Towards Backward-Compatible Representation Learning

Yantao Shen*   Yuanjun Xiong   Wei Xia   Stefano Soatto
AWS/Amazon AI

ytshen@link.cuhk.edu.hk,   {yuanjx, wxia, soattos}@amazon.com

Abstract

We propose a way to learn visual features that are compatible with previously computed ones even when they have different dimensions and are learned via different neural network architectures and loss functions. Compatible means that, if such features are used to compare images, then "new" features can be compared directly to "old" features, so they can be used interchangeably. This enables visual search systems to bypass computing new features for all previously seen images when updating the embedding models, a process known as backfilling. Backward compatibility is critical to quickly deploy new embedding models that leverage ever-growing large-scale training datasets and improvements in deep learning architectures and training methods. We propose a framework to train embedding models, called backward-compatible training (BCT), as a first step towards backward compatible representation learning. In experiments on learning embeddings for face recognition, models trained with BCT successfully achieve backward compatibility without sacrificing accuracy, thus enabling backfill-free model updates of visual embeddings.

1. Introduction

Visual classification in an "open universe" setting is often accomplished by mapping each image onto a vector space using a function ("model") implemented by a deep neural network (DNN). The output of such a function in response to an image is often called its "embedding" [7, 30]. Dissimilarity between a pair of images can then be measured by some type of distance between their embedding vectors. A good embedding is expected to cluster images belonging to the same class in the embedding space.

As images of a new class become available, their embedding vectors are used to spawn a new cluster in the open universe, possibly modifying its metric to avoid crowding, in a form of "life-long learning." This process is known as
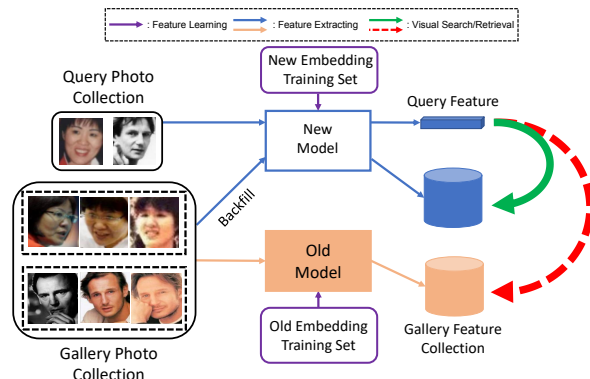
(a) Model update without backward compatible representation.
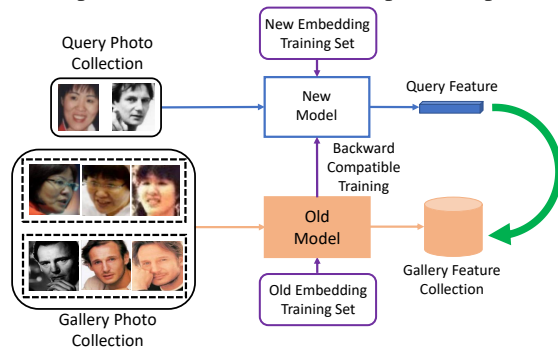
(b) Model update with backward compatible representation.

Figure 1: Without backward compatible representation, to update the embedding model for a retrieval/search system, all previously processed gallery features have to be recomputed by the new model (backfilling), as the new embedding cannot be directly compared with the old one. With a backward compatible representation, direct comparison becomes possible, thus eliminating the need to backfill.

indexing. It is common in modern applications to have millions, in some cases billions, of images indexed into hundreds of thousands to millions of clusters. This collection of images is usually referred to as the *gallery set*. A common use for the indexed gallery set is to identify the closest clusters to one or a set of input images, a process known as

---

visual search or *visual retrieval*. The set of input images for this task is known as the *query set*. Besides the gallery and the query set, there is usually a separate large repository of images used for training the embedding model [32, 29], called the *embedding training set*.

As time goes by, the datasets grow and the quality of the embeddings improves with newly trained models [36, 33, 5, 34]. However, to harvest the benefits of new models, one has to use the new models to *re-process all images* in the gallery set to generate their embedding and re-create the clusters, a process known as "backfilling" or "re-indexing."* In this paper, we aim to design a system that enables new models to be deployed without having to re-index existing image collections. We call such a system *backfill-free*, the resulting embedding *backward-compatible representation*, and the enabling process *backward-compatible training* (BCT).

We summarize our contributions as follow: 1) We formalize the problem of *backward compatible representation learning* in the context of open-set classification, or visual retrieval. The goal is to enable new models to be deployed without having to re-process the previously indexed gallery set. The core of this problem is backward compatibility, which requires a new embedding's output to be usable for comparisons against the old embedding model without compromising recognition accuracy. 2) We propose a novel backward compatible training (BCT) approach by adding an influence loss, which uses the learned classifier of the old embedding model in training the new embedding model. 3) We achieve backward compatible representation learning with minimal loss of accuracy, enabling backfill-free updates of the models. We empirically verify that BCT is robust against multiple changing factors in training the embedding models, *e.g.*, neural network architectures, loss functions, and data growth. Finally, 4) we show that compatibility between multiple models can be attained via chain-like pairwise BCT training.

1.1. Related Work

Embedding learning and open set recognition. Open-set visual recognition [24, 25] is relevant to retrieval [6, 3], face recognition [28, 32] and person re-identification [13, 45, 1]. Common approaches involve extracting visual features to instantiate test-time classifiers [27]. Deep neural networks (DNNs) are widely applied to 1) learn embedding models using closed-world classification as a surrogate task [12, 28], using various forms of loss functions [34, 33, 38] and supervision methods [9, 12] to im-

prove generalization 2) perform metric learning [20] enforcing affinity for pairs [26, 1] or triplets [22, 8] of representations in the embedding space. Specifically, [20] learns a single metric that is compatible for all tasks in a multi-task learning setting. Supervising representation learning with classifier weights from other versions of models was proposed in [37] for the task of unsupervised representation learning.

Learning across domains and tasks. In domain adaptation [19, 4, 35], techniques such as MMD [40] and related methods [31, 17, 40], can be used to align the (marginal) distribution of the new and old classes, included those trained adversarially [10]. Knowledge distillation in [9, 15] trains new models to learn from existing models, but, unlike backward compatible representation learning, knowledge distillation does not require the embedding of the new model and the existing one to be compatible in inference. Continual learning [21, 15], transfer learning [2], and life-long learning [23, 11] all deal with the cases where an existing model evolves over time. In [15], model distillation is used as a form of regularization when introducing new classes. In [23], old class centers are used to regularize samples from the new classes. Hou *et al*. [11] proposed a framework for learning a unified classifier in the incremental setting. In [2], the authors designed a re-training loss function. Methods addressing catastrophic forgetting [15] are most closely related to our work, as a common reason for forgetting is the changing of the visual embedding for the subsequent classifiers. The problem we are addressing differs in that we aim to achieve backward compatibility between any pair of old model and new model. The new model is not required to be initialized by nor share a similar network architecture as the old model.

Compatible representations. In [14], the authors discuss the possible mapping between feature vectors from multiple models trained on the same dataset; [44, 42, 43] introduce a design where multiple models with different channel widths but the same architecture share common subsets of parameters and representation, which implicitly imposes compatibility among representations from different models. we propose an approach to solve the problem of backward-compatibility in deep learning, in the sense defined in the previous section. We focus on open-universe classification using metric discriminant functions.

2. Methodology

We first formulate the problem of backward compatible representation learning, then describe a backward compatible training approach and its implementations.

2.1. Problem Formulation

As a prototypical application, we use the following case of a photo collection $\mathcal{D}$, serving the role of the *gallery*.

---

*The reader may have experienced this process when upgrading photo collection software, whereby the search feature is unavailable until the software has re-indexed the entire collection. This is a minor inconvenience in a personal photo collection, but for large-scale galleries, the cost of time and computation may be prohibitive, thus hampering the continuous update potential of the system.

$\mathcal{D}$ is grouped into a number of classes or identities $Y = \{y_1, \ldots, y_N\}$. We have an embedding model ϕ_{old} that maps each image x onto an embedding vector $z = \phi_{\text{old}}(x) \in \mathbb{R}^{K_{\text{old}}}$ with $x \in \mathcal{D}$. The embedding model ϕ_{old} is trained on an embedding training set, $\mathcal{T}_{\text{old}}$. The embedding of any image produced by $\mathcal{D}$ can then be assigned to a class through some distance $d : \mathbb{R}^K \times \mathbb{R}^K \to \mathbb{R}_+$. In the simplest case, dropping the subscript "old,", each class in Y is associated with a "prototype" or cluster center ϕ_i, $i \in Y$. The vector ϕ_i for the class i can be obtained by a set function $\phi_i = S(\{\phi(x)\}_{y(x)=i})$, where $y(x)$ is the corresponding class label of an image $x \in \mathcal{D}$. Common choices of the set function S include averaging and attention models [41]. A test sample x is assigned to the class $y = \arg\min_{i \in Y} d(\phi(x), \phi_i) \in Y$. Later, a new model ϕ_{new} with K_{new}-dimensional embedding vectors becomes available, for instance trained with additional data in the new embedding training set, $\mathcal{T}_{\text{new}}$ ($\mathcal{T}_{\text{new}}$ can be a superset of $\mathcal{T}_{\text{old}}$), or using a different architecture. The new embedding ϕ_{new} is potentially living in a different embedding space and it is possible that $K_{\text{new}} \neq K_{\text{old}}$.

To harvest the benefit of the new embedding model ϕ_{new}, we wish to use ϕ_{new} to process any new images that in the gallery set , $\mathcal{D}$, as well as images for the query set. Since the gallery set could get additional images and clusters, we denote it as $\mathcal{D}_{\text{new}} = \mathcal{D} \cup \{x | y = N + 1, \ldots, N_{\text{new}}\}$, where N_{new} is the number of clusters in $\mathcal{D}_{\text{new}}$. Then, the question becomes how to deal with images in $\mathcal{D}$. In order to make the system backfill-free, we wish to directly use the already computed embedding from ϕ_{old} for these images and obtain $\{\phi_i\}_{i \leq N}$. Our goal, then, is to design a training process for the new embedding model ϕ_{new} so that any test images can be assigned to classes, new or old, in $\mathcal{D}_{\text{new}}$, without the need to compute $\phi_{\text{new}}(\mathcal{D})$, *i.e.*, to backfill. The resulting embedding ϕ_{new}, is then backward-compatible with ϕ_{old}.

2.2. Criterion for Backward Compatibility

In a strict sense, a model ϕ_{new} is backward compatible if

$$
\begin{aligned}
d(\phi_{\text{new}}(x_i), \phi_{\text{old}}(x_j)) &\geq d(\phi_{\text{old}}(x_i), \phi_{\text{old}}(x_j)), \\
&\forall (i,j) \in \{(i,j) | y_i \neq y_j\}. \\
\text{and,} & \\
d(\phi_{\text{new}}(x_i), \phi_{\text{old}}(x_j)) &\leq d(\phi_{\text{old}}(x_i), \phi_{\text{old}}(x_j)), \\
&\forall (i,j) \in \{(i,j) | y_i = y_j\} (1)
\end{aligned}
$$

where $d(\cdot, \cdot)$ is a distance in the embedding space. These constraints formalize the fact that the new embedding, when used to compare against the old embedding, must be at least as good as the old one in separating images from different classes and grouping those from the same classes. Note that the solution $\phi_{\text{new}} = \phi_{\text{old}}$ is backward compatible. This trivial solution is excluded if the architectures are different, which is usually the case when updating a

model. Although, to simplify the discussion, we assume the embedding dimensions for the two models to be the same ($K_{\text{new}} = K_{\text{old}}$), our method is more general and not bound by this assumption.

The criterion introduced in Eq. 1 entails testing the gallery exhaustively, which is intractable at large scale and in the open-set setting. On the other hand, suppose we have an evaluation metric, $M(\phi_q, \phi_g; \mathcal{Q}, \mathcal{D},)$ on some testing protocols, *e.g.*, true positive identification rates for face search, where $\mathcal{Q}$ denotes the query set, $\mathcal{D}$ denotes the gallery set, and we use ϕ_q for extracting the query set feature and ϕ_g for the gallery set. Then, the **empirical compatibility criterion**, for the application can be defined as

$$
M(\phi_{\text{new}}, \phi_{\text{old}}; \mathcal{Q}, \mathcal{D}) > M(\phi_{\text{old}}, \phi_{\text{old}}; \mathcal{Q}, \mathcal{D}). \quad (2)
$$

This criterion can be interpreted as follows: In an open-set recognition task with a fixed query set and a fixed gallery set, when the accuracy using ϕ_{new} for queries without back-filling gallery images surpasses that of using ϕ_{old}, we consider backward compatibility achieved and backfill-free update feasible. Note that simply setting ϕ_{new} to ϕ_{old} will not satisfy this criterion.

2.3. Baseline and paragon

A naive approach to train the model ϕ_{new} to be compatible with ϕ_{old}, assuming they have the same dimension, is to minimize the ℓ^2 distance between their embeddings computed on the same images. This is enforced for every image in $\mathcal{T}_{\text{old}}$, which is used to train ϕ_{old}. This criterion can be framed as an additive regularizer R for the empirical loss $L(\phi_{\text{new}})$ when training the new embedding as

$$
\phi_{\text{new}} = \arg\min_\phi L(\phi, \mathcal{T}_{\text{new}}) + \lambda R(\phi), \text{ where}
$$

$$
R(\phi) = \sum_{x \in \mathcal{T}_{\text{old}}} \frac{1}{2} \|\phi(x) - \phi_{\text{old}}(x)\|_2^2. \quad (3)
$$

We label the solution of the problem above $\phi_{\text{new}-\ell^2}$. Note that ϕ_{old} will be fixed during training of $\phi_{\text{new}-\ell^2}$. As we show in Sect. 3.4, $\phi_{\text{new}-\ell^2}$ does **not** satisfy Eq. (2) and it will not converge to ϕ_{old}, since the training set has been changed to $\mathcal{T}_{\text{new}}$. So, this naive approach cannot be used to obtain a backward compatible representation.

On the other hand, performing the backfill on $\mathcal{D}$ with the model ϕ_{new}, trained without any regularization, can be taken as a paragon. Since the embedding for $\mathcal{D}$ is recomputed, we can fully enjoy the benefit of ϕ_{new} albeit at the cost of reprocessing the gallery. This sets the upper bound of accuracy for the backfill-free update, and thus the upper bound of the update gain.

2.4. Backward Compatible Training

We now focus on backward compatible training for classification using the cross-entropy loss. Let Φ be a model

parametrized by two disjoint sets of weights, w_c and w_ϕ. The first parametrizes the classifier κ, or the "head" of the model, whereas the second parametrizes the embedding ϕ, so that $\Phi(x) = \kappa_{w_c}(\phi_{w_\phi}(x))$. Now, the cross-entropy loss can be written as

$$L(w_c, w_\phi; \mathcal{T}) = \sum_{(x_i, y_i) \in \mathcal{T}} -\log \kappa_{w_c}(\phi_{w_\phi}(x_i))_{y_i}. \quad (4)$$

Note that the classifier κ_{w_c} can take many forms, from the simple SoftMax [28, 12] to recently proposed alternatives [16, 33, 34]. The old model ϕ_{old} is thus obtained by solving

$$w_{c\ \mathrm{old}}, w_{\phi\ \mathrm{old}} = \arg\min_w L(w_c, w_\phi; \mathcal{T}_{\mathrm{old}}). \quad (5)$$

As for the new model ϕ_{new}, while ordinary training would yield

$$w_{c\ \mathrm{new}}, w_{\phi\ \mathrm{new}} = \arg\min_w L(w_c, w_\phi; \mathcal{T}_{\mathrm{new}}), \quad (6)$$

to ensure backwards-compatibility, we add a second term to the loss that depends on the classifier of the old model:

$$w_{c\ \mathrm{new}}, w_{\phi\ \mathrm{new}} = \arg\min_w L_{\mathrm{BCT}}(w_c, w_\phi; \mathcal{T}_{\mathrm{new}}, \mathcal{T}_{\mathrm{BCT}}), \quad (7)$$

where

$$L_{\mathrm{BCT}}(w_c, w_\phi; \mathcal{T}_{\mathrm{new}}, \mathcal{T}_{\mathrm{BCT}}) = L(w_c, w_\phi; \mathcal{T}_{\mathrm{new}}) + \\ + \lambda L(w_{c\ \mathrm{old}}, w_\phi; \mathcal{T}_{\mathrm{BCT}}). \quad (8)$$

We call the second term "influence loss" since it biases the solution towards one that can use the old classifier. Note that $w_{c\ \mathrm{old}}$ in the influence loss will be fixed during training. Here, $\mathcal{T}_{\mathrm{BCT}}$ is a design parameter, referring to the set of images we apply the influence loss to. It can be either $\mathcal{T}_{\mathrm{old}}$ or $\mathcal{T}_{\mathrm{new}}$. The approach of using $\mathcal{T}_{\mathrm{new}}$ as $\mathcal{T}_{\mathrm{BCT}}$ will be introduced in Sect. 2.5. Note that the classifiers κ of the new and old models can be different. We call this method backward compatible training, and the result backward compatible representation or embedding, which we evaluate empirically in the next section.

2.5. Learning with Backward Compatible Training

In the proposed backward compatible training framework, there are several design choices to make.

Form of the classifier The classifiers κ of the new and old models κ_{new} and κ_{old} can be of the same form, for instance Softmax, angular SoftMax classifier [16], or cosine margin [34]. They can also be of different forms, which is common in the cases where better loss formulations are proposed and applied to training new embedding models.

Backward compatibility training dataset. The most straightforward choice for the dataset $\mathcal{T}_{\mathrm{BCT}}$, on which we

apply the influence loss, is $\mathcal{T}_{\mathrm{old}}$, which was used to train the old embedding ϕ_{old}. The intuition is that, since the old model ϕ_{old} is optimized together with its classifier κ_{old} on the original training set $\mathcal{T}_{\mathrm{old}}$, a new embedding model having a low influence loss will work with the old model's classifier and thus with the embedding vectors from ϕ_{old}. The second choice of $\mathcal{T}_{\mathrm{BCT}}$ is $\mathcal{T}_{\mathrm{new}}$; this means that we not only compute the influence loss on the old training data for ϕ_{old}, but also on the new training data. However, this choice poses a challenge in the computation of the loss value $L(w_c, w_\phi; \mathcal{T}_{\mathrm{new}})$ for the images in $\{x | x \in \mathcal{T}_{\mathrm{new}}, x \notin \mathcal{T}_{\mathrm{old}}\}$, due to the unknown classifier parameters for the classes. We propose two rules for computing the loss value for these images:

Synthesized classifier weights. For classes in $\mathcal{T}_{\mathrm{new}}$ which are not in the set of classes in $\mathcal{T}_{\mathrm{old}}$, we create their "synthesized" classifier weights by computing the average feature vector of ϕ_{old} on the images in each class. This approach is inspired by open-set recognition using the class vector ϕ_it as described in Sect. 2.1. We use averaging as the set function in this case. The synthesized classifier weights for the new classes are concatenated with the existing w_c to form the classifier parameters for the influence loss term.

Knowledge distillation. We penalize the KL-divergence of the classifier output probabilities between using ϕ_{new} and ϕ_{old} with existing classifier parameters w_c. This removes the requirement to add new classes in $\mathcal{T}_{\mathrm{new}}$ to the classifiers corresponding to ϕ_{old}.

Backward compatible training is not restricted to certain neural network architecture or loss function. It only requires that both the old and new embedding models be trained with classification-based losses, which is common in open-set recognition problems [34, 13]. It also does not need modification of the architecture nor of the parameters of the old model ϕ_{old}.

3. Experiments

We assess the effectiveness of the proposed backward compatibility training in face recognition. We start with several baselines, then test the hypothesis that BCT leads to backward compatible representation learning on two face recognition tasks: face verification and face search. Finally, we demonstrate the potential of BCT by applying it to the cases of multi-factor model changes and showing it is able to construct multiple compatible models.

3.1. Datasets and Face Recognition Metrics

We use the IMDB-Face dataset [32] for training face embedding models. The IMDB-Face dataset contains about 1.7M images of 59K celebrities. For the openset test, we use the widely adopted IJB-C face recognition benchmark dataset [18]. It has around 130k images from 3,531 identities. The images in IJB-C contain both still images and

video frames. We adopt the two standard testing protocols for face recognition: 1:1 verification and 1:N search (open set). For 1:1 verification, a pair of templates (a template contains one or more face images from the same person) are presented and the algorithm is required to decide whether they belong to the same person or not. The evaluation metrics for this protocol are true acceptance rate (TAR) at different false acceptance rates (FAR). We present the results of TAR at the FAR of 10^{-4}. For 1:N search, a set of templates is first indexed as the gallery set. Then each template in the query set is used to search against the indexed templates. The quality metrics for this protocols are true positive identification rates (TPIR) at different false positive identification rates (FPIR). We present the results of TPIR at 10^{-2} FPIR.

3.2. Implementation details

We use 8 NVIDIA Tesla V-100 GPUs in training the embedding models. The input size of embedding models is set to 112×112 pixels [36]. We use face mis-alignment and color distortion for data augmentation. Weight decay is set to 5×10^{-4} and standard stochastic gradient descent (SGD) is used to optimize the loss. The initial learning rate is set to 0.1 and decreased to 0.01, 0.001, and 0.0001 after 8, 12, and 14 epochs, respectively. The training stops after 16 epochs. The batchsize is set to 320. Unless stated otherwise, we use ResNet-101 [7] as the backbone, a linear transform after its global average pooling layer to emit 128-dimensional feature vectors , and Cosine Margin Loss [34] with margin=0.4 as the loss function in our experiments.

3.3. Measuring Backward-Compatibility

Based on the accuracy on the individual tests on the face recognition dataset, we can test whether a pair of models satisfies the empirical backward compatibility criterion. For a pair of models $(\phi_{\text{new}}, \phi_{\text{old}})$, on each evaluation protocol we test whether they satisfy Eq. (2). If so, we consider the new model **backward compatible** with the old model in the corresponding task. In testing using the IJB-C 1:N protocol [18], we use the new model ϕ_{new} to extract embeddings for the query set and the old model ϕ_{old} to compute embeddings for gallery set. For the IJB-C 1:1 verification protocol [18], we use ϕ_{new} to extract the embeddings for the first template in the pair and ϕ_{old} for the second.

To evaluate relative improvement brought by the backfill-free update, we define the **update gain** as

$$\mathcal{G}(\phi_{\text{new}}, \phi_{\text{old}}; \mathcal{Q}, \mathcal{D}) = \frac{M(\phi_{\text{new}}, \phi_{\text{old}}; \mathcal{Q}, \mathcal{D}) - M(\phi_{\text{old}}, \phi_{\text{old}}; \mathcal{Q}, \mathcal{D})}{M(\phi^*_{\text{new}}, \phi^*_{\text{new}}; \mathcal{Q}, \mathcal{D}) - M(\phi_{\text{old}}, \phi_{\text{old}}; \mathcal{Q}, \mathcal{D})}. \tag{9}$$

Here, $M(\phi^*_{\text{new}}, \phi^*_{\text{new}}; \mathcal{Q}, \mathcal{D})$ stands for the best accuracy level we can achieve from any variants of the new model by backfilling. It indicates the proportional gain we can obtain the performing backfill-free update compared with the

update which performs the backfill regardless of the cost and interruption of services. Note that the update gain is only valid when Eq. (2) is satisfied.

3.4. Baseline comparisons

The first hypothesis to be tested is whether BCT is necessary at all: is it possible to achieve backward compatibility with a more straightforward approach? In this section, we experiment with several baseline approaches and validate the necessity of BCT.

Independently trained ϕ_{new} and ϕ_{old} The first sanity-check is to directly compare the embedding of two versions of models trained independently. A similar experiment is done in [14] for multiple close-set classification models trained on the same dataset. Here we present two models. The ϕ_{old} is trained with the randomly sampled 50% IDs subset of the IMDBFace dataset [32]. The new model is trained on the full IMDBFace dataset [32]. This emulates the case where a new embedding model becomes available when the sizes of embedding training dataset grows. We name the new model ϕ^*_{new} according to the experiment in Sect. 3.5, showing that it currently achieves the best accuracy among all new models with the same setting. We directly test the compatibility of this pair of models $(\phi^*_{\text{new}}, \phi_{\text{old}})$ following the procedure described in Sect. 3.3. The results are illustrated in Tab. 1. In the backward test for both protocols, we observed almost 0% accuracy. Unsurprisingly, independently trained ϕ_{new} and ϕ_{old} does not naturally satisfy our compatibility criterion.

Does the naive baseline with ℓ^2-distance work? In Sect. 2.3 we described the naive approach of adding the ℓ^2-distance between the new and old embeddings as a regularizer when training the new model. We train a new model using the same old model above and train the new model with the loss function (3) on the whole IMDBFace dataset [32]. We name this model $\phi_{\text{new}-\ell^2}$ to reflect that it is ℓ^2-distance regularized towards the old model. The same backward compatibility test is conducted with this pair of models $(\phi_{\text{new}-\ell^2}, \phi_{\text{old}})$ on the same two protocols described in the previous baseline. The results are shown in Tab. 1. We can observe that this approach only leads to slightly above 0% backward test accuracy, which means the new model $\phi_{\text{new}-\ell^2}$ is far from satisfying the compatibility criterion. One possible reason is that imposing an ℓ_2 distance penalty creates a bias that is too local and restrictive to allow the new model to satisfy the compatibility constraints.

3.5. Learning with BCT

We now experiment with the proposed BCT framework for backward compatible representation learning, starting from its basic form described in Sect. 2.4. We use the same old model in the previous section. For the new model, we train it with the objective function described in Eq. (8).

New Model	Old Model	Data	Additional Loss
ϕ_{old}	-	50%	-
ϕ_{new}^*	-	100%	-
$\phi_{\text{new}-\ell^2}$	ϕ_{old}	100%	ℓ^2 distance ϕ_{old}
$\phi_{\text{new}-\text{LwF}}$	ϕ_{old}	50%	Learning w/o Forgetting
$\phi_{\text{new}-\beta}$	ϕ_{old}	100%	Influence loss on $\mathcal{T}_{\text{old}}$
$\phi_{\text{new}-\beta-kd}$	ϕ_{old}	100%	Influence loss on $\mathcal{T}_{\text{new}}$
$\phi_{\text{new}-\beta-sys}$	ϕ_{old}	100%	Influence loss on $\mathcal{T}_{\text{new}}$

(a) Training setting for different backward-compatible (new) models. 'ϕ_{old}': the compatible target (old) model for all new models. '$\phi_{new-\ell^2}$': the new model regularized with ℓ^2 distance to ϕ_{old} . '$\phi_{new-\text{LwF}}$': the new model that adopts learning w/o forgetting [15] in training. '$\phi_{new-\beta}$': the new model trained with the proposed BCT. '$\phi_{new-\beta-kd}$': the new model trained with the proposed BCT and knowledge distillation for new classes in the new embedding training dataset. '$\phi_{new-\beta-sys}$': the new model trained with proposed BCT and the synthesised classifiers for new classes in the new embedding training dataset.

Comparison Pair	Veri. Acc.	Backward Compatible?	Update Gain (%)	Absolute Gain
$(\phi_{\text{old}},\phi_{\text{old}})$ (Lower Bound)	77.86	-	-	-
$(\phi_{\text{new}}^*,\phi_{\text{old}})$	0.0	$\times$	-	-
$(\phi_{\text{new}-\ell^2},\phi_{\text{old}})$	3.10	$\times$	-	-
$(\phi_{\text{new}-\text{LwF}},\phi_{\text{old}})$	77.26	$\times$	-	-
$(\phi_{\text{new}-\beta},\phi_{\text{old}})$ (Ours)	80.25	$\checkmark$	26.26	2.39
$(\phi_{\text{new}-\beta-kd},\phi_{\text{old}})$ (Ours)	80.34	$\checkmark$	27.25	2.48
$(\phi_{\text{new}-\beta-sys},\phi_{\text{old}})$ (Ours)	80.59	$\checkmark$	30.00	2.73
$(\phi_{\text{new}}^*,\phi_{\text{new}}^*)$ (Upper Bound)	86.96	-	-	9.1

(b) Experiments on the IJB-C 1:1 verification task. The verification accuracy evaluation metric is TAR (%)@FAR=10^{-4}.

Comparison Pair	Search Acc.	Backward Compatible?	Update Gain (%)	Absolute Gain
$(\phi_{\text{old}},\phi_{\text{old}})$ (Lower Bound)	59.34	-	-	-
$(\phi_{\text{new}}^*,\phi_{\text{old}})$	0.0	$\times$	-	-
$(\phi_{\text{new}-\ell^2},\phi_{\text{old}})$	0.50	$\times$	-	-
$(\phi_{\text{new}-\text{LwF}},\phi_{\text{old}})$	59.27	$\times$	-	-
$(\phi_{\text{new}-\beta},\phi_{\text{old}})$ (Ours)	67.23	$\checkmark$	44.98	7.89
$(\phi_{\text{new}-\beta-kd},\phi_{\text{old}})$ (Ours)	69.02	$\checkmark$	55.11	9.68
$(\phi_{\text{new}-\beta-sys},\phi_{\text{old}})$ (Ours)	70.70	$\checkmark$	64.77	11.36
$(\phi_{\text{new}}^*,\phi_{\text{new}}^*)$ (Upper Bound)	76.88	-	-	17.54

(c) Experiments on the IJB-C 1:N search task. The search accuracy evaluation metric is TNIR(%)@FPIR=10^{-2}.

Table 1: Illustration of simple baselines and our proposed approach in backward compatibility test. In face search we use the first model of each comparison pair]for the query set and the second for the gallery set, and in face verification for the first and second template, respectively. The details of model training setting are illustrated in Tab. 1a. We report the relative update gain defined in Eq. 9.

This model is called $\phi_{\text{new}-\beta}$. As illustrated in Tab. 1b and Tab. 1c, the model pair, $(\phi_{\text{new}-\beta},\phi_{\text{old}})$, satisfies the backward compatibility criterion in Eq. (2). Additionally, we

Comparison Pair	IJB-C 1:1 Verifi. TAR (%)@FAR=10^{-4}	IJB-C 1:N Retri. TNIR(%)@FPIR=10^{-2}
$(\phi_{\text{old}},\phi_{\text{old}})$	77.86	59.34
$(\phi_{\text{new}-\beta},\phi_{\text{new}-\beta})$	85.36	73.86
$(\phi_{\text{new}-\beta-kd},\phi_{\text{new}-\beta-kd})$	84.95	73.56
$(\phi_{\text{new}-\beta-sys},\phi_{\text{new}-\beta-sys})$	85.58	74.40
$(\phi_{\text{new}}^*,\phi_{\text{new}}^*)$	86.96	76.88

Table 2: Backward compatibility test for old model ϕ_{old}, BCT trained model $\phi_{\text{new}-\beta}$, $\phi_{\text{new}-\beta-kd}$, $\phi_{\text{new}-\beta-sys}$ and paragon/upper bound model ϕ_{new}^*. The results shows that BCT does not lead to significant accuracy drop compared with paragon/upper bound model.

observe update gains of 26.26% and 44.98% on the 1:1 verification and 1:N search protocols respectively.

We also evaluate a baseline approach adapted from [15]. The model trained with this approach is denoted as $\phi_{\text{new}-\text{LwF}}$. It uses fixed ϕ_{old} and its classifier $w_{c\,\text{old}}$ to output soft labels of newly added samples $x \in \mathcal{T}_{\text{new}} \setminus \mathcal{T}_{\text{old}}$ as pseudo labels for training $\phi_{\text{new}-\text{LwF}}$. From Tab.1b and 1c we can see that model pair ($\phi_{\text{new}-\text{LwF}}, \phi_{\text{old}}$) does not satisfy the empirical backward compatibility criterion. Showing that directly adapting methods from the continual learning task does not work out of the box. However, it is able to improves to some extend the backward comparison accuracy, suggesting that the knowledge distillation used in continual learning could be useful in BCT. We further investigate its application in the following experiments.

BCT with newly added training data. In Sect. 2.5 we described two instantiations of BCT which can work with the new classes in the growing embedding training set. The first is using the synthesised classifier, we name the new model trained with this form of BCT as $\phi_{\text{new}-\beta-sys}$. The second applies the idea of knowledge distillation to bypass obtaining classifier parameters for the new classes in the embedding training set. We name the new model trained with this form of BCT as $\phi_{\text{new}-\beta-kd}$. The backward compatibility test results are summarized in Tab. 1. We can see that both new models can achieve backward compatibility. By fully utilizing the additional training data, they also lead to higher update gain (30.00% for $\phi_{\text{new}-\beta-sys}$ and 27.25% for $\phi_{\text{new}-\beta-kd}$) compared with the basic form of BCT (26.26% for $\phi_{\text{new}-\beta}$).

Does BCT hurt the accuracy of new models? One natural question is whether the influence loss is detrimental to the new model's recognition performance. We assess this by performing standard face recognition experiments on the 1:1 and 1:N protocols by extracting embedding only using the new models. This process can be considered as performing the backfill, or the paragon setting as described in Sect. 2.3. The results are summarized in Tab. 2. We can see that training without BCT still yields the best accuracy in this setting. So, we name the model trained without BCT as

New Model	Old Model	Training Data Usage	Feat. Dim.	Model Arc.	Classifier	Additional Loss
ϕ_{old}	-	50%	128	ResNet-101	Cosine Margin	-
$\phi_{\mathrm{new}-\beta}^{\mathrm{R152}}$	ϕ_{old}	100%	128	ResNet-152	Cosine Margin	Influence loss on $\mathcal{T}_{\mathrm{old}}$
$\phi_{\mathrm{new}-\beta}^{\mathrm{R152+256D}}$	ϕ_{old}	100%	256	ResNet-152	Cosine Margin	Influence loss on $\mathcal{T}_{\mathrm{old}}$
$\phi_{\mathrm{new}-\beta}^{\mathrm{ReLU}}$	ϕ_{old}	100%	128	ResNet-152	Cosine Margin	Influence loss on $\mathcal{T}_{\mathrm{old}}$
$\phi_{\mathrm{old}}^{\mathrm{NS}}$	-	100%	128	ResNet-101	Norm-Softmax	-
$\phi_{\mathrm{new}-\beta}^{\mathrm{Cos-NS}}$	$\phi_{\mathrm{old}}^{\mathrm{NS}}$	100%	128	ResNet-101	Cosine Margin	Influence loss on $\mathcal{T}_{\mathrm{old}}$
$\phi_{\mathrm{old}}^{\mathrm{S}}$	-	100%	128	ResNet-101	SoftMax	-
$\phi_{\mathrm{new}-\beta}^{\mathrm{Cos-S}}$	$\phi_{\mathrm{old}}^{\mathrm{S}}$	100%	128	ResNet-101	Cosine Margin	Influence loss on $\mathcal{T}_{\mathrm{old}}$

(a) 'ϕ_{old}': the compatible target model for New model. '$\phi_{\mathrm{new}-\beta}^{\mathrm{R152}}$': using ResNet-152 as backbone with the proposed BCT. '$\phi_{\mathrm{new}-\beta}^{\mathrm{R152+256D}}$': using ResNet-152 as backbone and feature dimension of 256 with the proposed BCT. '$\phi_{\mathrm{new}-\beta}^{\mathrm{ReLU}}$': adding a ReLU module after the embedding output of the new model when training with BCT. '$\phi_{\mathrm{old}}^{\mathrm{NS}}$': the old model with normalized SoftMax classifier [33]. '$\phi_{\mathrm{new}-\beta}^{\mathrm{Cos-NS}}$': the new model with cosine margin classifier [34] and trained by BCT with $\phi_{\mathrm{old}}^{\mathrm{NS}}$ as the old model. '$\phi_{\mathrm{old}}^{\mathrm{S}}$': using standard softmax loss as training loss. '$\phi_{\mathrm{new}-\beta}^{\mathrm{Cos-S}}$': the new model with cosine margin classifier class [34] and BCT with $\phi_{\mathrm{old}}^{\mathrm{S}}$ as the old model.

Comparison Pair	Veri. Acc.	Backward Compatible?	Update Gain (%)	Absolute Gain
$(\phi_{\mathrm{old}}, \phi_{\mathrm{old}})$ (Lower B.)	77.86	-	-	-
$(\phi_{\mathrm{new}-\beta}^{\mathrm{R152}}, \phi_{\mathrm{old}})$	80.54	√	29.45	2.68
$(\phi_{\mathrm{new}-\beta}^{\mathrm{R152+256D}}, \phi_{\mathrm{old}})$	80.92	√	33.63	3.06
$(\phi_{\mathrm{new}-\beta}^{\mathrm{ReLU}}, \phi_{\mathrm{old}})$	34.70	×	-	-
$(\phi_{\mathrm{old}}^{\mathrm{NS}}, \phi_{\mathrm{old}}^{\mathrm{NS}})$ (Lower B.)	80.10	-	-	-
$(\phi_{\mathrm{new}-\beta}^{\mathrm{Cos-NS}}, \phi_{\mathrm{old}}^{\mathrm{NS}})$	81.81	√	24.93	1.71
$(\phi_{\mathrm{old}}^{\mathrm{S}}, \phi_{\mathrm{old}}^{\mathrm{S}})$ (Lower B.)	73.27	-	-	-
$(\phi_{\mathrm{new}-\beta}^{\mathrm{Cos-S}}, \phi_{\mathrm{old}}^{\mathrm{S}})$	67.11	×	-	-
$(\phi_{\mathrm{new}}^{*}, \phi_{\mathrm{new}}^{*})$ (Upper B.)	86.96	-	-	9.1

(b) Experiments on the IJB-C 1:1 verification task. The verification accuracy evaluation metric is TAR (%)@FAR=10^{-4}.

Comparison Pair	Search Acc.	Backward Compatible?	Update Gain (%)	Absolute Gain
$(\phi_{\mathrm{old}}, \phi_{\mathrm{old}})$ (Lower B.)	59.34	-	-	-
$(\phi_{\mathrm{new}-\beta}^{\mathrm{R152}}, \phi_{\mathrm{old}})$	68.71	√	53.42	9.37
$(\phi_{\mathrm{new}-\beta}^{\mathrm{R152+256D}}, \phi_{\mathrm{old}})$	69.45	√	57.63	10.11
$(\phi_{\mathrm{new}-\beta}^{\mathrm{ReLU}}, \phi_{\mathrm{old}})$	17.73	×	-	-
$(\phi_{\mathrm{old}}^{\mathrm{NS}}, \phi_{\mathrm{old}}^{\mathrm{NS}})$ (Lower B.)	64.32	-	-	-
$(\phi_{\mathrm{new}-\beta}^{\mathrm{Cos-NS}}, \phi_{\mathrm{old}}^{\mathrm{NS}})$	71.16	√	54.46	6.84
$(\phi_{\mathrm{old}}^{\mathrm{S}}, \phi_{\mathrm{old}}^{\mathrm{S}})$ (Lower B.)	54.16	-	-	-
$(\phi_{\mathrm{new}-\beta}^{\mathrm{Cos-S}}, \phi_{\mathrm{old}}^{\mathrm{S}})$	45.46	×	-	-
$(\phi_{\mathrm{new}}^{*}, \phi_{\mathrm{new}}^{*})$ (Upper B.)	76.88	-	-	17.54

(c) Experiments on the IJB-C 1:N retrieval task. The search accuracy evaluation metric is TNIR(%)@FPIR=10^{-2}.

Table 3: Robustness analysis of BCT against different training factors. We train new models with BCT while changing the network structure, feature dimensions, data amount and supervision loss, respectively. The training details for different models are listed in Tab. 3a.

ϕ_{new}^{*}, to indicate that it is the paragon that achieves best accuracy among all variants of new models. Note that models trained with the basic form of BCT, $\phi_{\mathrm{new}-\beta}$, only leads to less than 3% drop of accuracy in both tasks. The new models $\phi_{\mathrm{new}-\beta-sys}$ and $\phi_{\mathrm{new}-\beta-kd}$ further reduce the gap.

3.6. Extensions of BCT

In the following experiments we explore whether BCT can be applied to different types of model training and achieve *multi-model* compatibility.

Other changes in training ϕ_{new}. Besides increasing the size of the embedding training set, the new model ϕ_{new} could have a new model architecture (*e.g.*, depth), a different loss function for supervision, or different embedding dimensions. We experiment the effect of these factors on BCT. For network architectures, we test a new model using ResNet-152 [7] instead of ResNet-101 [7] in previous experiments, denoted as $\phi_{\mathrm{new}-\beta}^{\mathrm{R152}}$. In terms of loss types, we test using Norm-Softmax Loss [33] for the old model, $\phi_{\mathrm{old}}^{\mathrm{NS}}$, and Cosine Margin Loss [34] for the new one, $\phi_{\mathrm{new}-\beta}^{\mathrm{Cos}}$. In terms of embedding dimension, we test increasing the dimensions from 128 to 256 in the new model. Note that

when the new model's feature dimension is changed, we will not be able to directly feed it to $w_{c\ \mathrm{old}}$. Here, we simply take the first 128 elements of the new model's feature to feed into $w_{c\ \mathrm{old}}$ during backward-compatible training and testing. We tried another approach of adding a linear transformer in training to match the feature dimension but without success.

We also test the cases of changing several factors together, denoted as $\phi_{\mathrm{new}-\beta}^{\mathrm{R152+256D}}$. The results are shown in Tab. 3. BCT can make most of new models backward compatible, even when several factors change simultaneously. This shows that BCT can be used as a general framework for achieving backward compatible representation learning.

There are two failure cases where backward compatibility is not achieved in the pairs: 1)($\phi_{\mathrm{new}-\beta}^{\mathrm{Cos-S}}$, $\phi_{\mathrm{old}}^{\mathrm{S}}$), which uses Softmax loss [12] for the old model and Cosine Margin Loss [34] for new model; this is possibly due to the drastic change of form of the loss functions. and 2) ($\phi_{\mathrm{new}-\beta}^{\mathrm{ReLU}}$, ϕ_{old}), which adds ReLU activation on the embedding of the new model. The latter is possibly due to the distributional shift introduced by the ReLU activation in the new model, which makes it difficult for the new model with non-negative

New Model	Old Model	Data	Additional Loss
ϕ_1	-	25%	-
ϕ_2	ϕ_1	50%	Influence loss on $\mathcal{T}_1$
ϕ_3	ϕ_2	100%	Influence loss on $\mathcal{T}_2$

(a) 'ϕ_1': the first version model trained with 25% of training data. 'ϕ_2': the second version model trained on 50% of training data with BCT towards ϕ_1. 'ϕ_3': the third version model trained on all training data with BCT towards ϕ_2.

Comparison Pair	Veri. Acc.	Backward Compatible?	Update Gain (%)	Absolute Gain
(ϕ_1, ϕ_1)	41.45	-	-	-
(ϕ_2, ϕ_1)	56.34	√	40.90	14.89
(ϕ_3, ϕ_1)	53.98	√	34.41	12.53
(ϕ_2, ϕ_2)	75.96	-	-	-
(ϕ_3, ϕ_2)	80.40	√	40.36	4.44
(ϕ_2^*, ϕ_2^*)	77.86	-	-	36.41
(ϕ_3^*, ϕ_3^*)	86.96	-	-	9.1

(b) Experiments on the IJB-C 1:1 verification task for chain update ability test.

Comparison Pair	Search Acc.	Backward Compatible?	Update Gain (%)	Absolute Gain
(ϕ_1, ϕ_1)	22.57	-	-	-
(ϕ_2, ϕ_1)	39.00	√	44.68	16.43
(ϕ_3, ϕ_1)	36.10	√	36.80	13.53
(ϕ_2, ϕ_2)	56.07	-	-	-
(ϕ_3, ϕ_2)	66.09	√	48.15	10.02
(ϕ_2^*, ϕ_2^*)	59.34	-	-	36.77
(ϕ_3^*, ϕ_3^*)	76.88	-	-	17.54

(c) Experiments on the IJB-C 1:N verification task for chain update ability test.

Table 4: Experiments on multi-model compatibility between three models trained with growing amount of data. In Tab. 4a, we illustrate the training details of all the models we trained. In Tab. 4b and Tab. 4c, we verify the compatibility of the trained models. Please be noted that the update gains are calculated by comparing (ϕ_1, ϕ_2), (ϕ_1, ϕ_3), and (ϕ_2, ϕ_3).

Comparison Pair	Mean AP (%)	Backward Compatible?	Update Gain (%)	Absolute Gain
(ϕ_{old}, ϕ_{old})	42.9	-	-	-
$(\phi_{old}, \phi_{new}^*)$	26.7	×	-	-
$(\phi_{old}, \phi_{new-\beta})$	45.0	√	12.0	2.1
$(\phi_{new-\beta}, \phi_{new-\beta})$	60.1	-	-	-
$(\phi_{new}^*, \phi_{new}^*)$	60.3	-	-	-

Table 5: Backward compatibility tests on Market-1501 person re-identification dataset [45]. Models ϕ_{old}, ϕ_{new}^* are trained following [39] with 50% and 100% of training data, without BCT. $\phi_{new-\beta}$ is trained with 100% of the training data and with BCT. Person search mean average precision (mean AP) is the accuracy metric.

embedding vector elements to be compatible with the old model. This suggests that additional work is needed to expand the set of models that BCT can support.
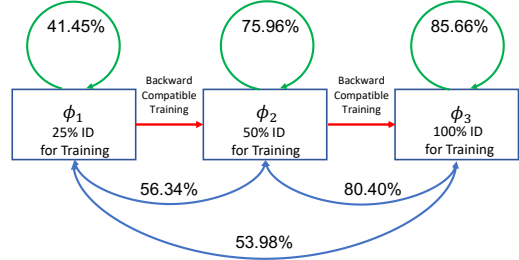

Figure 2: Visualization of the multi-model compatibility experiment results. We trained three models ϕ_1, ϕ_2, and ϕ_3. The backward compatible training is enforced as shown above. On the blue arrows we mark the backward compatibility test accuracy on the IJB-C 1:1 face verification benchmark, the green arrows mark the accuracy of the paragon settings.

Towards multi-model and sequential compatibility. Here we investigate a simple case of three model versions. The first version ϕ_1 is trained with $\mathcal{T}_1$, which is a randomly sampled 25% subset of the IMDBFace dataset [32]. The second version ϕ_2 is trained with $\mathcal{T}_2$, a 50% subset. And the third version ϕ_3 is trained with $\mathcal{T}_3$, which is the full IMDB-Face dataset [32]. We train ϕ_2 using BCT with ϕ_1 and train ϕ_3 using BCT with ϕ_2. Thus, in this process ϕ_3 has no direct influence from ϕ_1. The backward compatibility test results are shown in Tab. 4 and Fig. 2. We observe that, by training with BCT, the last model ϕ_3 is transitively compatible with ϕ_1 even though ϕ_1 is not directly involved in training ϕ_3. It shows that transitive compatibility between multiple models is indeed achievable through BCT, which could enable sequential update of the embedding models.

BCT in other open-set recognition tasks. We validate the BCT method on the person re-identification task using the Market-1501 [45] benchmark. We train an old embedding model following [39] with 50% of training data and two new embedding models with 100% of the new training data. Search mean average precision (mean AP) is used as the accuracy metric. Table 5 shows the results of backward compatibility test. We observe that the $\phi_{new-\beta}$ trained with BCT achieves backward compatibility without sacrificing its own search accuracy. This suggests that BCT can be a general approach for open-set recognition problems.

4. Discussion

We have presented a method for achieving backward-compatible representation learning, illustrated specific instances, and compared them with both baselines and paragons. Our approach has several limitations. The first is the accuracy gap of the new models trained with BCT relative to the new model oblivious of previous constraints. Though the gap is reduced by slightly more sophisticated forms of BCT, there is still work wo to be done in characterizing and achieving the attainable accuracy limits.

References

[1] Ejaz Ahmed, Michael Jones, and Tim K Marks. An improved deep learning architecture for person re-identification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3908–3916, 2015. 2

[2] Gagan Bansal, Besmira Nushi, Ece Kamar, Dan Weld, Walter Lasecki, and Eric Horvitz. A case for backward compatibility for human-ai teams. *arXiv preprint arXiv:1906.01148*, 2019. 2

[3] Abhijit Bendale and Terrance E Boult. Towards open set deep networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1563–1572, 2016. 2

[4] Konstantinos Bousmalis, George Trigeorgis, Nathan Silberman, Dilip Krishnan, and Dumitru Erhan. Domain separation networks. In *Advances in neural information processing systems*, pages 343–351, 2016. 2

[5] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4690–4699, 2019. 2

[6] Albert Gordo, Jon Almazán, Jerome Revaud, and Diane Larlus. Deep image retrieval: Learning global representations for image search. In *European conference on computer vision*, pages 241–257. Springer, 2016. 2

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. 1, 5, 7

[8] Alexander Hermans, Lucas Beyer, and Bastian Leibe. In defense of the triplet loss for person re-identification. *arXiv preprint arXiv:1703.07737*, 2017. 2

[9] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. 2

[10] Weixiang Hong, Zhenzhen Wang, Ming Yang, and Junsong Yuan. Conditional generative adversarial network for structured domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1335–1344, 2018. 2

[11] Saihui Hou, Xinyu Pan, Chen Change Loy, Zilei Wang, and Dahua Lin. Learning a unified classifier incrementally via rebalancing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 831–839, 2019. 2

[12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012. 2, 4, 7

[13] Wei Li, Rui Zhao, Tong Xiao, and Xiaogang Wang. Deepreid: Deep filter pairing neural network for person re-identification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 152–159, 2014. 2, 4

[14] Yixuan Li, Jason Yosinski, Jeff Clune, Hod Lipson, and John E Hopcroft. Convergent learning: Do different neural networks learn the same representations? In *Iclr*, 2016. 2, 5

[15] Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2935–2947, 2017. 2, 6, 13

[16] Weiyang Liu, Yandong Wen, Zhiding Yu, Ming Li, Bhiksha Raj, and Le Song. Sphereface: Deep hypersphere embedding for face recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 212–220, 2017. 4

[17] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael I Jordan. Learning transferable features with deep adaptation networks. *arXiv preprint arXiv:1502.02791*, 2015. 2

[18] 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 *2018 International Conference on Biometrics (ICB)*, pages 158–165. IEEE, 2018. 4, 5, 11, 13

[19] Pau Panareda Busto and Juergen Gall. Open set domain adaptation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 754–763, 2017. 2

[20] Shibin Parameswaran and Kilian Q Weinberger. Large margin multi-task metric learning. In *Advances in neural information processing systems*, pages 1867–1875, 2010. 2

[21] German I Parisi, Ronald Kemker, Jose L Part, Christopher Kanan, and Stefan Wermter. Continual lifelong learning with neural networks: A review. *Neural Networks*, 2019. 2

[22] Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, et al. Deep face recognition. In *bmvc*, volume 1, page 6, 2015. 2

[23] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 2001–2010, 2017. 2

[24] Walter J Scheirer, Anderson de Rezende Rocha, Archana Sapkota, and Terrance E Boult. Toward open set recognition. *IEEE transactions on pattern analysis and machine intelligence*, 35(7):1757–1772, 2012. 2

[25] Walter J Scheirer, Lalit P Jain, and Terrance E Boult. Probability models for open set recognition. *IEEE transactions on pattern analysis and machine intelligence*, 36(11):2317–2324, 2014. 2

[26] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823, 2015. 2

[27] Milan Sulc and Jiri Matas. Improving cnn classifiers by estimating test-time priors. In *The IEEE International Conference on Computer Vision (ICCV) Workshops*, Oct 2019. 2

[28] Yi Sun, Yuheng Chen, Xiaogang Wang, and Xiaoou Tang. Deep learning face representation by joint identification-verification. In *Advances in neural information processing systems*, pages 1988–1996, 2014. 2, 4

[29] Yi Sun, Xiaogang Wang, and Xiaoou Tang. Deep learning face representation from predicting 10,000 classes. In *The*

IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2014. 2

[30] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015. 1

[31] Eric Tzeng, Judy Hoffman, Ning Zhang, Kate Saenko, and Trevor Darrell. Deep domain confusion: Maximizing for domain invariance. *arXiv preprint arXiv:1412.3474*, 2014. 2

[32] Fei Wang, Liren Chen, Cheng Li, Shiyao Huang, Yanjie Chen, Chen Qian, and Chen Change Loy. The devil of face recognition is in the noise. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 765–780, 2018. 2, 4, 5, 8

[33] Feng Wang, Xiang Xiang, Jian Cheng, and Alan Loddon Yuille. Normface: l 2 hypersphere embedding for face verification. In *Proceedings of the 25th ACM international conference on Multimedia*, pages 1041–1049. ACM, 2017. 2, 4, 7, 12

[34] Hao Wang, Yitong Wang, Zheng Zhou, Xing Ji, Dihong Gong, Jingchao Zhou, Zhifeng Li, and Wei Liu. Cosface: Large margin cosine loss for deep face recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5265–5274, 2018. 2, 4, 5, 7, 11, 12, 13

[35] Mei Wang and Weihong Deng. Deep visual domain adaptation: A survey. *Neurocomputing*, 312:135–153, 2018. 2

[36] Yandong Wen, Kaipeng Zhang, Zhifeng Li, and Yu Qiao. A discriminative feature learning approach for deep face recognition. In *European conference on computer vision*, pages 499–515. Springer, 2016. 2, 5

[37] Zhirong Wu, Yuanjun Xiong, Stella X. Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 2

[38] Tong Xiao, Hongsheng Li, Wanli Ouyang, and Xiaogang Wang. Learning deep feature representations with domain guided dropout for person re-identification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1249–1258, 2016. 2

[39] Tong Xiao, Shuang Li, Bochao Wang, Liang Lin, and Xiaogang Wang. Joint detection and identification feature learning for person search. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3415–3424, 2017. 8

[40] Hongliang Yan, Yukang Ding, Peihua Li, Qilong Wang, Yong Xu, and Wangmeng Zuo. Mind the class weight bias: Weighted maximum mean discrepancy for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2272–2281, 2017. 2

[41] Jiaolong Yang, Peiran Ren, Dongqing Zhang, Dong Chen, Fang Wen, Hongdong Li, and Gang Hua. Neural aggregation network for video face recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4362–4371, 2017. 3

[42] Jiahui Yu and Thomas Huang. Network slimming by slimmable networks: Towards one-shot architecture search for channel numbers. *arXiv preprint arXiv:1903.11728*, 2019. 2

[43] Jiahui Yu and Thomas Huang. Universally slimmable networks and improved training techniques. *arXiv preprint arXiv:1903.05134*, 2019. 2

[44] Jiahui Yu, Linjie Yang, Ning Xu, Jianchao Yang, and Thomas Huang. Slimmable neural networks. *arXiv preprint arXiv:1812.08928*, 2018. 2

[45] Liang Zheng, Liyue Shen, Lu Tian, Shengjin Wang, Jingdong Wang, and Qi Tian. Scalable person re-identification: A benchmark. In *Proceedings of the IEEE international conference on computer vision*, pages 1116–1124, 2015. 2, 8

Appendices

A. Implementation Details

In Section 2.5, we describe how to use the influence loss on newly added training data for BCT to 1) compute synthesized classifier weights with the old model, and 2) use knowledge distillation. When we have a newly added training example whose class is not in the old embedding training set, we feed the new image into the old model ϕ_{old} and old classifier $w_{\text{c old}}$ to obtain the classifier responses. Then, we can provide supervision signal to the new model ϕ_{new} for this image by feeding the new model's embedding to the old classifier $w_{\text{c old}}$ and compute the knowledge distillation loss (cross-entropy with temperature-modulated SoftMax) between the two response vectors as the influence loss. Because we are using the cosine margin loss [34] in the experiments which also has a temprature parameter, we set the temperature parameter in knowledge distillation to the same as the one in the cosine margin loss, which is 32.

B. Partial Backfilling

Partial backfilling happens when only a part of the gallery classes have been processed by the new model ϕ_{new}. We test whether queries from the backward compatible new model $\phi_{new-\beta}$ can work with partially backfilled gallery sets. In Fig. 3, we illustrate the search accuracy on gallery sets of different backfill ratios. As higher percentages of the gallery set are backfilled, search accuracy grow proportionally towards the accuracy of the fully backfilled case. This suggests that one can upgrade to a new model, immediately benefiting from the improved accuracy, and optionally backfill the old gallery gradually in the background until paragon performance is achieved.

C. Detailed Benchmark Results

Due to space limitations, in the main paper we only report one specific operating point for each metric in the evaluation, *e.g.*, $\text{TAR@FAR} = 10^{-4}$ for face verification and $\text{TPIR@FPIR} = 10^{-2}$ for face identification. Here we report results at additional operating points on the IJB-C [18] benchmark. In Table 8, we show the performance of different compared baselines and our proposed method. In Table 7 and Table 9, we show the performance of extensions of our proposed backward-compatible training process. In Table 7, we illustrate extensions of BCT to different model depths, feature dimensions and supervision losses. In Table 9, we show extensions to multi-model compatibility towards sequential updating.
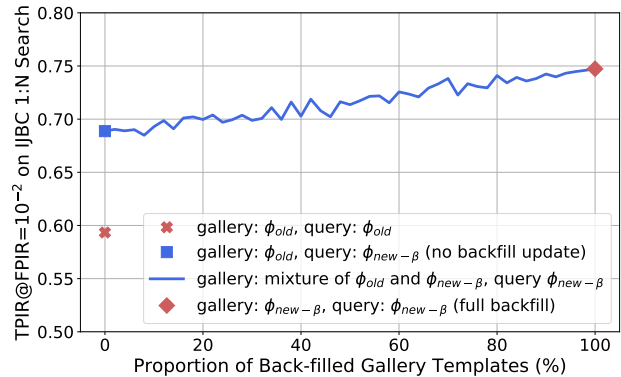
Figure 3: The curve for face search accuracy vs. backfill proportions. We gradually backfill the gallery set from 0% classes (same as the backward compatibility test or no-backfill update) to 100% (fully backfill or the paragon setting). Face search accuracy is measured between every 2% of partial backfill. The red cross shows the face search accuracy of the old model. The red diamonds marks the face search accuracy of the new embedding model.

	Continual Learning	Domain Adaptation	BCRL
Access to all old model parameters	Yes	Yes	Not required
Access to old training data	Not required	Yes	Yes
Re-processing of test data?	Yes	Yes	Not required
Consistent output	Yes	Not required	Yes
Compatible representation	Not required	Not required	Yes

Table 6: The differences between backward compatible representation learning (BCRL) , continual Learning, and domain adaptation. In this table, we list several features of the task each problem deals with to illustrate the difference.

New Model	Old Model	Training Data Usage	Feat. Dim.	Model Arc.	Classifier	Additional Loss
ϕ_{old}	-	50%	128	ResNet-101	Cosine Margin	-
$\phi_{\text{new}-\beta}^{\text{R152}}$	ϕ_{old}	100%	128	ResNet-152	Cosine Margin	Influence loss on $\mathcal{T}_{\text{old}}$
$\phi_{\text{new}-\beta}^{\text{R152}+256\text{D}}$	ϕ_{old}	100%	256	ResNet-152	Cosine Margin	Influence loss on $\mathcal{T}_{\text{old}}$
$\phi_{\text{new}-\beta}^{\text{ReLU}}$	ϕ_{old}	100%	128	ResNet-152	Cosine Margin	Influence loss on $\mathcal{T}_{\text{old}}$
$\phi_{\text{old}}^{\text{NS}}$	-	100%	128	ResNet-101	Norm-Softmax	-
$\phi_{\text{new}-\beta}^{\text{Cos}-\text{NS}}$	$\phi_{\text{old}}^{\text{NS}}$	100%	128	ResNet-101	Cosine Margin	Influence loss on $\mathcal{T}_{\text{old}}$
$\phi_{\text{old}}^{\text{S}}$	-	100%	128	ResNet-101	SoftMax	-
$\phi_{\text{new}-\beta}^{\text{Cos}-\text{S}}$	$\phi_{\text{old}}^{\text{S}}$	100%	128	ResNet-101	Cosine Margin	Influence loss on $\mathcal{T}_{\text{old}}$
$\phi_1^{25\%}$	-	25%	128	ResNet-101	Cosine Margin	-
$\phi_{\text{new}-\beta}^{25\%}$	$\phi_1^{25\%}$	100%	128	ResNet-101	Cosine Margin	Influence loss on $\mathcal{T}_{\text{old }25\%}$
$\phi_1^{90\%}$	-	90%	128	ResNet-101	Cosine Margin	-
$\phi_{\text{new}-\beta}^{90\%}$	$\phi_1^{90\%}$	100%	128	ResNet-101	Cosine Margin	Influence loss on $\mathcal{T}_{\text{old }90\%}$

(a) 'Old Model': the compatible target model for New model. '$\phi_{\text{new}-\beta}^{\text{R152}}$': using ResNet-152 as backbone with the proposed BCT. '$\phi_{\text{new}-\beta}^{\text{R152}+256\text{D}}$': using ResNet-152 as backbone and feature dimension of 256 with the proposed BCT. '$\phi_{\text{new}-\beta}^{\text{ReLU}}$': adding a ReLU module after the embedding output of the new model when training with BCT. '$\phi_{\text{old}}^{\text{NS}}$': the old model with normalized SoftMax classifier [33]. '$\phi_{\text{new}-\beta}^{\text{Cos}-\text{NS}}$': the new model with cosine margin classifier [34] and trained by BCT with $\phi_{\text{old}}^{\text{NS}}$ as the old model. '$\phi_{\text{old}}^{\text{S}}$': using standard softmax loss as training loss. '$\phi_{\text{new}-\beta}^{\text{Cos}-\text{S}}$': the new model with cosine margin classifier class [34] and BCT with $\phi_{\text{old}}^{\text{S}}$ as the old model. '$\phi_1^{25\%}$': Model trained by 25% of training data. '$\phi_1^{90\%}$': Model trained by 90% of training data.

	IJB-C 1:1 Verification			
Comparison Pair	TAR (%) @ FAR=			
	10^{-5}	10^{-4}	10^{-3}	10^{-2}
$(\phi_{\text{old}}, \phi_{\text{old}})$	59.41	77.86	88.80	95.35
$(\phi_{\text{new}-\beta}^{\text{R152}}, \phi_{\text{old}})$	66.44	80.54	89.87	95.71
$(\phi_{\text{new}-\beta}^{\text{R152}+256\text{D}}, \phi_{\text{old}})$	68.45	80.92	89.83	95.84
$(\phi_{\text{new}-\beta}^{\text{ReLU}}, \phi_{\text{old}})$	17.02	34.70	59.82	83.69
$(\phi_{\text{old}}^{\text{S}}, \phi_{\text{old}}^{\text{S}})$	57.30	73.27	85.91	94.45
$(\phi_{\text{new}-\beta}^{\text{Cos}-\text{S}}, \phi_{\text{old}}^{\text{S}})$	48.08	67.11	84.01	94.39
$(\phi_{\text{old}}^{\text{NS}}, \phi_{\text{old}}^{\text{NS}})$	65.56	80.10	89.80	95.61
$(\phi_{\text{new}-\beta}^{\text{Cos}-\text{NS}}, \phi_{\text{old}}^{\text{NS}})$	69.72	81.81	90.39	96.10
$(\phi_1^{25\%}, \phi_1^{25\%})$	29.40	51.77	73.30	88.97
$(\phi_{\text{new}-\beta}^{25\%}, \phi_1^{25\%})$	33.51	52.21	72.78	88.84
$(\phi_1^{90\%}, \phi_1^{90\%})$	74.49	86.58	93.24	97.03
$(\phi_{\text{new}-\beta}^{90\%}, \phi_1^{90\%})$	74.61	86.27	93.19	97.08

(b) Experiments on the IJB-C 1:1 verification task.

	IJB-C 1:N Retrieval				
Comparison Pair	TNIR (%) @ FPIR=			Retrieval Rate (%)	
	10^{-3}	10^{-2}	10^{-1}	Rank-1	Rank-5
$(\phi_{\text{old}}, \phi_{\text{old}})$	36.90	59.34	79.18	87.25	92.83
$(\phi_{\text{new}-\beta}^{\text{R152}}, \phi_{\text{old}})$	57.22	68.71	82.58	89.19	94.10
$(\phi_{\text{new}-\beta}^{\text{R152}+256\text{D}}, \phi_{\text{old}})$	54.77	69.45	83.60	89.12	94.07
$(\phi_{\text{new}-\beta}^{\text{ReLU}}, \phi_{\text{old}})$	6.98	17.73	33.13	52.58	73.32
$(\phi_{\text{old}}^{\text{S}}, \phi_{\text{old}}^{\text{S}})$	31.53	54.16	73.88	85.42	92.69
$(\phi_{\text{new}-\beta}^{\text{Cos}-\text{S}}, \phi_{\text{old}}^{\text{S}})$	31.44	45.46	69.19	85.37	92.80
$(\phi_{\text{old}}^{\text{NS}}, \phi_{\text{old}}^{\text{NS}})$	40.16	64.32	81.36	89.35	94.39
$(\phi_{\text{new}-\beta}^{\text{Cos}-\text{NS}}, \phi_{\text{old}}^{\text{NS}})$	54.34	71.16	83.40	90.24	94.74
$(\phi_1^{25\%}, \phi_1^{25\%})$	11.34	26.84	54.00	71.67	82.94
$(\phi_{\text{new}-\beta}^{25\%}, \phi_1^{25\%})$	16.48	34.24	57.66	76.89	87.44
$(\phi_1^{90\%}, \phi_1^{90\%})$	57.81	74.52	87.28	91.95	95.72
$(\phi_{\text{new}-\beta}^{90\%}, \phi_1^{90\%})$	61.41	74.57	87.50	91.92	95.51

(c) Experiments on the IJB-C 1:N retrieval task.

Table 7: Robustness analysis of our proposed method against different training factors. When we use the proposed Backward Compatible Training method to train the new model, we change the network structure, feature dimension, data amount and supervision loss, respectively.

New Model	Old Model	Data	Additional Loss
ϕ_{old}	-	50%	-
ϕ_{new}^*	-	100%	-
$\phi_{\text{new}-\ell^2}$	ϕ_{old}	100%	ℓ^2 distance ϕ_{old}
$\phi_{\text{new}-\text{LwF}}$	ϕ_{old}	50%	Learning w/o Forgetting
$\phi_{\text{new}-\beta}$	ϕ_{old}	100%	Influence loss on $\mathcal{T}_{\text{old}}$
$\phi_{\text{new}-\beta-kd}$	ϕ_{old}	100%	Influence loss on $\mathcal{T}_{\text{new}}$
$\phi_{\text{new}-\beta-sys}$	ϕ_{old}	100%	Influence loss on $\mathcal{T}_{\text{new}}$

(a) Training setting for different backward-compatible (new) models. 'Old Model': The compatible target model for New model. '$\phi_{\text{new}-\ell^2}$': The new model with cosine margin classifier [34] and regularized with ℓ^2 distance to ϕ_{old} output feature. '$\phi_{\text{new}-\text{LwF}}$': The new model with cosine margin classifier [34] and adopt Learning w/o Forgetting [15] approach for the new model training. '$\phi_{\text{new}-\beta}$': The new model trained with proposed BCT. '$\phi_{\text{new}-\beta-kd}$': Trained with proposed BCT and use knowledge distillation to bypass obtaining soft supervision labels for the new classes in the new embedding training dataset. '$\phi_{\text{new}-\beta-sys}$': Trained with proposed BCT and use the feature processed by ϕ_{old} as the synthesised classifier for new classes in the growing embedding training dataset.

Comparison Pair	IJB-C 1:1 Verification TAR (%) @ FAR=			
	10^{-5}	10^{-4}	10^{-3}	10^{-2}
$(\phi_{\text{old}}, \phi_{\text{old}})$	59.41	77.86	88.80	95.35
$(\phi_{\text{new}}^*, \phi_{\text{old}})$	0.0	0.0	0.0	0.0
$(\phi_{\text{new}-\ell^2}, \phi_{\text{old}})$	0.5	3.10	10.32	31.98
$(\phi_{\text{new}-\text{LwF}}, \phi_{\text{old}})$	57.09	77.26	88.65	95.46
$(\phi_{\text{new}-\beta}, \phi_{\text{old}})$ (Proposed)	66.06	80.25	89.79	95.62
$(\phi_{\text{new}-\beta-kd}, \phi_{\text{old}})$ (Proposed)	67.82	80.34	89.60	95.74
$(\phi_{\text{new}-\beta-sys}, \phi_{\text{old}})$ (Proposed)	68.35	80.59	89.42	95.23
$(\phi_{\text{new}}^*, \phi_{\text{new}}^*)$	76.77	86.96	93.66	97.18

(b) Experiments on the IJB-C 1:1 verification task.

Comparison Pair	IJB-C 1:N Retrieval TNIR (%) @ FPIR=			Retrieval Rate (%)	
	10^{-3}	10^{-2}	10^{-1}	Rank-1	Rank-5
$(\phi_{\text{old}}, \phi_{\text{old}})$	36.90	59.34	79.18	87.25	92.83
$(\phi_{\text{new}}^*, \phi_{\text{old}})$	0.0	0.0	0.0	0.0	0.01
$(\phi_{\text{new}-\ell^2}, \phi_{\text{old}})$	0.14	0.50	2.93	8.41	20.25
$(\phi_{\text{new}-\text{LwF}}, \phi_{\text{old}})$	35.89	59.27	79.00	87.35	93.12
$(\phi_{\text{new}-\beta}, \phi_{\text{old}})$ (Proposed)	52.58	67.23	82.34	88.95	93.95
$(\phi_{\text{new}-\beta-kd}, \phi_{\text{old}})$ (Proposed)	56.20	69.02	82.50	89.01	93.96
$(\phi_{\text{new}-\beta-sys}, \phi_{\text{old}})$ (Proposed)	59.48	70.70	82.97	90.09	94.53
$(\phi_{\text{new}}^*, \phi_{\text{new}}^*)$	61.93	76.88	87.70	92.11	95.72

(c) Experiments on the IJB-C 1:N search task.

Table 8: We experiment with different approaches towards compatibility of comparison pair. In (a), we illustrate the training details of all the models we trained. In (b), we show the benchmarking results of 1:1 verification on the IJB-C dataset [18]. In (c), we show the benchmarking results of 1:N search on the IJB-C dataset.

New Model	Old Model	Data	Additional Loss
ϕ_1	-	25%	-
ϕ_2	ϕ_1	50%	Influence loss on $\mathcal{T}_1$
ϕ_3	ϕ_2	100%	Influence loss on $\mathcal{T}_2$

(a) 'Old Model': the compatible target model for New model. 'ϕ_1': model trained with 25% of training data. 'ϕ_2': Model trained by 50% of training data and proposed BCT with ϕ_1. 'ϕ_3': Model trained by all of training data and proposed BCT with ϕ_2.

Comparison Pair	IJB-C 1:1 Verification TAR (%) @ FAR=			
	10^{-5}	10^{-4}	10^{-3}	10^{-2}
(ϕ_1, ϕ_1)	29.24	41.45	73.40	89.06
(ϕ_2, ϕ_1)	36.13	56.34	75.14	89.54
(ϕ_3, ϕ_1)	32.14	53.98	71.95	88.69
(ϕ_2, ϕ_2)	58.01	75.96	87.97	95.18
(ϕ_3, ϕ_2)	64.84	80.40	89.84	95.84
(ϕ_3, ϕ_3)	73.61	85.66	92.88	96.90

(b) Experiments on the IJB-C 1:1 verification task.

Comparison Pair	IJB-C 1:N Retrieval TNIR (%) @ FPIR=			Retrieval Rate (%)	
	10^{-3}	10^{-2}	10^{-1}	Rank-1	Rank-5
(ϕ_1, ϕ_1)	11.30	22.57	54.39	71.52	82.87
(ϕ_2, ϕ_1)	19.21	39.00	59.85	78.22	87.81
(ϕ_3, ϕ_1)	14.85	36.10	56.35	77.18	87.48
(ϕ_2, ϕ_2)	38.59	56.07	78.50	86.97	92.49
(ϕ_3, ϕ_2)	50.81	66.09	82.97	89.26	94.11
(ϕ_3, ϕ_3)	59.22	74.12	86.70	91.61	95.53

(c) Experiments on the IJB-C 1:N verification task.

Table 9: Robustness analysis of our proposed method against different training factors. When we use the proposed Backward Compatible Training method to train the new model, we change the network structure, feature dimension, data amount and supervision loss, respectively.