results also remain stable. To sum up, our loss is not sensitive to the numbers of b and bins.

4 Pose on CFP-FP

We compare our loss with other representative losses in face recognition, including Softmax, Norm-Softmax, Triplet loss, SphereFace, CosFace and ArcFace on

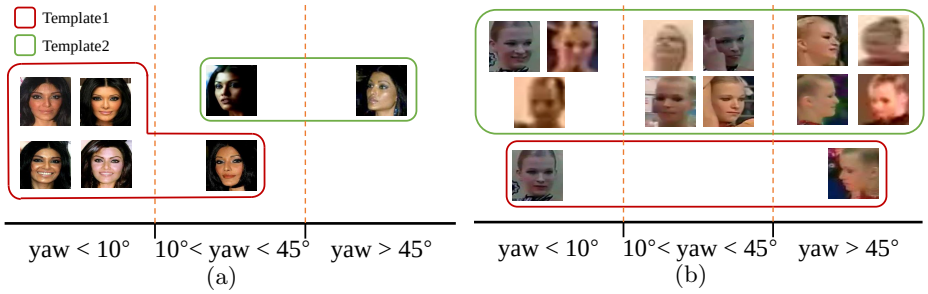


Fig. 5: **Examples of positive template pairs classified correctly by ours while ArcFace fails** at FAR=$1e{-}5$ from IJB-B.

Table 1: **Comparisons among Arcface and two versions of DDL**. We choose VGGFace2 as the large scale dataset and test on IJB-B dataset, while CASIA is adopted as a small scale dataset and CFP-FP is tested.

Methods (%)	IJB-B(1:1)			IJB-B(1:N)				CFP-FP
	FAR=$1e{-}5$	FAR=$1e{-}4$	FAR=$1e{-}3$	FPIR=0.01	FPIR=0.1	Rank1	Rank5	Acc
Arcface	80.5	89.9	94.5	73.1	88.2	93.6	96.5	95.58
Online DDL	81.5	90.7	95.1	75.1	89.0	93.6	96.6	96.82
Offline DDL	83.4	90.7	95.2	76.3	89.5	93.9	96.6	96.98

CFP-FP dataset. All of the methods are trained by the same dataset CASIA with the same backbone ResNet50. For our method, we estimate the pose of each image in CASIA and construct teacher (yaw$<10°$) and student distributions (yaw$>45°$), respectively. Fig. 3 indicates that our loss not only performs better than the triplet-based method (*i.e.*, Triplet, 91.9%), but also better than the softmax-based methods (*i.e.*, Arcface, 95.6%).

5 Comparisons on IJB-B and IJB-C

Fig. 4 shows the ROC curves of our DDL and ArcFace on IJB-B and IJB-C datasets. As we can see, our method achieves better performance and forms an upper envelope compared to ArcFace [4]. Fig. 5 illustrates two examples of positive template pairs that are classified correctly by ours but Arcface fails. Benefiting from the distribution constraint on pose variation, our DDL learns more *discriminant pose-invariant features*, and thus leads to better performance on templates that differ on poses (Fig. 5(a)). Interestingly, when comprehensive variations exist, *e.g.*, resolution (Fig. 5(b)), Arcface again fails while our method shows robustness to some extent.

[4] Results are from the official ResNet100 pre-trained with MS1M: https://github.com/deepinsight/insightface.

6 Simple Attempt of Online DDL

In our submission, the hard samples are defined by the variation of the image, such as pose, resolution, and race (*i.e.*, offline DDL). Here, we simply explore an online version of DDL, which mines the hard samples during training. Specifically, in each iteration, we randomly select 64 classes, in which 8 samples are randomly picked from each class and thus construct a mini-batch with 512 samples. We then sort the samples from the same class with losses during training.

In our online DDL, *the first* 4 *samples with larger losses are defined as hard samples, and the others are easy samples.* We conduct the experiments on both small and large-scale training datasets, *i.e.,* CASIA and VGGFace2, respectively. As shown in Tab. 1, our simple attempt of online DDL is slightly inferior to the offline DDL in our submission, but is still superior to ArcFace, which demonstrates the effectiveness of the idea of DDL, even when easy and hard samples are defined in different ways. The nature of our method lies in the distillation between two similarity distributions constructed from easy and hard samples, it's also interesting future direction to find out a better solution to define easy and hard samples.

7 Time Complexity

Compared with conventional finetuning, the only additional operation in our training is to construct the positive and negative pairs in each mini-batch. The positive pairs are constructed offline, and thus has little influence on the training speed. For negative pairs, since we adopt online hard sample mining to grab the hardest b pairs from the total $b \times (b-1)/2$ negative pairs for each distribution, our training step is slightly slower than conventional finetuning, which is still fast. Specifically, with the same environment, conventional finetuning on COX dataset costs 39 minutes for training, while ours costs 43 minutes. Hence, our DDL only has small effects on the training process and has **NO** influence on inference.