

BT^2 : Backward-compatible Training with Basis Transformation

Yifei Zhou*

University of California, Berkeley
yifei_zhou@berkeley.edu

Zilu Li*

Cornell University
zl327@cornell.edu

Abhinav Shrivastava

University of Maryland, College Park
abhinav@cs.umd.edu

Hengshuang Zhao

University of Hong Kong
hszhao@cs.hku.hk

Antonio Torralba

MIT
torralba@mit.edu

Taipeng Tian

Meta AI
ttp@fb.com

Ser-Nam Lim

Meta AI
sernamlim@meta.com

Abstract

Modern retrieval system often requires recomputing the representation of every piece of data in the gallery when updating to a better representation model. This process is known as backfilling and can be especially costly in the real world where the gallery often contains billions of samples. Recently, researchers have proposed the idea of Backward Compatible Training (BCT) where the new representation model can be trained with an auxiliary loss to make it backward compatible with the old representation. In this way, the new representation can be directly compared with the old representation, in principle avoiding the need for any backfilling. However, follow-up work shows that there is an inherent trade-off where a backward compatible representation model cannot simultaneously maintain the performance of the new model itself. This paper reports our “not-so-surprising” finding that adding extra dimensions to the representation can help here. However, we also found that naively increasing the dimension of the representation did not work. To deal with this, we propose Backward-compatible Training with a novel Basis Transformation (BT^2). A basis transformation (BT) is basically a learnable set of parameters that applies an orthonormal transformation. Such a transformation possesses an important property whereby the original information contained in its input is retained in its output. We show in this paper how a BT can be utilized to add only the necessary amount of additional dimensions. We empirically verify the advantage of BT^2 over other state-of-the-art methods in a wide range of settings. We then further extend BT^2 to other challenging yet more practical settings, including significant changes in model architecture (CNN to Transformers), modality change, and even a series of updates in the model architecture mimicking the evolution of deep learning models in the past decade. Our code is available at <https://github.com/YifeiZhou02/BT-2>.

*Equal contribution

1. Introduction

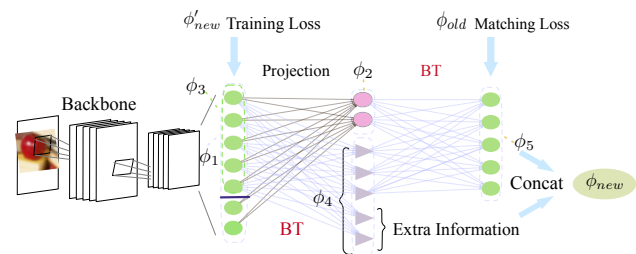


Figure 1. **Illustration of BT^2** The backbone produces a representation (light green ovals) that is encouraged to match the new model’s representation, ϕ'_{new} , via a matching and classification loss. A subset of this then goes through a BT transformation, which *retains the information* (purple triangles) from the new representation. At the same time, the new representation is then projected into a layer (pink ovals) which is combined with part of the BT-transformed new representation (the three purple triangles). This layer then goes through a BT transformation that is then encouraged to match the old model’s representation, ϕ_{old} , in effect, resulting in a backward compatible representation as the BT transformations have to inherently retain information from both ϕ'_{new} and ϕ_{old} . This is akin to the BCT procedure. The two purple triangles (i.e., what we referred to as the additional dimensions) that are not part of this are used to capture extra information in the new representation that may not be compatible. The resulting ϕ_{new} is then the representation used for all subsequent queries and new gallery samples. Refer to Section 3.4 on the definitions of $\phi_{1,2,3,4,5}$.

Modern visual retrieval systems retrieve similar images from a pool of stored data (referred to as gallery) with a given image (referred to as query). This is often done by training a model to encode all the images in the gallery and storing the resulting representations. A given query is encoded with the same model and its representation is used to retrieve the images with the most similar representations from the gallery. As better representation model design becomes available, practitioners often desire to update the representations in the

gallery with the new model to achieve better performances. The issue is that if the new model has been trained independently from the old model, the representations generated by the new model will not be compatible with those generated by the old model, which necessitates re-calculating the representations of the gallery set, a process known as “backfilling” [31]. This process gets very costly or even impossible for real world galleries which often contain billions and billions of images.

Shen *et al.* [31] therefore proposes a framework to train the new model while being compatible with the old model, known as Backward Compatible Training (BCT), with the hope of removing the need for backfilling. They propose to add an “influence loss” to the training objective of the new model to heuristically induce a backward-compatible representation. However, as pointed out by [29], adding this influence loss can significantly hurt the performance of the new model when compared to its independently trained counterpart. To mitigate this issue, subsequent works [46, 24, 47] have proposed various more sophisticated influence losses, but these endeavors have achieved limited success. Indeed, as shall be detailed in Section 3.2, it may be impossible to find a new representation model that is at the same time backward compatible yet achieves the fullest potential of the new model. In view of this, another line of work in which researchers utilize a light-weight transformation of the old representation into the new representation [37, 29, 33] looks promising. However, despite their effort to make the transformation light-weight, it still requires a costly procedure of applying a neural network to update billions of images in the gallery.

In this paper, we present findings that the conflict between backward compatibility and new model performance can be mitigated by expanding the representation space to simultaneously accommodate both the old model and the best independently trained new model. To motivate this, one can first consider an upper bound solution along this direction, where the representation of the old model is concatenated with that of an independently trained new model - being independently trained, the new model is no more limited by the backward compatibility requirement. Subsequently, queries and new samples added to the gallery are now encoded with the concatenated representations. During retrieval, since it is easy to distinguish between the gallery samples that are still of the old representations and those with the concatenated representations due to the difference in size, we can simply truncate the new representation from the query when comparing with the old representations in the gallery. This upper bound solution is “perfectly” backward compatible but suffers from two critical drawbacks: (1) it significantly increases computations due to the additional number of forward passes when computing the query representation, and (2) it begets a significant dimension expansion as a result of the concatenation. In fact, both (1) and (2) can get especially

severe after multiple model updates.

Nevertheless, such an upper bound solution provides us with the inspiration to consider adding dimensions to the representation as *necessary* while conducting BCT. We first tried naively adding dimensions (e.g., directly adding an extra 32 dimensions while training a BCT model) to the new representation, but found that this did not lead to a clear advantage as shown in Section 4. Instead, we conjecture and show that what would be more desirable is to add dimensions for the purpose of storing any information that is not compatible between the old and new representation. Towards this end, we propose a novel Backward-compatible Training with Basis Transformation (BT^2) that exploits a series of learnable basis transformations (BT) to find the information in the new representation that is incompatible with the old representation. Because a BT is basically an orthonormal transformation, the output of a BT retains the entirety of the information stored in the input (see Lemma 3). With this in mind, we introduce some clever manipulation with BT that helps to exactly “force” incompatible information in the new representation into the additional dimensions, while keeping the compatible information in the BCT representation. Fig. 1 provides a conceptual explanation of our BT^2 design.

In summary, our contributions are three-fold:

- We show that the dilemma between backward compatibility and new model development can be reconciled with extra dimensions.
- We propose BT^2 that exploits a series of learnable changes of basis to effectively exploit the extra dimensions, and verify its empirical advantage over other state-of-the-art methods in a wide range of backward compatibility tasks.
- We extend BT^2 to more challenging and practical scenarios that have not been considered by existing works to the best of our knowledge. These include significant changes in model architecture, compatibility between different modalities, and even a series of updates in the model architecture mimicking the history of deep learning in the past decade.

2. Related Works

Model Compatibility and Backward Compatibility.

Model compatibility has received an increasing amount of attention in the research community due to its practical utility [24, 37, 6, 35], where the goal is to learn a shared representation space in which representations from different models can be directly compared. In particular, backward compatibility was introduced in [31], where the authors proposed an influence loss that tries to move the new and old representations closer. Subsequent works either introduce a transformation module [17, 29, 37] or enhanced regularization loss functions [24, 46, 47]. However, some key disadvantages associated with these approaches include that some

of them depend on an auxiliary loss that prevents the new model from reaching its fullest potential, while others still require a lightweight backfilling. For the latter, a recent work known as Forward Compatible Training (FCT) [29] has been proposed that trains a lightweight transformation module to transform the old representations into new representations for backward compatibility. However, a key difference between this paper with FCT is that FCT still requires lightweight backfilling and a side-information model (which hopefully contains sufficient information to train the transformation module) but those are not required in this paper.

Continual Learning and Transfer Learning. The field of backward compatible representation learning is also related to continual learning [8, 2, 5, 30, 1] and transfer learning [51, 49, 26, 50, 23]. However, these two fields have different focuses. Continual learning focuses on training a model to perform well on a new task without forgetting the old task, and transfer learning focus on transferring a model to perform well on a different domain with the original training domain. On the other hand, backward compatible representation learning focuses on the same task, i.e., representation learning, such that the representation from the improved new model can be directly used to compare with the old model.

3. Methodology

To facilitate discussions, we follow standard notations in this field, using ϕ_1/ϕ_2 to represent the retrieval performance of using ϕ_1 for representing queries while using ϕ_2 for representing samples in the gallery. We also denote the representations from the old and final new model as ϕ_{old} and ϕ_{new} respectively. ϕ'_{new} is the independently trained new model representation while ϕ_{new} is the final new model representation after combining ϕ_{old} and ϕ'_{new} . Our goal is to learn the best ϕ_{new} possible (either when more data and/or better model architectures becomes available) while respecting the following commonly accepted criterion for backward compatible representations [31, 37].

3.1. Problem Setup

Definition 1 (Backward Compatibility). ϕ_{new} is backward compatible with ϕ_{old} , if $\forall x_i, x_j$ from the distribution of interest,

$$\begin{aligned} d(\phi_{new}(x_i), \phi_{old}(x_j)) &\geq d(\phi_{old}(x_i), \phi_{old}(x_j)), \forall y_i \neq y_j, \\ d(\phi_{new}(x_i), \phi_{old}(x_j)) &\leq d(\phi_{old}(x_i), \phi_{old}(x_j)), \forall y_i = y_j, \end{aligned}$$

where d is a distance measure, and y_i, y_j are the corresponding labels of x_i, x_j .

Alternatively, to relax the above criterion that enforce the constraints on every data point, if there is some empirical metrics $M(\phi_1, \phi_2)$ (for example, top1 accuracy for ϕ_1/ϕ_2),

we consider backward compatibility as $M(\phi_{new}, \phi_{old}) \geq M(\phi_{old}, \phi_{old})$. In addition, we desired another key property when learning ϕ_{new} :

Definition 2 (Not Hurting New Model). ϕ_{new} is said to be not hurting the new model if, $\forall x_i, x_j$ from the distribution of interest:

$$d(\phi_{new}(x_i), \phi_{new}(x_j)) = d(\phi'_{new}(x_i), \phi'_{new}(x_j)).$$

Similarly, if we relax it with an empirical metric M , the definition becomes $M(\phi_{new}, \phi_{new}) \geq M(\phi'_{new}, \phi'_{new})$. In this paper, we adopt the the negative dot product as distance metric for simplicity:

$$d(a, b) = -a^\top b$$

Note that in this paper, all final representations are normalized, so this dot product is equivalent to cosine similarity up to a constant multiplier.

3.2. Backward Compatibility vs ϕ_{new} Performance

We argue that only adding an influence loss [31, 46, 24] fails to reliably guarantee backward compatibility without hurting the new model, and this idea is formalized in Lemma 1. The implication of this Lemma 1 can be considered as an inherent trade-off between backward compatibility and ϕ_{new} performance, which inspires us to “hold” incompatible information of the new model on the additional orthogonal dimensions to avoid this conflict.

Lemma 1. *There exist cases where backward compatibility will significantly limit the potential of the new model while using negative cosine similarity as the distance metric.*

3.3. Learnable Basis Transformation

We make heavy use of basis transformation (BT). A BT is essentially a learnable change of basis represented by an orthonormal matrix, P . P can be parameterized as the exponential of a left skew-symmetric A , so that $P = e^A$, where the upper entries in A are learnable parameters. This design is made possible by the following Lemma 2.

Lemma 2. *If A is a left skew-symmetric matrix, $P = e^A$ is orthonormal where $P = e^A$ is defined as:*

$$P = e^A = \sum_{k=0}^{\infty} \frac{A^k}{k!}$$

Intuitively, for any representation $\phi(x)$, applying a change of basis on $\phi(x)$ to get $P\phi(x)$ should not lose any information and will not hurt the quality of the ϕ . This intuition can be formalized by the following Lemma 3.

Lemma 3. *If P is an orthonormal matrix of dimension $m \times m$, $\forall \phi(x_1), \phi(x_2)$ that are m dimensional vectors:*

$$\phi(x_1)^\top \phi(x_2) = (P\phi(x_1))^\top (P\phi(x_2))$$

Algorithm 1 Dimension Reduction by Learnable Change of Basis

Require: Learnable backbone F with output dimension $m+n$, learnable projection layer f mapping from dimension $m+n$ to d , BT block B_1 of $m \times m$ and B_2 of $n \times n$. Constant C and image x .

$$\begin{aligned}\phi_1 &\leftarrow F(x) \\ \phi_2 &\leftarrow f(\phi_1), \phi_2 \leftarrow \frac{\phi_2}{\|\phi_2\|} \\ \phi_3 &\leftarrow \frac{\phi_{1[:m]}}{\|\phi_{1[:m]}\|} \\ \phi_4 &\leftarrow CB_1\phi_3 \\ \phi_5 &\leftarrow B_2 \begin{bmatrix} \phi_2 \\ \phi_4[:n-d] \end{bmatrix} \\ \phi_{new} &\leftarrow \begin{bmatrix} \phi_5 \\ \phi_4[n-d:] \end{bmatrix}\end{aligned}$$

3.4. Merging ϕ'_{new} and ϕ_{old} with BT

With all the premises set up in the previous sections, we will now describe our proposal on achieving both criteria in definition 1 and 2. Further, we would like to minimize the additional dimensions required to do that (Lemma 1) as well as ensuring that ϕ_{new} requires only a single forward pass to obtain. Our proposal utilizes the BT (Lemma 2) as described in Fig. 1. We conjecture that although new and old models might differ in their model architecture or training data, they should still encode a lot of information in common. Therefore, we propose to train ϕ_{new} to automatically pick out the information from ϕ'_{new} not compatible with ϕ_{old} , and only encode this extra information on the dimension orthogonal to ϕ_{old} . This can be realized by restricting ourselves to using only learnable change of basis, thanks to the information-preserving property detailed in Lemma 3.

Referring to Fig. 1, we mainly add two BT layers. Concretely, suppose ϕ'_{new}, ϕ_{old} are of dimension m, n respectively, and we allow a dimension expansion of d , so that ϕ_{new} has a dimension of $m+d$ in our budget. Our backbone produces ϕ_1 of dimension $m+n$, and the first m dimension (referred to as ϕ_3) is trained to mimic ϕ'_{new} with the same loss that is used to train ϕ'_{new} . Notably, there is no backward influence loss on ϕ_3 so that it is hoped ϕ_3 can be as good as independent ϕ'_{new} .

After that, we pass the entire ϕ_1 into the projection layer to produce a ϕ_2 of dimension d which will be some additional features of ϕ_{old} that are not used by ϕ'_{new} . ϕ_3 is passed into the first BT layer B_1 to get ϕ_4 , allowing us to split ϕ_3 into compatible information with the old representation (the first $n-d$ dimensions) and incompatible information (the remaining $m-n+d$ dimensions). The first $n-d$ dimension is concatenated with ϕ_2 and passed into the second BT layer B_2 to mix the information of ϕ_2 and from ϕ_4 to get ϕ_5 . Similarly, ϕ_5 is trained to mimic ϕ_{old} and there is no new model training loss so it is hoped that ϕ_5 can be as compatible to the old representations as possible. Lastly,

ϕ_5 is concatenated with the remaining $m-n+d$ dimension of ϕ_4 to get the final ϕ_{new} of dimension $m+d$. Basis transformations B_1 and B_2 are designed such that all the information is preserved from ϕ_3 that is trained to match an independent model ϕ'_{new} . Algorithm 1 detailed our proposed method, with additional normalization details to ensure the information of ϕ'_{new} dominates the information of ϕ_{old} in the final representation ϕ_{new} via the factor C . Note that this does not cause ϕ_{new}/ϕ_{old} to suffer since we truncate ϕ_{new} when comparing to ϕ_{old} , effectively eliminating the extra incompatible information between ϕ'_{new} and ϕ_{old} .

Intuitively, the BT layers serve the purpose of retaining the entirety of the information between the input and the output, which as a result means that ϕ_5 is akin to a BCT procedure. It also means that any incompatible information between ϕ'_{new} and ϕ_{old} is “forced” into $\phi_4[n-d:]$ as a result of the training.

4. Experiments

We provide experimental results that (1) benchmark our method with existing backward compatible representation learning methods based on both criteria in definition 1 and 2, (2) demonstrate our method’s ability to handle cases such as data changes or what [31] referred to as open classes (*e.g.* the old model is trained on 500 Imagenet classes while the new model on 1000 Imagenet classes, with both using the same architecture), significant changes in model architecture (*e.g.*, ResNet to pretrained Transformers), different modalities, and a series changes of model architecture (mimicking the historical development of deep learning), and (3) ablating the effect of the number of extra dimensions on the performance of our method.

4.1. Datasets

This work makes use of the following datasets:

- **Cifar-100**[20]: Cifar-100 is a popular image classification dataset consisting of 60000 images in 100 classes. We use Cifar-50 to refer to the partition consisting of all the images from the first 50 classes.
- **Imagenet-1k** [9]: Imagenet-1k is a large-scale image recognition dataset for ILSVRC 2012 challenge. It has 1000 image classes with approximately 1.2k images per class. We follow the same partition as [29] where we use the images from the first 500 classes as Imagenet-500.*

4.2. Evaluation Metrics

Cumulative Matching Characteristics (CMC). CMC corresponds to the top-k accuracy, where we sort the gallery

*We use the class split from <https://gist.github.com/aaronpolhamus/964a4411c0906315deb9f4a3723aac57>.

representations by their similarity to the query representation. It is considered correct if a match with the same class is in the first k gallery representations. We report CMC top-1 and top-5 for all models.

Mean Average Precision (mAP). mAP is a standard metrics that summarizes precision and recall metrics by taking the area under the precision-recall curve. We compute the average precision in the recall range [0.0, 1.0].

4.3. Baselines

We benchmark against the following baselines to validate our method.

Independent. For this baseline, it is basically the two models ϕ'_{new} and ϕ_{old} without accounting for any backward compatibility. ϕ_{old}/ϕ_{old} and ϕ'_{new}/ϕ_{old} respectively provide a rough upper and lower bound for ϕ_{new}/ϕ_{old} , while ϕ'_{new}/ϕ'_{new} provides a rough upper bound for ϕ_{new}/ϕ_{new} .

Backward Compatible Training (BCT). BCT was introduced in Shen *et al.* [31]. As the first attempt for backward compatibility, it is frequently adopted as the baseline in many recent papers[29, 33, 17]. Specifically, BCT utilizes a classification loss but adds an “influence loss” during training to achieve backward compatibility. Denoting w_ϕ as the new representation model, w^c a trainable classification head, and w^c_{old} the fixed classification head that was trained with the old representation head, the following loss terms are used in BCT:

$$\mathcal{L}_{BCT}(\phi, w^c, x) = \mathcal{L}(w^c, \phi|x) + \lambda \mathcal{L}(w^c_{old}, \phi|x),$$

where λ is a hyperparameter to tune, and both $\mathcal{L}(w^c, \phi|x)$ and $\mathcal{L}(w^c_{old}, \phi|x)$ represent a classification loss with w^c/w^c_{old} as the classification head and ϕ as the representation. Backward propagation trains ϕ and w_ϕ .

BCT with 32 Extra Dimensions (BCT(+32)). To ensure a fair comparison with our method, where we add 32 dimensions, we create a variant of BCT with the dimension expanded by 32. We use the same loss function as BCT except that we pad the missing dimension of w^c_{old} with 0.

Regression-alleviating Compatibility Regularization (Contrast). This is a recently proposed method with a more sophisticated auxiliary loss [46] to replace influence loss:

$$\begin{aligned} \mathcal{L}_{ra-comp}(\phi_{new}, x) = & \\ & \log\left(1 + \frac{\sum_{k \in \mathcal{B}-p(x)} \exp \phi_{new}(k) \cdot \phi_{old}(k) / \tau}{\exp \phi_{new}(x) \cdot \phi_{old}(x) / \tau}\right) \\ & + \frac{\sum_{k \in \mathcal{B}-p(x)} \exp \phi_{new}(k) \cdot \phi_{new}(k) / \tau}{\exp \phi_{new}(x) \cdot \phi_{old}(x) / \tau}, \end{aligned}$$

where \mathcal{B} is the mini-batch for training, $p(x)$ is the set of samples in the minibatch with the same label as x , and τ is a temperature hyperparameter.

BT^2 (Ours). We use a dimension expansion of 32, and a classification loss that is the same as that used to train

Method	Case	CMC top1-top5	mAP@1.0
Independent	ϕ_{old}/ϕ_{old}	33.6-55.4	24.4
	ϕ'_{new}/ϕ_{old}	0.8-4.9	1.5
	ϕ'_{new}/ϕ'_{new}	62.7-74.6	49.9
BCT	ϕ_{new}/ϕ_{old}	23.5-60.4	23.9
	$\phi_{new}^{bct}/\phi_{new}^{bct}$	56.1-70.8	43.6
BCT (+32)	$\phi_{new}^{bct(+32)}/\phi_{old}$	22.1-61.3	23.8
	$\phi_{new}^{bct(+32)}/\phi_{new}^{bct(+32)}$	56.1-71.3	44.1
Contrast	$\phi_{new}^{contrast}/\phi_{old}$	26.1-61.8	25.1
	$\phi_{new}^{contrast}/\phi_{new}^{contrast}$	57.9-75.2	36.7
BT^2 (Ours)	$\phi_{new}^{bt^2}/\phi_{old}$	38.7-67.1	28.0
	$\phi_{new}^{bt^2}/\phi_{new}^{bt^2}$	64.4-78.7	53.2

Table 1. Backward compatible experiments on Cifar-50 to Cifar-100 with only data change. Both the old model and the new model uses Resnet50-128 architecture.

ϕ'_{new} , and a combination of cosine similarity loss and BCT influence loss for matching loss for ϕ_{old} . Specifically, we use the following loss as “ ϕ'_{new} training loss” for ϕ_3 :

$$\mathcal{L}_{\phi'_{new}}(\phi_3, w^c_{\phi_3}, x) = \mathcal{L}(w^c_{\phi_3}, \phi_3|x) + \lambda_1(1 - \phi_3^\top \phi'_{new}),$$

where λ_1 is a hyperparameter and ϕ'_{new} is an independently trained model, both ϕ_3 and $w^c_{\phi_3}$ are trained. Similarly, we use the following loss as “ ϕ_{old} matching loss”:

$$\mathcal{L}_{\phi_{old}}(\phi_5, x) = \lambda_2 \mathcal{L}(w^c_{old}, \phi_5|x) + \lambda_3(1 - \phi_5^\top \phi_{old}),$$

where λ_2, λ_3 are two hyperparameters and w^c_{old} is the fixed classification head used to train ϕ_{old} . C is taken to be 2. For transformer-based models, we found it helpful to apply the classification loss to the final representation ϕ_{new} instead of ϕ_3 .

4.4. Implementations Details

Experiments on Table 1, 2, 3. Experiments for these tables are carried out on 8x Nvidia 2080Ti. For all baselines and methods, transformer models are finetuned with sgd optimizer with a learning rate 0.01 and batch size 64 for 25 epochs, while ResNet50 models are trained with adam optimizer with a learning rate 0.001 and batch size 256 or 128 for 100 epochs.

Experiments on Table 4, 5, 6, 7. Experiments for these tables are carried out on 8x Nvidia A100. For all baselines and methods, transformer models are finetuned with sgd optimizer with a learning rate 0.01 and batch size 512 for 25 epochs, while AlexNet, ResNet50 models are trained with adam optimizer with a learning rate 0.001 and batch size 2048 for 100 epochs. VGGNet-13 with batch normalization is trained with adam optimizer with a learning rate 0.001 and batch size 1024 for 100 epochs.

4.5. Data Change

Cifar-50 to Cifar-100. In this experiment, the model for ϕ_{old} is a ResNet50 with an output feature dimension of size

Method	Case	CMC top1-top5	mAP@1.0
Independent	ϕ_{old}/ϕ_{old}	43.1-58.3	30.9
	ϕ'_{new}/ϕ_{old}	0.1-0.5	0.2
	ϕ'_{new}/ϕ'_{new}	67.9-81.4	52.3
BCT	$\phi_{new}^{bct}/\phi_{old}$	41.3-64.4	33.3
	$\phi_{new}^{bct}/\phi_{new}^{bct}$	63.7-79.0	51.2
BCT (+32)	$\phi_{new}^{bct(+32)}/\phi_{old}$	37.4-64.4	30.0
	$\phi_{new}^{bct(+32)}/\phi_{new}^{bct(+32)}$	65.7-80.1	52.0
Contrast	$\phi_{new}^{contrast}/\phi_{old}$	39.0-66.7	29.4
	$\phi_{new}^{contrast}/\phi_{new}^{contrast}$	65.6- 81.2	47.6
BT^2 (Ours)	$\phi_{new}^{bt^2}/\phi_{old}$	47.8-68.0	33.8
	$\phi_{new}^{bt^2}/\phi_{new}^{bt^2}$	66.5-80.9	54.4

Table 2. Backward compatible experiments on Imagenet-500 to Imagenet-1k with only data change. Both the old model and the new model uses Resnet50-128 architecture.

128, and trained on Cifar-50. The model for ϕ_{new} is also a ResNet50 but trained on the entire Cifar-100 dataset. For retrieval, we use the Cifar-100 validation set as both the gallery set and the query set.

Imagenet-500 to Imagenet-1k. Same as above except the model for ϕ_{old} is trained on Imagenet-500, and that for ϕ_{new} on the entire Imagenet-1k dataset. For retrieval, we use the Imagenet-1k validation set as both gallery and query.

Results. The results are shown in Table 1 and 2, where we observe that BCT can indeed achieve backward compatibility on large-scale image classification datasets like Imagenet and also achieves reasonable performance on $\phi_{new}^{bct}/\phi_{new}^{bct}$. However, as has also been discussed in the previous literature, $\phi_{new}^{bct}/\phi_{new}^{bct}$ is significantly influenced by the auxiliary influence loss. Comparing to the upper bound of training independently, ϕ'_{new}/ϕ'_{new} , BCT is only 63.7% and 56.1% for CMC top 1 on Imagenet and Cifar-100 respectively. Furthermore, the backward compatibility of BCT can be unstable in some of the datasets where we observe that $\phi_{new}^{bct}/\phi_{old}$ is only 23.5% for CMC top 1 on Cifar-100 while ϕ_{old}/ϕ_{old} is 33.6%. Its unstable performance might be because the influence loss in BCT does not directly encourage the model to be compatible with the old representation, but rather to be compatible with the old classification head, and the inherent conflict between $\phi_{new}^{bct}/\phi_{old}$ and $\phi_{new}^{bct}/\phi_{new}^{bct}$. We also observe a similar pattern for Contrast, indicating that even a more sophisticated influence loss may not suffice in overcoming this issue. In contrast, with the extra dimension introduced by BT^2 , we observe a significant improvement across the board. In particular, for CMC top 1 on Imagenet, BT^2 achieves 47.8% on $\phi_{new}^{bt^2}/\phi_{old}$ and 66.5% on $\phi_{new}^{bt^2}/\phi_{new}^{bt^2}$, which are a 6.5% and 2.8% improvement over BCT respectively. It shows that BT^2 can mitigate the trade-off between backward compatibility and performance of the new model, by exploiting extra dimensions. We would like to also note that just naively adding new dimension like BCT (+32) did not show a clear improvement over BCT, which highlights

Method	Case	CMC top1-top5	mAP@1.0
Independent	ϕ_{old}/ϕ_{old}	33.6-55.4	24.4
	ϕ'_{new}/ϕ_{old}	0.3-4.7	1.7
	ϕ'_{new}/ϕ'_{new}	89.5-94.3	87.5
BCT	$\phi_{new}^{bct}/\phi_{old}$	45.7-83.7	32.9
	$\phi_{new}^{bct}/\phi_{new}^{bct}$	88.7-93.5	85.7
BCT (+32)	$\phi_{new}^{bct(+32)}/\phi_{old}$	44.9- 86.2	32.7
	$\phi_{new}^{bct(+32)}/\phi_{new}^{bct(+32)}$	88.6-93.7	84.8
Contrast	$\phi_{new}^{contrast}/\phi_{old}$	45.6-81.0	32.8
	$\phi_{new}^{contrast}/\phi_{new}^{contrast}$	88.2-94.0	81.9
BT^2 (Ours)	$\phi_{new}^{bt^2}/\phi_{old}$	51.2-85.5	34.0
	$\phi_{new}^{bt^2}/\phi_{new}^{bt^2}$	90.0-94.8	88.4

Table 3. Backward compatible experiments on cifar-50 to cifar-100 with both data change and model change. The old model uses Resnet50-128 architecture, while the new model uses a transformer “ViT-B16” [10] pretrained on full training set of Imagenet21K.

the importance of adding dimensions in a principled manner.

4.6. Model Change

Cifar-50 to Cifar-100 (ResNet50 to Transformer). The setting of this experiment is similar to Cifar-50 to Cifar-100 except that the new model is finetuned from “ViT-B16” [10, 36] pretrained on entire Imagenet-21k training set.

Imagenet-500 to Imagenet-1k (ResNet50 to Transformer). The setting of this experiment is similar to Imagenet-500 to Imagenet-1k except that the new model is from the same “ViT-B16” [10] pretrained on entire Imagenet-21k training set.

Results. The results are shown in Table 4. This is a challenging setting which, to be best of our knowledge, has not been studied in the previous literature. We observe a similar pattern to the data change setting. Designing sophisticated backward compatible loss or naively adding dimensions, as in the case of Contrast and BCT (+32), did not produce any clear improvement. However, BT^2 outperforms BCT by 12.5% and 2.1% in terms of CMC top 1 on Imagenet $\phi_{new}^{bct}/\phi_{old}$ and $\phi_{new}^{bct}/\phi_{new}^{bct}$ respectively.

4.7. Modality Change

Setting. In addition to data and model change, we propose an even more challenging scenario where the modality changes. We consider the application of modality fusion, where a single gallery of image representations can support both image-to-image retrieval and text-to-image retrieval. In a standard setting, this needs to be done with two separate models: one trained for good image representations (with classification loss on a large-scale image datasets for example) and the other one trained to align the representations of images and text in the same representation space. The first model is usually good only for image-to-image retrieval, while the second model, although good for text-to-image matching, performs much worse than the first model in terms of image to image

Method	Case	CMC top1-top5	mAP@1.0
Independent	ϕ_{old}/ϕ_{old}	43.1-58.3	30.9
	ϕ'_{new}/ϕ_{old}	0.0-0.2	0.1
	ϕ'_{new}/ϕ'_{new}	78.0-87.5	72.4
BCT	$\phi_{new}^{bct}/\phi_{old}$	41.1-69.5	33.0
	$\phi_{new}^{bct}/\phi_{new}^{bct}$	74.8-86.7	66.5
BCT (+32)	$\phi_{new}^{bct(+32)}/\phi_{old}$	40.5-69.4	32.9
	$\phi_{new}^{bct(+32)}/\phi_{new}^{bct(+32)}$	75.1-87.1	66.8
Contrast	$\phi_{new}^{contrast}/\phi_{old}$	43.5-71.3	34.0
	$\phi_{new}^{contrast}/\phi_{new}^{contrast}$	72.5-86.3	58.5
BT^2 (+32)	$\phi_{new}^{bt^2}/\phi_{old}$	53.6-74.5	37.5
	$\phi_{new}^{bt^2}/\phi_{new}^{bt^2}$	76.9-88.2	70.4

Table 4. Backward compatible experiments on Imagenet-500 to Imagenet-1k with both data change and model change. The old model uses Resnet50-128 architecture, while the new model uses a transformer “ViT-B16” [10] pretrained on full training set of Imagenet21K.

retrieval. One such popular model is CLIP [27]. CLIP comprises of a pair of text and image encoder, both of which are trained to align in representation. CLIP has been shown to be very effective for text-to-image matching but its performance on image-to-image matching lags behind. Denoting the image encoder of CLIP as $\phi^{clip-img}$ and referring to Table 5, the CMC top 1 on Imagenet $\phi^{clip-img}/\phi^{clip-img}$ is only 54.7% compared to using a specialized image-to-image retrieval model ϕ_{img}/ϕ_{img} achieving 78.0%, where ϕ_{img} is a pretrained Visual Transformer “ViT-B16” [10] finetuned on Imagenet-1k classification. We believe that BT^2 has the potential to bridge this gap. Specifically, in the BT^2 setting, we use the text encoder of CLIP, denoted as $\phi^{clip-txt}$, as the old model and the ViT-B16 as ϕ'_{new} .

We measure the performances of different baselines and our method by $\phi^{clip-text}/\phi_{img}$ and ϕ_{img}/ϕ_{img} , which are the performances of text-to-image and image-to-image retrievals respectively. For the former, we simulate text-to-image by using a pretrained GPT2 [28] image captioning model to automatically generate captions for all images in Imagenet with “vit-gpt2-image-captioning” from [43]. Subsequently, during evaluation, the queries are taken from the same set of captions (for which we know the corresponding images), then encoded ($\phi^{clip-txt}$) before the nearest image encodings are retrieved. Since we know the corresponding class of the image of $\phi^{clip-txt}$, we consider the retrieval to be correct if the retrieved image is from the same class. For comparison, we use $\phi^{clip-txt}/\phi^{clip-img}$ to denote the text-to-image retrieval performance with CLIP text model and CLIP image model.

Results. First, we notice that the performance of $\phi^{clip-text}/\phi^{clip-image}$, where we measure text-to-image retrieval performance with CLIP text model and CLIP image model, is relatively low compared to other models. This is mainly because we use automatic image captioning instead

Method	Case	CMC top1-top5	mAP@1.0
Independent	$\phi^{clip-txt}/\phi^{clip-img}$	11.7-25.0	7.0
	$\phi^{clip-txt}/\phi_{img}$	0.1-0.4	0.3
	$\phi^{clip-img}/\phi^{clip-img}$	54.7-78.0	21.8
BCT	ϕ_{img}/ϕ_{img}	78.0-87.5	72.4
	$\phi^{clip-txt}/\phi_{img}^{bct}$	9.1-21.4	12.1
BCT (+32)	$\phi_{img}^{bct}/\phi_{img}^{bct}$	73.7-85.9	65.9
	$\phi^{clip-txt}/\phi_{img}^{bct(+32)}$	9.2-21.8	12.1
Contrast	$\phi_{img}^{bct(+32)}/\phi_{img}^{bct(+32)}$	73.8-85.5	65.5
	$\phi^{clip-txt}/\phi_{img}^{contrast}$	13.0-30.0	8.8
BT^2 (Ours)	$\phi_{img}^{contrast}/\phi_{img}^{contrast}$	48.0-68.2	23.1
	$\phi^{clip-text}/\phi_{img}^{bt^2}$	11.4-25.6	13.6
	$\phi_{img}^{bt^2}/\phi_{img}^{bt^2}$	77.6-87.4	71.5

Table 5. Backward compatible experiments on Imagenet-1k with modality change. The old model uses a pretrained CLIP text encoder with automatically generated text, while the new model uses a transformer pretrained on full training set of Imagenet21K.

of manually annotated ones to caption the text descriptions associated with each image, so many of them are not accurate. Second, because of the challenge of modality change and noise in the training data, BCT fails to achieve backward compatibility (CMC top 1 on $\phi^{clip-text}/\phi_{img}^{bct}$ is 9.1% compared to 11.7% on $\phi^{clip-txt}/\phi^{clip-img}$) and the performance on $\phi_{img}^{bct}/\phi_{img}^{bct}$ is also significantly hurt by 4.4% (CMC top 1 of 73.7% compared to 78.0%). On the other hand, although Contrast achieves backward compatibility, its image-to-image retrieval performance is extremely unreliable (CMC top 1 of 48.0% compared to 78.0%). Finally, we observe that BT^2 is particularly robust in this setting with rigorous backward compatibility (CMC top 1 11.4% compared to 11.7%) and marginal loss in $\phi_{img}^{bt^2}/\phi_{img}^{bt^2}$ (CMC top 1 77.6% compared to 78.0%). We also want to note that unlike the experiments of data change and model change, we set the dimension of ϕ_{img}^{bct} and $\phi^{clip-txt}$ to be 512, same as the dimension used in CLIP while still managing to keep the dimension expansion of $\phi_{img}^{bt^2}$ to 32. It shows that our BT^2 is extremely “dimension efficient” in the sense that we only need to expand the dimension by 6.25%.

4.8. Series of Model Updates

The evolution of deep learning brought in an era of very prolific scientific production, bringing us new and better model designs and architectures every so often. From the early days of the past decade (2010-2022) when AlexNet [21] was first introduced, we have seen an evolution that subsequently brought us VGGNet13 [32], ResNet50 [15], and finally ViT [10]. There are of course numerous other model designs but we will focus on this list of better-known architectures. The experiments in this section is as follow. We first train a VGGNet13, ϕ_{vgg} , that is backward compatible with AlexNet. Then, we train a ResNet50, ϕ_{res} , model that is backward compatible with ϕ_{vgg} , and finally we train a ViT,

Method	Case	CMC top1-top5	mAP@1.0	
Independent	ϕ_{alex}/ϕ_{alex}	46.6-66.3	29.1	
	ϕ_{vgg}/ϕ_{vgg}	63.2-79.0	49.6	
	ϕ_{res}/ϕ_{res}	67.9-81.4	52.3	
	ϕ_{vit}/ϕ_{vit}	78.0-87.5	72.4	
BCT	$\phi_{vgg}^{bct}/\phi_{alex}$	54.4-74.1	36.2	
	$\phi_{vgg}^{bct}/\phi_{vgg}^{bct}$	58.4-75.4	47.0	
	$\phi_{res}^{bct}/\phi_{alex}$	46.0-71.9	30.6	
	$\phi_{res}^{bct}/\phi_{vgg}^{bct}$	48.9-75.2	44.4	
	$\phi_{res}^{bct}/\phi_{res}^{bct}$	64.3-79.1	52.7	
	$\phi_{vit}^{bct}/\phi_{alex}$	54.9-82.0	36.3	
	$\phi_{vit}^{bct}/\phi_{vgg}^{bct}$	57.5-84.1	50.5	
	$\phi_{vit}^{bct}/\phi_{res}^{bct}$	70.3-85.1	57.0	
	$\phi_{vit}^{bct}/\phi_{vit}^{bct}$	73.9-86.0	65.8	
	BT^2 (ours)	$\phi_{vgg}^{bt^2}/\phi_{alex}$	56.5-75.6	37.1
		$\phi_{vgg}^{bt^2}/\phi_{vgg}^{bt^2}$	61.0-77.2	48.5
		$\phi_{res}^{bt^2}/\phi_{alex}$	56.7-78.5	37.2
		$\phi_{res}^{bt^2}/\phi_{vgg}^{bt^2}$	61.5-80.8	50.6
		$\phi_{res}^{bt^2}/\phi_{res}^{bt^2}$	66.6-80.8	56.8
		$\phi_{vit}^{bt^2}/\phi_{alex}$	57.9-83.5	37.6
		$\phi_{vit}^{bt^2}/\phi_{vgg}^{bt^2}$	62.5-86.5	52.7
$\phi_{vit}^{bt^2}/\phi_{res}^{bt^2}$		72.0-87.0	60.6	
$\phi_{vit}^{bt^2}/\phi_{vit}^{bt^2}$		75.6-87.4	68.0	

Table 6. Backward compatible experiments on a series of model updates on Imagenet-1k. Our BT^2 adds 32 extra dimensions for each update, all other models use an embedding size of 128.

ϕ_{vit} , that is compatible with ϕ_{res} . Therefore, ϕ_{vit} would be a model that is backward compatible with all the earlier models. All models are trained on Imagenet-1k training and results are tested on Imagenet-1k validation set. Our BT^2 adds 32 extra dimensions for each update, while all other models use an embedding size of 128.

Results. The results are shown in Table 6. Note that because BCT (+32) and Contrast did not show clear advantage over BCT in previous experiments, we only compare our method with BCT here. We first note the failure of BCT in this challenging setting. In the first update from AlexNet to VGGNet13, although backward compatibility is achieved (for Top1, $\phi_{vgg}^{bct}/\phi_{alex}$ is 54.4% while independent ϕ_{alex}/ϕ_{alex} is 46.6%), the performance of $\phi_{vgg}^{bct}/\phi_{vgg}^{bct}$ is significantly hurt (for Top1, $\phi_{vgg}^{bct}/\phi_{vgg}^{bct}$ is 58.4% while independent ϕ_{vgg}/ϕ_{vgg} is 63.2%). Furthermore, after another round of update in model architecture from VGGNet13 to ResNet50, the model does not maintain a decent backward compatibility with its former versions ϕ_{alex} and ϕ_{vgg}^{bct} with a loss of 0.6% on $\phi_{res}^{bct}/\phi_{alex}$ and 4.3% on $\phi_{res}^{bct}/\phi_{vgg}^{bct}$ compared to independent ϕ_{alex}/ϕ_{alex} and ϕ_{vgg}/ϕ_{vgg} in Top1. Although its backward compatibility is improved after updating to ϕ_{vit}^{bct} possibly because of the power of pretraining for ViT, it still hurts the performance of the new model sig-

Method	Case	CMC top1-top5	mAP@1.0
Independent	ϕ_{old}/ϕ_{old}	43.1-58.3	30.9
	ϕ'_{new}/ϕ_{old}	0.0-0.2	0.1
	ϕ'_{new}/ϕ'_{new}	78.0-87.5	72.4
BCT	$\phi_{new}^{bct}/\phi_{old}$	41.1-69.5	33.0
	$\phi_{new}^{bct}/\phi_{new}^{bct}$	74.8-86.7	66.5
BT^2 (+8)	$\phi_{new}^{+8}/\phi_{old}$	50.7-75.9	37.1
	$\phi_{new}^{+8}/\phi_{new}^{+8}$	74.8-87.0	66.0
BT^2 (+16)	$\phi_{new}^{+16}/\phi_{old}$	51.6- 76.1	37.6
	$\phi_{new}^{+16}/\phi_{new}^{+16}$	76.4-88.1	69.0
BT^2 (+32)	$\phi_{new}^{+32}/\phi_{old}$	53.6 -74.5	37.5
	$\phi_{new}^{+32}/\phi_{new}^{+32}$	76.9-88.2	70.4
BT^2 (+128)	$\phi_{new}^{+128}/\phi_{old}$	53.4-75.0	37.9
	$\phi_{new}^{+128}/\phi_{new}^{+128}$	77.8-88.5	71.1

Table 7. Dimension ablation experiments on Imagenet-500 to Imagenet-1k with both data change and model change. The old model uses Resnet50-128 architecture, while the new model uses a transformer pretrained on full training set of Imagenet21K. Our BT^2 adds 32 extra dimensions for each update, all other models use an embedding size of 128.

nificantly (for Top1, $\phi_{vit}^{bct}/\phi_{vit}^{bct}$ is only 73.9% compared to independent ϕ_{vit}/ϕ_{vit} with 78.0%). On the contrary, we observe that by a strategic use of extra dimensions, our BT^2 maintains backward compatibility at each stage of the sequential updates and closes the gap between the performance of the independent new models and the models trained with BCT. In summary, BT^2 shows a clear advantage on all rows across the board with gains up to 12.6%.

5. Ablations on Extra Dimensions

We now answer the question on how many additional dimensions are needed for BT^2 . We carry out ablation experiments in the setting of ResNet50 to ViT (with both data change and model change) on Imagenet-1k. All settings are the same as in Section 4.6 except we vary the number of extra dimensions in our BT^2 . Results of the ablations are shown in Table 7, where we compare BCT (no extra dimension) with BT^2 extra 8, 16, 32, 128 dimensions. We observe that BT^2 already shows a clear advantage over BCT with as few as 8 extra dimensions, showing the effectiveness of our proposed Basis Transformation Block. As extra dimensions grow from 8 to 32, though less significant, we observe an overall trend of gradual improvement both in terms of $\phi_{new}^{+n}/\phi_{new}^{+n}$ and $\phi_{new}^{+n}/\phi_{old}$. As a result, the CMC top 1 of BT^2 (+32) is 2.9% and 2.1% higher than BT^2 (+8) in terms of $\phi_{new}^{+n}/\phi_{new}^{+n}$ and $\phi_{new}^{+n}/\phi_{old}$ respectively. Finally, we notice that the improvement from BT^2 (+32) to BT^2 (+128) is somewhat marginal by -0.2% and 0.9% in terms of $\phi_{new}^{+n}/\phi_{new}^{+n}$ and $\phi_{new}^{+n}/\phi_{old}$ respectively. This shows that some of the information of ϕ'_{new} and ϕ_{old} can be shared so that we do not need as much as +128 to capture extra

information from ϕ'_{new} .

6. Conclusions

We presented BT^2 in this paper, a method for backward compatibility that makes use of additional dimensions *efficiently*. In spite of this, one of the open questions following this work is that the size of the representation will still grow over time especially after multiple model updates. Eventually, system practitioners will have to fully backfill the gallery to reset. However, we hope that BT^2 will “buy enough time” to backfill real world galleries with ϕ'_{new} that usually can contain millions of samples.

References

- [1] Hongjoon Ahn, Donggyu Lee, Sungmin Cha, and Taesup Moon. Uncertainty-based continual learning with adaptive regularization. *CoRR*, abs/1905.11614, 2019. [3](#)
- [2] Rahaf Aljundi, Punarjay Chakravarty, and Tinne Tuytelaars. Expert gate: Lifelong learning with a network of experts, 2016. [3](#)
- [3] Aurélien Bellet, Amaury Habrard, and Marc Sebban. Metric learning. *Synthesis lectures on artificial intelligence and machine learning*, 9(1):1–151, 2015. [11](#)
- [4] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1798–1828, 2013. [11](#)
- [5] Arslan Chaudhry, Marc Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. Efficient lifelong learning with A-GEM. *CoRR*, abs/1812.00420, 2018. [3](#)
- [6] Ken Chen, Yichao Wu, Haoyu Qin, Ding Liang, Xuebo Liu, and Junjie Yan. R³ adversarial network for cross model face recognition. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9860–9868, 2019. [2](#), [11](#)
- [7] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. A simple framework for contrastive learning of visual representations. *CoRR*, abs/2002.05709, 2020. [11](#)
- [8] Matthias DeLange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Ales Leonardis, Greg Slabaugh, and Tinne Tuytelaars. A continual learning survey: Defying forgetting in classification tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2021. [3](#)
- [9] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. [4](#)
- [10] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *CoRR*, abs/2010.11929, 2020. [6](#), [7](#)
- [11] Jean Gallier. *Geometric Methods and Applications For Computer Science and Engineering*, volume 38. 01 2011. [12](#)
- [12] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Ávila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap your own latent: A new approach to self-supervised learning. *CoRR*, abs/2006.07733, 2020. [11](#)
- [13] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, Bilal Piot, koray kavukcuoglu, Remi Munos, and Michal Valko. Bootstrap your own latent - a new approach to self-supervised learning. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 21271–21284. Curran Associates, Inc., 2020. [11](#)
- [14] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. [11](#)
- [15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015. [7](#)
- [16] Elad Hoffer and Nir Ailon. Deep metric learning using triplet network. In *International workshop on similarity-based pattern recognition*, pages 84–92. Springer, 2015. [11](#)
- [17] Weihua Hu, Rajas Bansal, Kaidi Cao, Nikhil Rao, Karthik Subbian, and Jure Leskovec. Learning backward compatible embeddings. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. ACM, aug 2022. [2](#), [5](#)
- [18] Ashish Jaiswal, Ashwin Ramesh Babu, Mohammad Zaki Zadeh, Debapriya Banerjee, and Fillia Makedon. A survey on contrastive self-supervised learning. *Technologies*, 9(1):2, 2020. [11](#)
- [19] Mahmut Kaya and Hasan Şakir Bilge. Deep metric learning: A survey. *Symmetry*, 11(9):1066, 2019. [11](#)
- [20] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. Cifar-100 (canadian institute for advanced research). [4](#)
- [21] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C.J. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012. [7](#)
- [22] Weiyang Liu, Yandong Wen, Zhiding Yu, Ming Li, Bhiksha Raj, and Le Song. Spheraface: Deep hypersphere embedding for face recognition. *CoRR*, abs/1704.08063, 2017. [11](#)
- [23] Jie Lu, Vahid Behbood, Peng Hao, Hua Zuo, Shan Xue, and Guangquan Zhang. Transfer learning using computational intelligence: A survey. *Knowledge-Based Systems*, 80:14–23, 2015. 25th anniversary of Knowledge-Based Systems. [3](#)
- [24] Qiang Meng, Chixiang Zhang, Xiaoqiang Xu, and Feng Zhou. Learning compatible embeddings. *CoRR*, abs/2108.01958, 2021. [2](#), [3](#), [11](#)
- [25] Kevin Musgrave, Serge J. Belongie, and Ser-Nam Lim. A metric learning reality check. *CoRR*, abs/2003.08505, 2020. [11](#)

- [26] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, 2010. 3
- [27] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. *CoRR*, abs/2103.00020, 2021. 7, 11
- [28] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019. 7
- [29] Vivek Ramanujan, Pavan Kumar Anasosalu Vasu, Ali Farhadi, Oncel Tuzel, and Hadi Pouransari. Forward compatible training for large-scale embedding retrieval systems. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2022. 2, 3, 4, 5, 11
- [30] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, and Christoph H. Lampert. icarl: Incremental classifier and representation learning. *CoRR*, abs/1611.07725, 2016. 3
- [31] Yantao Shen, Yuanjun Xiong, Wei Xia, and Stefano Soatto. Towards backward-compatible representation learning. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6367–6376, 2020. 2, 3, 4, 5, 11
- [32] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 7
- [33] Shupeng Su, Binjie Zhang, Yixiao Ge, Xuyuan Xu, Yexin Wang, Chun Yuan, and Ying Shan. Privacy-preserving model upgrades with bidirectional compatible training in image retrieval, 2022. 2, 5
- [34] Yumin Suh, Bohyung Han, Wonsik Kim, and Kyoung Mu Lee. Stochastic class-based hard example mining for deep metric learning. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7244–7252, 2019. 11
- [35] Frederik Träuble, Julius von Kügelgen, Matthäus Kleindessner, Francesco Locatello, Bernhard Schölkopf, and Peter Gehler. Backward-compatible prediction updates: A probabilistic approach, 2021. 2
- [36] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. 6
- [37] Chien-Yi Wang, Ya-Liang Chang, Shang-Ta Yang, Dong Chen, and Shang-Hong Lai. Unified representation learning for cross model compatibility. *CoRR*, abs/2008.04821, 2020. 2, 3
- [38] Feng Wang, Weiyang Liu, Haijun Liu, and Jian Cheng. Additive margin softmax for face verification. *CoRR*, abs/1801.05599, 2018. 11
- [39] Feng Wang, Xiang Xiang, Jian Cheng, and Alan L. Yuille. Normface: L_2 hypersphere embedding for face verification. *CoRR*, abs/1704.06369, 2017. 11
- [40] Hao Wang, Yitong Wang, Zheng Zhou, Xing Ji, Zhifeng Li, Dihong Gong, Jingchao Zhou, and Wei Liu. Cosface: Large margin cosine loss for deep face recognition. *CoRR*, abs/1801.09414, 2018. 11
- [41] Xun Wang, Xintong Han, Weilin Huang, Dengke Dong, and Matthew R. Scott. Multi-similarity loss with general pair weighting for deep metric learning, 2019. 11
- [42] Xun Wang, Xintong Han, Weiling Huang, Dengke Dong, and Matthew R. Scott. Multi-similarity loss with general pair weighting for deep metric learning. *CoRR*, abs/1904.06627, 2019. 11
- [43] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. Huggingface’s transformers: State-of-the-art natural language processing. *CoRR*, abs/1910.03771, 2019. 7, 12, 13
- [44] Chao-Yuan Wu, R. Manmatha, Alexander J. Smola, and Philipp Krähenbühl. Sampling matters in deep embedding learning. *CoRR*, abs/1706.07567, 2017. 11
- [45] Andrew Zhai and Hao-Yu Wu. Making classification competitive for deep metric learning. *CoRR*, abs/1811.12649, 2018. 11
- [46] Binjie Zhang, Yixiao Ge, Yantao Shen, Yu Li, Chun Yuan, Xuyuan Xu, Yexin Wang, and Ying Shan. Hot-refresh model upgrades with regression-alleviating compatible training in image retrieval. *CoRR*, abs/2201.09724, 2022. 2, 3, 5
- [47] Binjie Zhang, Yixiao Ge, Yantao Shen, Shupeng Su, Fanzi Wu, Chun Yuan, Xuyuan Xu, Yexin Wang, and Ying Shan. Towards universal backward-compatible representation learning, 2022. 2, 11
- [48] Daokun Zhang, Jie Yin, Xingquan Zhu, and Chengqi Zhang. Network representation learning: A survey. *IEEE Transactions on Big Data*, 6(1):3–28, 2020. 11
- [49] Yu Zhang and Qiang Yang. A survey on multi-task learning. *CoRR*, abs/1707.08114, 2017. 3
- [50] Peilin Zhao and Steven C. H. Hoi. Otl: A framework of online transfer learning. In *ICML*, 2010. 3
- [51] Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. A comprehensive survey on transfer learning. *CoRR*, abs/1911.02685, 2019. 3