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	Veri.	Cross Veri. Acc.			Perf.	Upgrade
	Dimension	Acc.	Direct	Backward	Froward	Gain (%)	Gain (%)
ϕ^1	256	93.99	-	-	-		
ϕ^2_{l2}	512	93.94	-	93.53	93.85	-26.32	-73.68
ϕ^2_{bct}	512	93.58	93.77	-	-	-215.79	-115.79
ϕ^2_{lce}	512	94.01	-	91.85	94.27	+10.53	+147.37
ϕ^2_{upper}	512	94.18	-	-	-	+100.00	-

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

Model	Feature	Veri.	Cross Veri. Acc.		Perf.	Upgrade
	Dimension	Acc.	Backward	Froward	Gain (%)	Gain (%)
ϕ^1	512	94.18	-	-	-	
ϕ^2_{l2}	256	90.49	91.59	90.29	-1942.11	-1363.16
ϕ^2_{lce}	256	93.69	92.40	94.02	-257.89	-84.21
ϕ^2_{upper}	256	93.99	-	-	-100.00	-

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 the new model when experimenting decreasing feature dimensions. Tab. 1 represents the results of dimension increasing, where our proposed LCE framework ϕ^2_{lce} is compared with ϕ^2_{bct} and ϕ^2_{l2}. ϕ^2_{l2} acts negatively on performance and upgrade gains, and ϕ^2_{bct} performs even worse on both criterions. Compared to them, our approach ϕ^2_{lce} earns a much higher upgrade gain while persisting a positive performance gain.

Similar results are represented in Tab. 2 for dimension decreasing. ϕ^2_{l2} ends up with catastrophic scores on performance and upgrade gains. In contrast, ϕ^2_{lce} presents considerable superiority on both criterions. Since ϕ^2_{bct} is not capable of dimension decreasing, results of ϕ^2_{bct} 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.

	ϕ^1	ϕ^2	ϕ^3
ϕ^1	91.00	91.80	91.87
ϕ^2	-	93.05	93.65
ϕ^3	-	-	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 ϕ^1, ϕ^2, ϕ^3 are trained with 25%, 50%, 100% identities from MS1Mv2 [1] dataset, respectively. ϕ^1 is viewed as the initial version and thus trained without compatibility constraints. We endow ϕ^2 with LCE constraints that guarantee compatibility with ϕ^1, and ϕ^3 with LCE constraints that guarantee compatibility with ϕ^2. Self-verifications are implemented on ϕ^1, ϕ^2 and ϕ^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 ϕ^1, ϕ^2 and ϕ^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 ϕ^1, ϕ^2 and ϕ^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.

Model	Backbone	Transformation	Veri.	Cross Veri. Acc.			Perf.	Upgrade
			Acc.	Direct	Backward	Forward	Gain (%)	Gain (%)
ϕ^1	ResNet50	-	94.18	-	-	-	-	-
ϕ^2_{lce}	ResNet18		89.71	92.81	-	-	-114.91	-35.22
$\phi^2_{lce\_t}$	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
$\phi^2_{lce\_t}$	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
$\phi^2_{lce\_t}$	ResNet100	✓	94.87	-	94.76	95.04	+70.41	+87.76
ϕ^2_{upper}	ResNet100	-	95.16	0.03	-	-	+100.00	-

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

Upgraded models trained with both LCE constraints and the transformation module are noted as $\phi^2_{lce\_t}$. Results of introducing transformation module are summarized in Tab. 4 where the performance gain of each $\phi^2_{lce\_t}$ witnesses an increment of about 20% compared with ϕ^2_{lce}, which infers a positive impact.

Tab. 4 also reveals model compatibility achievements of direct, backward and forward compatible directions. Observed from each backbone, cross verification accuracy of the backward direction performs slightly worse than the direct compatible method but is still considerable, while the forward compatible method has a competitive (or even better) cross verification accuracy compared with the direct manner. This indicates that our framework guarantees model compatibility for each compatible direction.

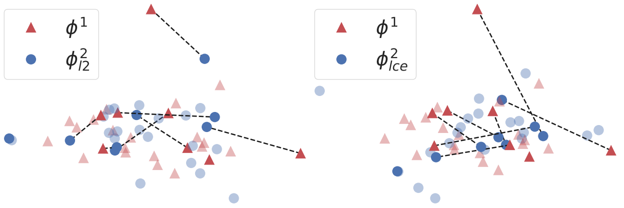
2. Experiments on Person Re-Identification

Besides the face recognition, we also conduct experiments on person re-identification to further reveal the generalization of our method. Our implementation is based on a public repository[1] provided by authors of Gu *et al*. [2]. We use the market-1501 dataset [4] consisting of 1501 identities and 32217 images. Following the protocol in Zheng *et al*. [4], 751 identities are reserved for training and the remaining 750 identities are used for testing. We use the first half of identities to train the old model and the full dataset to train the new model. The results are reported in Tab. 5.

Model	ϕ^1	ϕ^2_{upper}	ϕ^2_{lce}	(ϕ^1, ϕ^2_{upper})	(ϕ^1, ϕ^2_{lce})
mAP	76.9	86.2	86.1	70.0	77.4 (+7.4)

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

The old model ϕ^1 achieves a mAP of 76.9% and the upper bound model gets 86.2%. If using LCE, the performance degrades an acceptable value of 0.1%. For the compatibility, our model achieves 77.4%, which is 7.4% better than the model trained without compatibility constraints.



(a) (b)

Figure 1: Visualizations of deep features from a same class. (a) Deep features from ϕ^1 and $\phi^1_{l_2}$. The average pair-wise distance $d_{pw}(\phi^2_{l_2}, \phi^1)$ is 0.230. (b) Deep features from ϕ^1 and ϕ^2_{lce}. The average pair-wise distance $d_{pw}(\phi^2_{lce}, \phi^1)$ is 0.409. **Best viewed in color**.

The results have demonstrated the efficacy of our method.

3. Visualization

	ϕ^1	ϕ^2_{l2}	ϕ^2_{lce}
intra-class	0.437	0.362	0.344

Table 6: Average intra-class distances of features from different models.

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 $\phi^2_{l_2}$ 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_{l2} 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 $d_{pw}(\cdot)$ is

[1] https://github.com/guxinqian/Simple-ReID

introduced to measure the expected distance between each feature pair of the 8 classes by calculating $d_{pw}(\phi^1, \phi^2) = \frac{1}{N}\sum_{i=1}^{N}(\frac{f_i^1}{\|f_i^1\|} - \frac{f_i^2}{\|f_i^2\|})^2$, where f_i^1, f_i^2 represents the i^{th} feature pair from model ϕ^1, ϕ^2 respectively. Fig. 1 presents the average pair-wise distances of two types of feature pairs where $d_{pw}(\phi_{lce}^2, \phi^1)$ has a greater value than $d_{pw}(\phi_{l2}^2, \phi^1)$. 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.