Learning Compatible Embeddings  
(Supplementary Material)

Qiang Meng  Chixiang Zhang  Xiaoqiang Xu  Feng Zhou  
Algorithm Research, AiBee Inc.  
{qmeng, cxzhang, xiaoqiangxu, fzhou}@aibee.com

1. Compatible Training

Changes of feature dimensions.

| Model   | Feature Dimension | Veri. Acc. | Cross Veri. Acc. | Perf. Upgrade |
|---------|-------------------|------------|------------------|---------------|
|         |                   |            |                  |               |
| ϕ₁      | 256               | 93.99      | -                | -             |
| ϕ₂      | 512               | 93.94      | 93.53            | 93.85         |
| ϕ₃      | 512               | 93.58      | 93.77            | 94.27         |
| ϕ₂upper | 512               | 94.18      | -                | -             |

Table 1: 1:1 verification TAR (%@FAR=1e-4) on the IJB-C dataset [3] with increasing feature dimensions.

| Model   | Feature Dimension | Veri. Acc. | Cross Veri. Acc. | Perf. Upgrade |
|---------|-------------------|------------|------------------|---------------|
|         |                   |            |                  |               |
| ϕ₁      | 512               | 94.18      | -                | -             |
| ϕ₂      | 256               | 90.49      | 91.59            | 90.29         |
| ϕ₃      | 256               | 93.69      | 92.40            | 94.02         |
| ϕ₂upper | 256               | 93.99      | -                | -             |

Table 2: 1:1 verification TAR (%@FAR=1e-4) on the IJB-C dataset [3] with decreasing feature dimensions.

Changes of feature dimensions can be applied on increasing or decreasing feature dimensions from the old to the new model. For the scenario of increasing feature dimensions, 256 and 512 are used as the feature dimensions for the old and the new model, respectively. We reverse the feature dimensions of the old and new model when experimenting decreasing feature dimensions. Tab. 1 represents the results of dimension increasing, where our proposed LCE framework \( \phi_{lce}^2 \) is compared with \( \phi_{bct}^2 \) and \( \phi_{l2}^2 \). \( \phi_{l2}^2 \) acts negatively on performance and upgrade gains, and \( \phi_{lce}^2 \) performs even worse on both criterions. Compared to them, our approach \( \phi_{lce}^2 \) earns a much higher upgrade gain while persisting a positive performance gain.

Similar results are represented in Tab. 2 for dimension decreasing. \( \phi_{l2}^2 \) ends up with catastrophic scores on performance and upgrade gains. In contrast, \( \phi_{lce}^2 \) presents considerable superiority on both criterions. Since \( \phi_{bct}^2 \) is not capable of dimension decreasing, results of \( \phi_{bct}^2 \) are excluded in Tab. 2. This further emphasizes the flexibility of our LCE framework that is capable of both dimension increasing and decreasing scenarios.

Multi-model and sequential compatibility.

| Model  | ϕ₁   | ϕ₂   | ϕ₃   |
|--------|------|------|------|
| ϕ₁      | 91.00 | 91.80 | 91.87 |
| ϕ₂      | -     | 93.05 | 93.65 |
| ϕ₃      | -     | -     | 94.30 |

Table 3: 1:1 verification TAR (%@FAR=1e-4) on the IJB-C dataset [3] with sequential changes on training datasets.

Multi-model and sequential compatibility is utilized to where three or more different models are required to be compatible with each other, which is commonly exists in industrial scenarios such as performing sequential model upgrades. To verify sequential compatibility, three versions of models \( \phi^1, \phi^2, \phi^3 \) are trained with 25%, 50%, 100% identities from MS1Mv2 [1] dataset, respectively. \( \phi^1 \) is viewed as the initial version and thus trained without compatibility constraints. We endow \( \phi^2 \) with LCE constraints that guarantee compatibility with \( \phi^1 \), and \( \phi^3 \) with LCE constraints that guarantee compatibility with \( \phi^2 \). Self-verifications are implemented on \( \phi^1, \phi^2 \) and \( \phi^3 \) themselves, whose results are considered as lower/upper bound for cross-model verifications. Cross-model verifications are performed between all possible permutations of model pairs from \( \phi^1, \phi^2 \) and \( \phi^3 \). Results are represented in Tab. 3. Each TAR of cross-model stays between the lower and upper bound from its model pair, which indicates that \( \phi^1, \phi^2 \) and \( \phi^3 \) are compatible with each other.

Transformation module and compatible directions.

In this section we extend Tab. 7 of Sec. 4 with transformation module introduced during LCE training, aiming at verifying the effectiveness of the transformation module as well as model compatibility for each compatible direction.
Table 4: 1:1 verification TAR (%@FAR=1e-4) on the IJB-C dataset [3] with different compatible directions.

| Model   | Backbone   | Transformation | Veri. Acc. | Cross Veri. Acc. | Perf. Gain (%) | Upgrade Gain (%) |
|---------|------------|----------------|------------|------------------|----------------|------------------|
| ϕ^1    | ResNet50   | -              | 94.18      | -                | -              | -                |
| ϕ^2_lce, ResNet18 | -      | 89.71          | 92.81      | -                | -              | -114.91 -35.22   |
| ϕ^2_lce, ResNet18 | ✓       | 90.46          | 92.09      | 92.76            | -95.63         | -36.50           |
| ϕ^2_upper, ResNet18 | -          | 90.29          | 0.01       | -                | -100.00        | -                |
| ϕ^2_lce, MobileFace | -     | 87.84          | 91.31      | -                | -105.66        | -47.50           |
| ϕ^2_lce, MobileFace | ✓     | 88.83          | 89.76      | 91.80            | -89.17         | -39.67           |
| ϕ^2_upper, MobileFace | -       | 88.18          | 0.00       | -                | -100.00        | -                |
| ϕ^2_lce, ResNet100 | -           | 94.64          | 95.07      | -                | +46.94         | +90.82           |
| ϕ^2_lce, ResNet100 | ✓       | 94.87          | 94.76      | 95.04            | +70.41         | +87.76           |
| ϕ^2_upper, ResNet100 | -       | 95.16          | 0.03       | -                | +100.00        | -                |

Table 5: Mean average precision (mAP) (%) on Market-1501.

| Model   | mAP | ϕ^1 | ϕ^2_upper | ϕ^2_lce | ϕ^1, ϕ^2_upper | ϕ^1, ϕ^2_lce | ϕ^1, ϕ^2_upper, ϕ^2_lce |
|---------|-----|-----|-----------|---------|----------------|---------------|-------------------------|
|         | 76.9 | 86.2 | 86.1      | 70.0    | 77.4 (+7.4)    |               |                         |

The results have demonstrated the efficacy of our method.

3. Visualization

To further study the effects of our LCE constraints, we sample 8 classes from MS1MV2 [1] and visualize one of the classes in Fig. 1. Models used to extract those features are chosen from Tab. 5 of Sec. 4.5 where LCE is conducted in the direct compatible method. For Fig. 1a, the new model is ϕ^2_lce, and serves as a baseline. For Fig. 1b, we use ϕ^2_lce as the new model. ϕ^1 is the old model for both two figures. Average intra-class distances are calculated upon the 8 classes as Sec. 4.3 mentioned. Tab. 6 represents the intra-class distances of three types of features where ϕ^2_lce produces a smaller intra-class distance than ϕ^2_lce and the original old model ϕ^1, and this shows the capability of our LCE framework to shrink distributions of features in the same class.

A new metric called average pair-wise distance d_{pw} (·) is
introduced to measure the expected distance between each feature pair of the 8 classes by calculating $d_{pw}(\phi^1, \phi^2) = \frac{1}{N} \sum_{i=1}^{N} \left( \frac{f^1_i}{\|f^1_i\|} - \frac{f^2_i}{\|f^2_i\|} \right)^2$, where $f^1_i, f^2_i$ represents the $i^{th}$ feature pair from model $\phi^1, \phi^2$ respectively. Fig. 1 presents the average pair-wise distances of two types of feature pairs where $d_{pw}(\theta_{IRF}, \phi^1)$ has a greater value than $d_{pw}(\theta_{L2}, \phi^1)$. The results indicate that our method works in a point-to-set scheme that provides flexibility for feature locations.

References

[1] 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.

[2] Xinqian Gu, Hong Chang, Bingpeng Ma, Hongkai Zhang, and Xilin Chen. Appearance-preserving 3d convolution for video-based person re-identification. In European Conference on Computer Vision, 2020.

[3] 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. IEEE, 2018.

[4] Liang Zheng, Liyue Shen, Lu Tian, Shengjin Wang, Jingdong Wang, and Qi Tian. Scalable person re-identification: A benchmark. In International Conference on Computer Vision, 2015.