DearKD: Data-Efficient Early Knowledge Distillation for Vision Transformers

Xianing Chen[1]*, Qiong Cao[2]†, Yujie Zhong[3], Jing Zhang[4], Shenghua Gao[156]†, Dacheng Tao[24]
[1]ShanghaiTech University, [2]JD Explore Academy, [3]Meituan Inc.,
[4]The University of Sydney, [5]Shanghai Engineering Research Center of Intelligent Vision and Imaging,
[6]Shanghai Engineering Research Center of Energy Efficient and Custom AI IC
{chenxn1,gaoshh}@shanghaitech.edu.cn {mathqiong2012,dacheng.tao}@gmail.com
jaszhong@hotmail.com jing.zhang1@sydney.edu.au

Abstract

Transformers are successfully applied to computer vision due to their powerful modeling capacity with self-attention. However, the excellent performance of transformers heavily depends on enormous training images. Thus, a data-efficient transformer solution is urgently needed. In this work, we propose an early knowledge distillation framework, which is termed as DearKD, to improve the data efficiency required by transformers. Our DearKD is a two-stage framework that first distills the inductive biases from the early intermediate layers of a CNN and then gives the transformer full play by training without distillation. Further, our DearKD can be readily applied to the extreme data-free case where no real images are available. In this case, we propose a boundary-preserving intra-divergence loss based on DeepInversion to further close the performance gap against the full-data counterpart. Extensive experiments on ImageNet, partial ImageNet, data-free setting and other downstream tasks prove the superiority of DearKD over its baselines and state-of-the-art methods.

1. Introduction

Transformers [4, 14, 47] have shown a domination trend in NLP studies owing to their strong ability in modeling long-range dependencies by the self-attention mechanism. Recently, transformers are applied to various computer vision tasks and achieve strong performance [7, 15, 32]. However, transformers require an enormous amount of training data since they lack certain inductive biases (IB) [12, 15, 46, 52]. Inductive biases can highly influence the generalization of learning algorithms, independent of data, by pushing learning algorithms towards particular solutions [16, 17, 35].

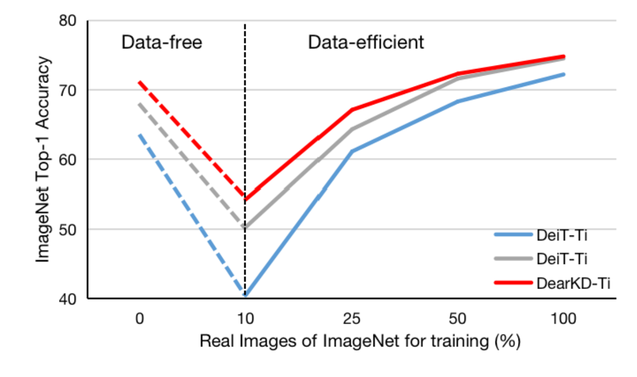


Figure 1. **Illustration of data-efficient of our DearKD.** We compare the data-efficient properties of DearKD in three situations with different numbers of real training images: the full ImageNet, the partial ImageNet and the data-free case (i.e. without any real images) with DeiT and DeiT⚗.

Unlike transformers, CNNs are naturally equipped with strong inductive biases by two constraints: locality and weight sharing mechanisms in the convolution operation. Thus, CNNs are sample-efficient and parameter-efficient due to the translation equivariance properties [12, 41, 42].

Recently, some researchers have proposed to explicitly insert convolution operations into vision transformers to introduce inductive biases [11, 18, 30, 50–52]. However, the forcefully modified structure may destroy the intrinsic properties in transformers and reduce their capacity.

Another line of work [46] utilizes Knowledge Distillation (KD) [23] to realize data-efficient transformers. By distillation, the inductive biases reflected in the dark knowledge from the teacher network can be transferred to the student [1]. DeiT [46], as a typical method in this line, has successfully explored the idea of distilling knowledge from CNNs to transformers and greatly increased the data efficiency of transformer training. Nevertheless, DeiT still suffers two drawbacks:

*This work was done when Xianing Chen was intern at JD Explore Academy.

†Corresponding authors.

Firstly, some works [11, 51] reveal that *inserting convolutions to the early stage of the network brings the best performance*, while DeiT only distills from the classification logits of the CNN and thus makes it difficult for the early (i.e. shallow) transformer layers to capture the inductive biases. Furthermore, the distillation throughout the training implicitly hinders transformers from learning their own inductive biases [12] and stronger representations [11].

To solve these problems, we propose a two-stage learning framework, named as Data-efficient EARly Knowledge Distillation (DearKD), to further push the limit of data efficiency of training vision transformers. Here the term 'early' refers to two novel designs in our proposed framework: knowledge distillation in the early layers in transformers and in the early stage of transformer training. **First**, we propose to distill from both the classification logits and the intermediate layers of the CNN, which can provide more explicit learning signals for the intermediate transformer layers (especially the early layers) to capture the inductive biases. Specifically, we draw the inspiration from [10] and design a Multi-Head Convolutional-Attention (MHCA) layer to better mimic a convolutional layer without constraining the expressive capacity of self-attention. Further, we propose an aligner module to solve the problem of feature misalignment between CNN features and transformers tokens. **Second**, the distillation only happens in the first stage of DearKD training. We let transformers learn their own inductive biases in the second stage, in order to fully leverage the flexibility and strong expressive power of self-attention.

To fully explore the power of DearKD with respect to data efficiency, we investigate DearKD in three situations with different number of real training images (Figure 1): the full ImageNet [13], the partial ImageNet and the data-free case (i.e. without any real images). In the extreme case where no real images are available, networks can be trained using data-free knowledge distillation methods [8, 34, 55]. In this work, we further enhance the performance of transformer networks under the data-free setting by introducing a boundary-preserving intra-divergence loss based on DeepInversion [55]. The proposed loss significantly increases the diversity of the generated images by keeping the positive samples away from others in the latent space while maintaining the class boundaries.

Our main contributions are summarized as follows:

- We introduce DearKD, a two-stage learning framework for training vision transformers in a data-efficient manner. In particular, we propose to distill the knowledge of intermediate layers from CNNs to transformers in the early phase, which has never been explored in previous works.

- We investigate DearKD in three different settings and propose an intra-divergence loss based on DeepInversion to greatly diversify the generated images and further improve the transformer network in the data-free situation.

- With the full ImageNet, our DearKD achieves state-of-the-art performance on image classification with similar or less computation. Impressively, training DearKD with only 50% ImageNet data can outperform the baseline transformer trained with all data. Last but not least, the data-free DearKD based on DeiT-Ti achieves 71.2% on ImageNet, which is only 1.0% lower than its full-ImageNet counterpart.

2. Related work

Knowledge Distillation. Knowledge Distillation [23] is a fundamental training technique, where a student model is optimized under the effective information transfer and supervision of a teacher model or ensembles. Hinton [23] performed knowledge distillation via minimizing the distance between the output distribution statistics between student and teacher networks to let the student learn dark knowledge that contains the similarities between different classes, which are not provided by the ground-truth labels. To learn knowledge from teacher network with high fidelity, [58] further took advantage of the concepts of attention to enhance the performance of the student network. [20] focus on transferring activation boundaries formed by hidden neurons. [43] proposed to match the Jacobians. [31] proposed to distill the structured knowledge. Moreover, [25] proposed a Transformers distillation method to transfer the plenty of knowledge encoded in a large BERT [14] to a small student Transformer network. However, all of them do not consider the problem of distillation between two networks with different architectures. Moreover, the teacher network has lower capacity than the student network in our setting.

Vision Transformers. With the success of Transformers [47] in natural language processing, many studies [7, 15, 40, 46] have shown that they can be applied to the field of computer vision as well. Since they lack inductive bias, they indeed learn inductive biases from amounts of data implicitly and lag behind CNNs in the low data regime [15]. Recently, some works try to introduce CNNs into vision transformers explicitly [9, 11, 18, 30, 50–52, 60]. However, their forcefully modified structure destroyed the intrinsic properties in transformers. [12] introduced local inductive bias in modeling local visual structures implicitly, which still learns local information through training from amounts of data. [46] proposed to distill knowledge from CNNs to transformers which does not consider the differences in their inherent representations and the Transformers intrinsic inductive biases. Thus, we propose the two-stage learning framework for Transformers to learn convolutional as well as their own

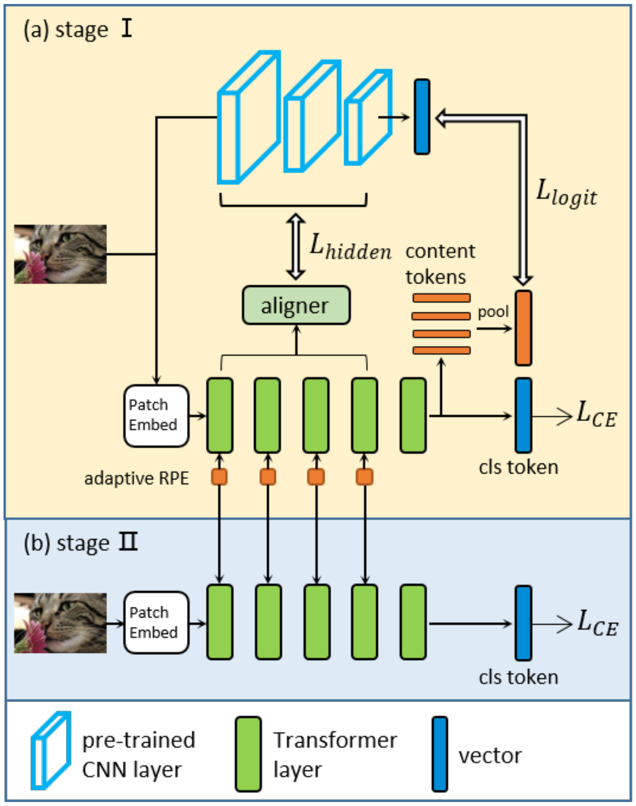


Figure 2. **The pipeline of our proposed method.** (a) The convolutional inductive biases knowledge distillation phase. (b) The transformers instrinsic inductive biases learning phase.

inductive biases.

Data-Free KD. Data-Free KD [33] aims to learn a student model from a cumbersome teacher without accessing real-world data. The existing works can be roughly divide into two categories: GAN-based and prior-based methods. GAN-based methods [8, 34, 54, 62] synthesized training samples through maximizing response on the discriminator. Prior-based methods [5] provide another perspective for data-free KD, where the synthetic data are forced to satisfy a pre-defined prior, such as total variance prior [3, 36] and batch normalization statistics [5, 8]. However, they all has the problem of mode collapse [6, 44], so we propose a boundary-preserving intra-divergence loss for DeepInversion [55] to generate diverse samples.

3. Data-efficient Early Knowledge Distillation

In this section, we first recap the preliminaries of Vision Transformers, and then introduce our proposed two-stage learning framework DearKD.

Preliminary. Vanilla multi-head self-attention (MHSA) [47] is based on a trainable associative memory with (key, value) vector pairs. Specifically, input sequences $X \in R^{T\times d}$ are first linearly projected to queries (Q), keys (K) and values (V) using projection matrices, i.e. $(Q, K, V) = (XW^Q, XW^K, XW^V)$, where $W^{Q/K/V} \in R^{d\times d}$ denotes the projection matrix for query, key, and value, respectively. Then, to extract the semantic dependencies between each parts, a dot product attention scaled and normalized with a Softmax layer is performed. The sequences of values are then weighted by the attention. This self-attention operation is repeated h times to formulate the MHSA module, where h is the number of heads. Finally, the output features of the h heads are concatenated along the channel dimension to produce the output of MHSA.

$$\begin{aligned}\text{MHSA}(\boldsymbol{X}) &= \boldsymbol{AXW}^V \\ \boldsymbol{A} &= \text{Softmax}(\boldsymbol{QK})\end{aligned} \tag{1}$$

Inductive Biases Knowledge Distillation. It is revealed in [11,51] that convolutions in the early stage of the network can significantly enhance the performance since local patterns (like texture) can be well captured by the convolution in the early layers. Therefore, providing explicit guidance of inductive biases to the early transformer layers becomes crucial for improving data efficiency. However, in the later phase, this guidance may restrict the transformer from fully exploring its expressive capacity. To this end, we propose a two-stage knowledge distillation framework DearKD (Figure 2) for learning inductive biases for transformers, which is elaborated in the following.

3.1. DearKD: Stage I

Multi-Head Convolutional-Attention (MHCA). Recently, [10] proves that a multi-head self-attention layer with N_h heads and a relative positional encoding of dimension $D_p \geq 3$ can express any convolutional layer of kernel size $\sqrt{N_h} \times \sqrt{N_h}$ by setting the quadratic encoding:

$$\begin{aligned}&\boldsymbol{v}^{(h)} := -\alpha^{(h)}\left(1, -2\boldsymbol{\Delta}_1^{(h)}, -2\boldsymbol{\Delta}_2^{(h)}\right) \\ &\boldsymbol{r}_{\boldsymbol{\delta}} := \left(\|\boldsymbol{\delta}\|^2, \boldsymbol{\delta}_1, \boldsymbol{\delta}_2\right) \\ &\boldsymbol{W}_{\text{qry}} = \boldsymbol{W}_{\text{key}} := \boldsymbol{0}, \quad \widehat{\boldsymbol{W}_{\text{key}}} := \boldsymbol{I}\end{aligned} \tag{2}$$

where the learned parameters $\boldsymbol{\Delta}^{(h)} = \left(\boldsymbol{\Delta}_1^{(h)}, \boldsymbol{\Delta}_2^{(h)}\right)$ and $\alpha^{(h)}$ control the center and width of attention of each head, $\boldsymbol{\delta} = (\boldsymbol{\delta}_1, \boldsymbol{\delta}_2)$ is fixed and indicates the relative shift between query and key pixels.

Motivated by [10], we propose a Multi-Head Convolutional-Attention (MHCA) layer to enable a transformer layer to act as a convolution layer by using the relative positional self-attention [40]. Specifically, given an input $X \in R^{T\times d}$, our MHCA layer performs multi-head self-attention as follows:

$$\begin{aligned}\text{MHCA}(\boldsymbol{X}) &= \boldsymbol{AXW}^V \\ \boldsymbol{A} &= \text{Softmax}(\boldsymbol{QK} + \boldsymbol{v}^{(h)}\boldsymbol{r}_{ij})\end{aligned} \tag{3}$$

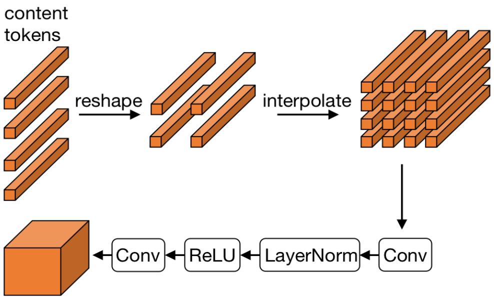


Figure 3. **Illustration of the aligner.** The aligner aligns transformer tokens to have the same size of convolution features by the stacking of reshape, bilinear interpolate, depth-wise convolution, LayerNorm and ReLU layers.

where $v^{(h)}$ contains a learnable parameter $\alpha^{(h)}$ (see Equation (2)) to adaptively learn appropriate scale of the relative position embedding (adaptive RPE). To prevent the network from falling into the local optimum where the attention highly focuses on the local information, we add a dropout layer after the adaptive RPE.

Different from MHSA in Equation (1), the proposed MHCA consists of two parts, i.e., the content part and position part, to incorporate the relative positional information. The former learns the non-local semantic dependencies described above, and the latter makes the attention aware of local details.

Early Knowledge Distillation. Now we consider the distillation of the convolutional inductive biases with the proposed MHCA. To capture the inductive biases and provide rich spatial information and local visual patterns for the intermediate transformer layers, we propose to distill from the intermediate layers of the CNN to transformers in the first stage. The objective is formulated as follows:

$$L_{\text{hidden}} = MSE(\text{aligner}(H^S), H^T) \tag{4}$$

where $H^S \in R^{l \times d}$ and $H^T \in R^{h \times w \times c}$ refer to the content tokens of student and the feature map of teacher networks, respectively. The major difficulty is that the feature maps of the CNN and the transformer tokens are in different shapes, and therefore it is infeasible to apply a distillation loss on top directly. To tackle the problem of feature misalignment, we design an aligner module to match the size of the content tokens H^S to that of H^T by the stacking of reshape. As shown in Figure 3, the aligner includes depth-wise convolution [45], LayerNorm [2] and ReLU layers. Note that, to the best of our knowledge, this work is the first to explore the knowledge distillation from the intermediate layers of the CNNs to transformers.

In addition to imitating the behaviors of intermediate CNN layers, we adopt the commonly used divergence between the teacher and student network logits in knowledge distillation. Instead of adding an additional distillation token [46] which requires additional trained CNNs networks when fine-tuning on downstream tasks, we directly pool the content tokens following [21, 38] which contains discriminative information and is consistent with the design principles of CNNs. The objective with hard-label distillation [46] is as follow:

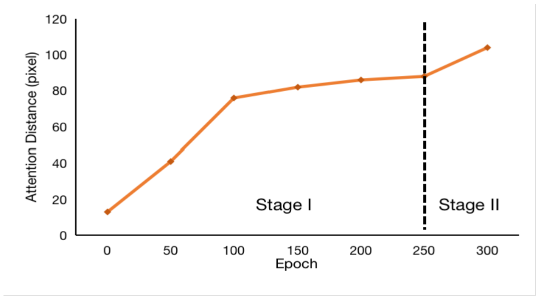


Figure 4. **The average attention distance of our DearKD for each epoch.**

$$L_{\text{logit}} = L_{\text{CE}}(logit, y_t) \tag{5}$$

where $y_t = argmax(logit_T)$ is the hard decision of the teacher.

The overall loss function is as follows:

$$L = \alpha L_{\text{CE}} + (1 - \alpha) L_{\text{logit}} + \beta L_{\text{hidden}} \tag{6}$$

where L_{CE} is the cross-entropy loss for the [CLS] token.

3.2. DearKD: Stage II

Transformers Instrinsic Inductive Biases Learning. Considering that transformers have a larger capacity than CNNs, we propose to encourage the transformers to learn their own inductive biases in a second stage. This is a critical step to leverage their flexibility and strong expressive power fully. To this end, we formulate the objective of stage II as follows:

$$L = L_{\text{CE}}(logit, y) \tag{7}$$

Note that the relative position encoding in stage I is unchanged. In this stage, the network will learn to explore a larger reception field to form the non-local representation automatically. We calculate the average attention distance of each layer in DearKD for each epoch. The results are shown in Figure 4. It can be observed that with the usage of convolutional IBs knowledge distillation, the transformer layers in the first stage will focuse on modeling locality. After training our model in the second stage, the model escapes the locality, and thus, the intrinsic IBs of Transformers can be learned automatically.

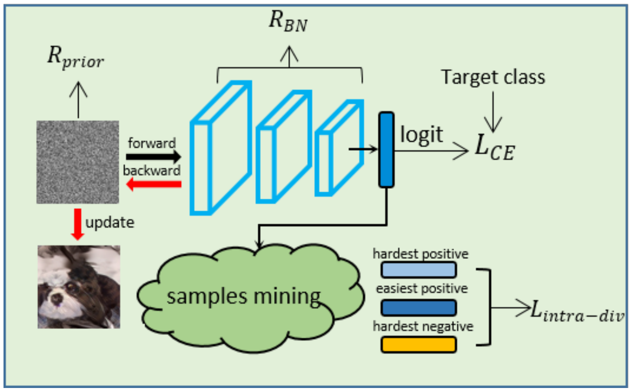
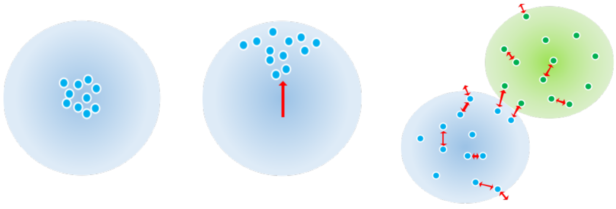


Figure 5. **The pipeline of our proposed DF-DearKD.**



(a) DeepInversion (b) ADI (c) Ours

Figure 6. **The concept of the proposed boundary-preserving intra-divergence loss.** Given a set of samples in the latent space (shown as dots), the boundary-preserving intra-divergence loss in (c) pushes the easiest positive samples away from others (shown as red arrows between the same class samples) while keep the activation boundaries (shown as circle) unaffected.

4. DF-DearKD: Training without Real Images

To fully explore the power of DearKD with respect to data efficiency, we investigate it in the extreme setting (i.e. data-free) where no real images are available. In this section, we propose DF-DearKD, a data-free variant of DearKD, for crafting a transformer network without accessing any real image. Compared to DearKD, DF-DearKD has an extra image generation component, as illustrated in Figure 5. In the following, we first briefly review the closely related method DeepInversion [55], and then introduce a novel boundary-preserving intra-divergence loss to further increase the diversity of the generated samples.

DeepInversion. Assume that we have access to a trained convolution classifier as a teacher model. Given a randomly initialized input $x \in \mathbb{R}^{H\times W\times C}$ and the corresponding target label y, DeepInversion [55] synthesized the image by optimizing

$$x = \arg\min_{x} L_{\text{CE}}(x, y) + R(x) + L_{\text{diversity}}(x, y) \tag{8}$$

where $L_{\text{CE}}(\cdot)$ is the cross-entropy loss for classification. $R(\cdot)$ is the image regularization term to steer x away from unrealistic images and towards the distribution of images presented. $L_{\text{diversity}}(\cdot)$ is the diversity loss to avoid repeated and redundant synthetic images. Specifically, R consists of two terms: the prior term R_{prior} [36] that acts on image priors and the BN regularization term R_{BN} that regularizes feature map distributions:

$$R(x) = R_{\text{prior}}(x) + R_{\text{BN}}(x) \tag{9}$$

where R_{prior} penalizes the total variance and l2 norm of x, respectively. R_{BN} matches the feature statistics, i.e., channel-wise mean $\mu(x)$ and variance $\sigma^2(x)$ of the current batch to those cached in the BN [24] layers at all levels.

Boundary-preserving intra-divergence loss. To synthesize diverse images, Adaptive DeepInversion (ADI) [55] proposes a competition scheme to encourage the synthesized images out of student's learned knowledge and to cause student-teacher disagreement. However, it usually generates hard and ambiguous samples. To address the over-clustering of the embedding space (Figure 6a and 6b), which is similar to the mode collapse problem [6, 44], we propose the boundary-preserving intra-divergence loss to keep the easiest positive samples away from others in the latent space while the class boundaries are unaffected. Figure 6c illustrates the main idea of our proposed loss. Specifically, for each anchor image x_{a} within a batch, the easiest positive samples [53] are the most similar images that have the same label as the anchor images:

$$x_{\text{ep}} = \arg\min_{x:C(x)=C(x_{\text{a}})} dist(f(x_{\text{a}}), f(x)) \tag{10}$$

where $dist(f(x_a), f(x)) = \|f(x_{\text{a}}) - f(x)\|_2$ measures the euclidean distance between two samples in the latent space. Inspired by the finding that when two latent codes are close, the corresponding images are similar [53], we increase the intra-class diversity by maximizing the distance between the latent code of the easiest pair of images:

$$L_{\text{ep}}(x) = -dist(f(x_{\text{a}}), f(x_{\text{ep}})) \tag{11}$$

This loss encourages the optimizer to explore the latent space inside the whole decision boundaries. However, this will push some generated samples out of decision boundaries. We solve this by enforcing that the anchor-positive pairs are at least closer than the anchor-negative pairs by the margin, i.e., $dist_{ap} - dist_{an} > margin$, which has the same form with the triplet loss [22, 48]:

$$L_{\text{triplet}}(x) = max(0, dist_{\text{ap}} - dist_{\text{an}} + \text{margin}) \tag{12}$$

where $dist_{\text{ap}} = \|f(x_{\text{a}}) - f(x_{\text{hp}})\|_2$ and $dist_{\text{an}} = \|f(x_{\text{a}}) - f(x_{\text{hn}})\|_2$ measure the distance between the anchor images and the corresponding hardest positive and negative images in the latent space, respectively. And $x_{\text{hp}} = \arg\max_{x:C(x)=C(x_{\text{a}})} dist(f(x_{\text{a}}), f(x))$ are the hardest positive samples which are the least similar images that have the same label with the anchor images, $x_{hn} = \arg\max_{x:C(x)=C(x_{\text{a}})} dist(f(x_{\text{a}}), f(x))$ are the hardest negative samples which are the most similar images which have different labels from the anchor images. Therefore, the overall proposed intra-divergence loss is:

$$L_{\text{intra-div}}(x) = \alpha_{\text{ep}} L_{\text{ep}}(x) + \alpha_{\text{triplet}} L_{\text{triplet}}(x) \tag{13}$$

5. Experiments

In this section, we evaluate the effectiveness of our proposed DearKD on ImageNet to show that our two-stage learning framework for Transformers can boost the performance of Transformers. First, we provide an ablation study for the impact of each choice and analyze of data efficiency for transformers. Then, we compare with state-of-the-arts and investigate its generalization ability on downstream tasks. Finally, we analyse the results of DF-DearKD.

5.1. Implementation Details

We based our model on the DeiT [46], which is a hyperparameter-optimized version of ViT. Our models have three variants named DearKD-Ti, DearKD-S, DearKD-B, which are the same with DeiT-Ti, DeiT-S, DeiT-B, except that we increase the heads number of our three variants to 12, 12, 16 while keeping the vector dimension unchanged to increase the ability to represent convolution [10, 12]. Specifically, we first embed input images of size 224 into 16 × 16 non-overlapping patches. Then we propagate the patches through 8 MHCA and 4 MHSA blocks. Since the relative position embedding in MHCA is not suitable for the [CLS] token, which should disregard the positions of all other tokens, we simply pad the relative position embedding with zero vector and add them to all tokens. During testing or fine-tuning, we only use the [CLS] token to obtain the probability distribution. Note that our method can be easily extended to any vision transformer model.

Following [46], we use a pre-trained RegNetY-16GF from timm [49] that achieves 82.9% top-1 accuracy as our teacher model. Our models are trained from scratch using AdamW optimizer for 300 epochs with cosine learning rate decay. We optimize the model in the first stage with 250 epochs. The learning rate is 0.0005. When we train models with more epochs, we append the epochs number at the end, e.g. DearKD-Ti-1000, and train the model in the first stage with 800 epochs. A batch size of 2048 is used. The image size during training is set to 224 × 224. We use Mixup [59], Cutmix [57], Random Erasing [63] and Random Augmentation [63] for data augmentation. Experiments are conducted on 8 NVIDIA A100 GPUs.

MHCA	$L_{hiddent}$	distill	two-stage	Top1
				72.3
✓				72.5
		✓		74.3
	✓	✓		74.1
✓	✓	✓		74.6
✓	✓	✓	✓	74.8

Table 1. **Ablation of different modules** evaluated on ImageNet classification. DeiT-Ti and DearKD-Ti are used. Here, ·distill' indicates the first stage of our learning framework. The symbol ✓indicates that we use the corresponding element.

Train size	DeiT-Ti		DeiT-Ti⚗		DearKD-Ti
	Top1	Gap	Top1	Gap	
10%	40.5	13.8%	50.3	4.0%	54.3
25%	61.1	6.0%	64.3	2.8%	67.1
50%	68.3	4.0%	71.6	0.7%	72.3
100%	72.2	2.6%	74.5	0.3%	74.8

Table 2. **Comparison of data efficiency of DearKD and DeiT on ImageNet.**

5.2. Ablation Study

In this section, we ablate the important elements of our design in the proposed DearKD. We use DeiT-Ti with attention heads changed as our baseline model in the following ablation study. All the models are trained for 300 epochs on ImageNet and follow the same training setting and data augmentation strategies as described above.

As can be seen in Table 1, using our two-stage learning framework achieves the best 74.8% Top-1 accuracy among other settings. By adding our MHCA, our model reaches a Top-1 of 72.5%, outperforming the original DeiT-Ti with comparable parameters. This mild improvement is mainly because of the introduction of the locality. Note that our DearKD uses pooled content tokens as our distillation token and achieves comparable performance with DeiT-Ti⚗, which adds additional distillation tokens. Thus our model can be applied to downstream tasks without a pre-trained teacher model while the inductive biases are stored in the adaptive RPE in our MHCA. Since the differences between the feature representations of CNNs and Transformers, adding the hidden stage distillation loss decreases the model performance. Thanks to our proposed MHCA, the hidden stage distillation loss with our MHCA together brings +2.3%, illustrating their complementarity. Finally, after using a two-stage learning framework which introduces the intrinsic IBs of transformers, the performance increases to 74.8% Top-1 accuracy, demonstrating the effectiveness of learning transformers intrinsic IB.

5.3. Analysis of Data Efficiency

To validate the effectiveness of the introduced inductive biases learning framework in improving data efficiency and training efficiency, we compare our DearKD with DeiT, DeiT by training them using 10%, 25%, 50%, and 100% ImageNet training set. The results are shown in Table 2. As can be seen, DearKD consistently outperforms the DeiT baseline and DeiT⚗ by a large margin. Impressively, DearKD using only 50% training data achieves better performance with DeiT baseline using all data. When all training data are used, DearKD significantly outperforms DeiT baseline using all data by about an absolute 2.6% accuracy. It is also noteworthy that as the data volume is decreased, the gap between our DearKD and DeiT is increased, which demonstrates that our method can facilitate the training of vision transformers in the low data regime and make it possible to learn more efficiently with less training data.

5.4. Comparison with Full ImageNet

We compare our DearKD with both CNNs and vision Transformers with similar model sizes in Table 3. As we can see from the table that our DearKD achieves the best performance compared with other methods. Compared with CNNs, our DearKD-Ti achieves a 74.8% Top-1 accuracy, which is better than ResNet-18 with more parameters. The Top-1 accuracy of the DearKD-S model is 81.5%, which is comparable to RegNetY-8GF which has about two times of parameters than ours. Moreover, our DearKD-S achieves a better result than ResNet-152 with only a third of the parameters, showing the superiority of inductive biases learning procedure by design. Similar phenomena can also be observed when comparing DearKD with EffiNet, which requires a larger input size than ours.

In addition, we compare with multiple variants of vision transformers. We use the same structure with ViT and DeiT except that we increase the head number while keeping the channel dimension unchanged. Thanks to our carefully designed learning framework, DearKD can boost the performance of the model with ignorable additional parameters and computation cost. DearKD outperforms T2T-ViT, which adds an additional module on ViT to model local structure. Compared with Swin Transformer, DearKD with fewer parameters also achieves comparable or better performance. For example, DearKD-S achieves better performance with Swin-T but has 7M fewer parameters, demonstrating the superiority of the proposed MHCA and learning framework.

Generalization on downstream tasks. To showcase the generalization of the proposed method, we fine-tune the DearKD models on several fine-grained classification benchmarks. We transfer the models initialized with DearKD on full ImageNet to several benchmark tasks: CIFAR-10/100 [28], Flowers [37], Cars [27], and preprocess them follow [15,26]. The results are shown in Table 4. It can be seen that DearKD achieves SOTA performance on most of the datasets. These results demonstrate that the good generalization ability of our DearKD even without a teacher model when fine-tuning to downstream tasks.

Method	Params	size	throughput	Top1
CNNs				
ResNet-18 [19]	12M	224^2	4458.4	69.8
ResNet-50 [19]	25M	224^2	1226.1	76.2
ResNet-101 [19]	45M	224^2	753.6	77.4
ResNet-152 [19]	60M	224^2	526.4	78.3
RegNetY-4GF [39]	21M	224^2	1156.7	80.0
RegNetY-8GF [39]	39M	224^2	591.6	81.7
RegNetY-16GF [39]	84M	224^2	334.7	82.9
EffiNet-B0 [45]	5M	224^2	2694.3	77.1
EffiNet-B3 [45]	12M	300^2	732.1	81.6
EffiNet-B4 [45]	19M	380^2	349.4	82.9
EffiNet-B6 [45]	43M	528^2	96.9	84.0
EffiNet-B7 [45]	66M	600^2	55.1	84.3
Transformers				
ViT-B/16 [15]	86M	384^2	85.9	77.9
ViT-L/16 [15]	307M	384^2	27.3	76.5
T2T-ViT-7 [56]	4M	224^2	2638.4	71.7
T2T-ViT-14 [56]	22M	224^2	1443.9	81.5
T2T-ViT-19 [56]	39M	224^2	781.0	81.9
DeiT-Ti [46]	5M	224^2	2536.5	72.2
DeiT-S [46]	22M	224^2	940.4	79.8
DeiT-B [46]	86M	224^2	292.3	81.8
DeiT-Ti⚗ [46]	6M	224^2	2529.5	74.5
DeiT-S⚗ [46]	22M	224^2	936.2	81.2
DeiT-B⚗ [46]	87M	224^2	290.9	83.4
DeiT-Ti⚗-1000 [46]	6M	224^2	2529.5	76.6
DeiT-S⚗-1000 [46]	22M	224^2	936.2	82.6
DeiT-B⚗-1000 [46]	87M	224^2	290.9	84.2
Swin-T [32]	29M	224^2	755.2	81.3
Swin-S [32]	50M	224^2	436.9	83.0
Swin-B [32]	88M	224^2	278.1	83.3
Swin-B [32]	88M	384^2	84.7	84.2
DearKD-Ti	5M	224^2	1416.7	74.8
DearKD-S	22M	224^2	570.1	81.5
DearKD-B	86M	224^2	253.7	83.6
DearKD-Ti-1000	5M	224^2	1416.7	77.0
DearKD-S-1000	22M	224^2	570.1	82.8
DearKD-B-1000	86M	224^2	253.7	84.4

Table 3. **Comparison of different backbones on ImageNet classification.** Throughput is measured using the GitHub repository of [49] and a V100 GPU, following [46].

5.5. Performance of DF-DearKD

Implementation details. For the training samples generation, we use multi-resolution optimization strategy following [55]. We first downsample the input to resolution

Method	Cifar10	Cifar100	Flowers	Cars
ViT-B/32 [15]	97.8	86.3	85.4	-
ViT-B/16 [15]	98.1	87.1	89.5	-
ViT-L/32 [15]	97.9	87.1	86.4	-
ViT-L/16 [15]	97.9	86.4	89.7	-
T2T-ViT-14 [56]	98.3	88.4	-	-
EffiNet-B5 [45]	98.1	91.1	98.5	-
DeiT-B [46]	99.1	90.8	98.4	92.1
DeiT-B⚗ [46]	99.1	91.3	98.8	92.9
DearKD-Ti	97.5	85.7	95.1	89.0
DearKD-S	98.4	89.3	97.4	91.3
DearKD-B	99.2	91.1	98.8	92.7

Table 4. **Generalization of DearKD and SOTA methods on different downstream tasks.**

112×112 and optimize for $2k$ iterations. Then, we optimize the input of resolution 224×224 for $2k$ iterations. We use Adam optimizer and cosine learning scheduler. Learning rates for each step are 0.5 and 0.01, respectively. We set $\alpha_{\text{TV}} = 1e-4, \alpha_{l_2} = 1e-5, \alpha_{\text{BN}} = 5e-2, \alpha_{\text{ep}} = 50, \alpha_{\text{triplet}} = 0.5$. We set batch size to 42 and generate 6 classes each batch randomly. Image pixels are randomly initialized i.i.d. from Gaussian noise of $\mu = 0$ and $\sigma = 1$. We use RegNetY-16GF [39] from timm [49] pre-trained on ImageNet [13]. Experiments are conducted on NVIDIA TITAN X GPUs.

Performance comparison. Table 5 shows the performance of the student model obtained with different methods. As shown in the table, our method performs significantly better than training with other data-free methods. Although our methods achieves results lower than distillation on real images with the same number, the results are close to training from scratch with original ImageNet dataset. For example, the student model trained with our method gets only 1.0% decrease on DeiT-Ti compared with training from scratch.

Furthermore, the ablation experiments can be seen on the last three rows in Table 5. The third-to-last row denotes distillation with images generated from DeepInversion without diverse loss achieves accuracy of only 62.7%. When further training with the diversity loss of ADI, we observe 7.4% accuracy improvement. And by applying the our intra-divergence loss brings in 8.6% increase.

Diversity comparison. We demonstrate the diversity by comparing the LPIPS [29,61] of our generated images with other methods in Table 6. We compute the distance between 4000 pairs of images. We randomly sample 4 pairs of images for each class. The highest score compared with other methods shows that our method can generate diverse images. Although there is still a gap between our generated images and real images, the generated samples can be a data source to train the high-performance model.

Teacher Network	ResNet-101	ResNet-101
Teacher Accuracy	77.37%	77.37%
Student Network	DeiT-Ti	DeiT-S
Train from scratch		
ImageNet	72.2%	79.8%
Distill on real images		
ImageNet	74.6% (2.4% ↑)	81.5% (1.7% ↑)
partial ImageNet	72.2% (0.0% ↓)	79.1% (0.7% ↓)
Distill on generated samples		
DeepInversion	62.7% (9.5% ↓)	66.3 (13.5% ↓)
ADI	70.1% (2.1% ↓)	73.1 (6.7% ↓)
DF-DearKD	71.2% (1.0% ↓)	74.0 (5.8% ↓)

Table 5. **Knowledge distillation results from a pre-trained ResNet-101 classifier to a ViT initialized from scratch on the ImageNet dataset.** ↑ and ↓ indicate performance increase and decrease, respectively.

Method	LPIPS
real images	0.710
DeepInversion	0.668
ADI	0.687
DF-DearKD	0.693

Table 6. **Diversity quantitative comparison.** We use the LPIPS metric to measure the diversity of the generated images. Higher LPIPS score indicates better diversity among the generated images.

6. Conclusion

In this paper, we propose DearKD, an early knowledge distillation framework, to improve the data efficiency for training transformers. In the first stage, inductive biases are distilled from the early intermediate layers of a CNN to the transformer, while the second stage allows the transformer to make full use of its capacity by training without distillation. Moreover, we enhance the performance of DearKD under the extreme data-free case by introducing a boundary-preserving intra-divergence loss to generate diverse training samples. We conduct experiments on ImageNet, partial ImageNet, data-free setting and downstream tasks, and demonstrate that DearKD achieves superior performance.

Acknowledgements. This work is supported by the Major Science and Technology Innovation 2030 ”New Generation Artificial Intelligence” key project (No. 2021ZD0111700), and National Key R&D Program of China (2018AAA0100704), NSFC 61932020, 62172279, Science and Technology Commission of Shanghai Municipality (Grant No.20ZR1436000), and ”Shuguang Program” supported by Shanghai Education Development Foundation and Shanghai Municipal Education Commission. Dr. Jing Zhang is supported by ARC FL-170100117.

References

[1] Samira Abnar, Mostafa Dehghani, and Willem Zuidema. Transferring inductive biases through knowledge distillation. *arXiv preprint arXiv:2006.00555*, 2020. 1

[2] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016. 4

[3] Kartikeya Bhardwaj, Naveen Suda, and Radu Marculescu. Dream distillation: A data-independent model compression framework. *arXiv preprint arXiv:1905.07072*, 2019. 3

[4] Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020. 1

[5] Akshay Chawla, Hongxu Yin, Pavlo Molchanov, and Jose Alvarez. Data-free knowledge distillation for object detection. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3289–3298, 2021. 3

[6] Tong Che, Yanran Li, Athul Paul Jacob, Yoshua Bengio, and Wenjie Li. Mode regularized generative adversarial networks. *arXiv preprint arXiv:1612.02136*, 2016. 3, 5

[7] Hanting Chen, Yunhe Wang, Tianyu Guo, Chang Xu, Yiping Deng, Zhenhua Liu, Siwei Ma, Chunjing Xu, Chao Xu, and Wen Gao. Pre-trained image processing transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12299–12310, 2021. 1, 2

[8] Hanting Chen, Yunhe Wang, Chang Xu, Zhaohui Yang, Chuanjian Liu, Boxin Shi, Chunjing Xu, Chao Xu, and Qi Tian. Data-free learning of student networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3514–3522, 2019. 2, 3

[9] Xianing Chen, Jialang Xu, Jiale Xu, and Shenghua Gao. Ohformer: Omni-relational high-order transformer for person re-identification. *arXiv preprint arXiv:2109.11159*, 2021. 2

[10] Jean Baptiste Cordonnier, Andreas Loukas, and Martin Jaggi. On the relationship between self-attention and convolution. In *International Conference on Learning Representations (ICLR)*, 2020. 2, 3, 6

[11] Zihang Dai, Hanxiao Liu, Quoc V Le, and Mingxing Tan. Coatnet: Marrying convolution and attention for all data sizes. *arXiv preprint arXiv:2106.04803*, 2021. 1, 2, 3

[12] Stéphane d'Ascoli, Hugo Touvron, Matthew Leavitt, Ari Morcos, Giulio Biroli, and Levent Sagun. Convit: Improving vision transformers with soft convolutional inductive biases. *arXiv preprint arXiv:2103.10697*, 2021. 1, 2, 6

[13] 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. Ieee, 2009. 2, 8

[14] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 1, 2

[15] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 1, 2, 7, 8

[16] Diana F Gordon and Marie Desjardins. Evaluation and selection of biases in machine learning. *Machine learning*, 20(1):5–22, 1995. 1

[17] Anirudh Goyal and Yoshua Bengio. Inductive biases for deep learning of higher-level cognition. *arXiv preprint arXiv:2011.15091*, 2020. 1

[18] Ben Graham, Alaaeldin El-Nouby, Hugo Touvron, Pierre Stock, Armand Joulin, Hervé Jégou, and Matthijs Douze. Levit: a vision transformer in convnet's clothing for faster inference. *arXiv preprint arXiv:2104.01136*, 2021. 1, 2

[19] 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. 7

[20] Byeongho Heo, Minsik Lee, Sangdoo Yun, and Jin Young Choi. Knowledge transfer via distillation of activation boundaries formed by hidden neurons. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 3779–3787, 2019. 2

[21] Byeongho Heo, Sangdoo Yun, Dongyoon Han, Sanghyuk Chun, Junsuk Choe, and Seong Joon Oh. Rethinking spatial dimensions of vision transformers. *arXiv preprint arXiv:2103.16302*, 2021. 4

[22] Alexander Hermans, Lucas Beyer, and Bastian Leibe. In defense of the triplet loss for person re-identification. *arXiv preprint arXiv:1703.07737*, 2017. 5

[23] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. 1, 2

[24] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. PMLR, 2015. 5

[25] Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. Tinybert: Distilling bert for natural language understanding. *arXiv preprint arXiv:1909.10351*, 2019. 2

[26] Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Joan Puigcerver, Jessica Yung, Sylvain Gelly, and Neil Houlsby. Big transfer (bit): General visual representation learning. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part V 16*, pages 491–507. Springer, 2020. 7

[27] J. Krause, M. Stark, J. Deng, and F. F. Li. 3d object representations for fine-grained categorization. In *IEEE International Conference on Computer Vision Workshops*, 2014. 7

[28] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 7

[29] Hsin-Ying Lee, Hung-Yu Tseng, Jia-Bin Huang, Maneesh Singh, and Ming-Hsuan Yang. Diverse image-to-image translation via disentangled representations. In *Proceedings of the European conference on computer vision (ECCV)*, pages 35–51, 2018. 8

[30] Yehao Li, Ting Yao, Yingwei Pan, and Tao Mei. Contextual transformer networks for visual recognition. *arXiv preprint arXiv:2107.12292*, 2021. 1, 2

[31] Yifan Liu, Ke Chen, Chris Liu, Zengchang Qin, Zhenbo Luo, and Jingdong Wang. Structured knowledge distillation for semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2604–2613, 2019. 2

[32] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. *arXiv preprint arXiv:2103.14030*, 2021. 1, 7

[33] Raphael Gontijo Lopes, Stefano Fenu, and Thad Starner. Data-free knowledge distillation for deep neural networks. *arXiv preprint arXiv:1710.07535*, 2017. 3

[34] Paul Micaelli and Amos Storkey. Zero-shot knowledge transfer via adversarial belief matching. *arXiv preprint arXiv:1905.09768*, 2019. 2, 3

[35] Tom M Mitchell. *The need for biases in learning generalizations*. Department of Computer Science, Laboratory for Computer Science Research . . . , 1980. 1

[36] A. Mordvintsev, Christopher Olah, and Mike Tyka. Inceptionism: Going deeper into neural networks. 2015. 3, 5

[37] M. E. Nilsback and A. Zisserman. Automated flower classification over a large number of classes. In *Sixth Indian Conference on Computer Vision, Graphics Image Processing, ICVGIP 2008, Bhubaneswar, India, 16-19 December 2008*, 2008. 7

[38] Zizheng Pan, Bohan Zhuang, Jing Liu, Haoyu He, and Jianfei Cai. Scalable vision transformers with hierarchical pooling. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 377–386, 2021. 4

[39] Ilija Radosavovic, Raj Prateek Kosaraju, Ross Girshick, Kaiming He, and Piotr Dollár. Designing network design spaces. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10428–10436, 2020. 7, 8

[40] Prajit Ramachandran, Niki Parmar, Ashish Vaswani, Irwan Bello, Anselm Levskaya, and Jonathon Shlens. Stand-alone self-attention in vision models. *arXiv preprint arXiv:1906.05909*, 2019. 2, 3

[41] Daniel L Ruderman and William Bialek. Statistics of natural images: Scaling in the woods. *Physical review letters*, 73(6):814, 1994. 1

[42] Eero P Simoncelli and Bruno A Olshausen. Natural image statistics and neural representation. *Annual review of neuroscience*, 24(1):1193–1216, 2001. 1

[43] Suraj Srinivas and François Fleuret. Knowledge transfer with jacobian matching. In *International Conference on Machine Learning*, pages 4723–4731. PMLR, 2018. 2

[44] Akash Srivastava, Lazar Valkov, Chris Russell, Michael U Gutmann, and Charles Sutton. Veegan: Reducing mode collapse in gans using implicit variational learning. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 3310–3320, 2017. 3, 5

[45] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*, pages 6105–6114. PMLR, 2019. 4, 7, 8

[46] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*, pages 10347–10357. PMLR, 2021. 1, 2, 4, 6, 7, 8

[47] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017. 1, 2, 3

[48] Kilian Q Weinberger and Lawrence K Saul. Distance metric learning for large margin nearest neighbor classification. *Journal of machine learning research*, 10(2), 2009. 5

[49] Ross Wightman. Pytorch image models. https://github.com/rwightman/pytorch-image-models, 2019. 6, 7, 8

[50] Haiping Wu, Bin Xiao, Noel Codella, Mengchen Liu, Xiyang Dai, Lu Yuan, and Lei Zhang. Cvt: Introducing convolutions to vision transformers. *arXiv preprint arXiv:2103.15808*, 2021. 1, 2

[51] Tete Xiao, Mannat Singh, Eric Mintun, Trevor Darrell, Piotr Dollár, and Ross Girshick. Early convolutions help transformers see better. *arXiv preprint arXiv:2106.14881*, 2021. 1, 2, 3

[52] Yufei Xu, Qiming Zhang, Jing Zhang, and Dacheng Tao. Vitae: Vision transformer advanced by exploring intrinsic inductive bias. *arXiv preprint arXiv:2106.03348*, 2021. 1, 2

[53] Hong Xuan, Abby Stylianou, and Robert Pless. Improved embeddings with easy positive triplet mining. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2474–2482, 2020. 5

[54] Jingwen Ye, Yixin Ji, Xinchao Wang, Xin Gao, and Mingli Song. Data-free knowledge amalgamation via group-stack dual-gan. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12516–12525, 2020. 3

[55] Hongxu Yin, Pavlo Molchanov, Jose M Alvarez, Zhizhong Li, Arun Mallya, Derek Hoiem, Niraj K Jha, and Jan Kautz. Dreaming to distill: Data-free knowledge transfer via deepinversion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8715–8724, 2020. 2, 3, 5, 7

[56] L. Yuan, Y Chen, T. Wang, W. Yu, Y Shi, F. E. Tay, J. Feng, and S. Yan. Tokens-to-token vit: Training vision transformers from scratch on imagenet. 2021. 7, 8

[57] S. Yun, D. Han, S. Chun, S. J. Oh, Y. Yoo, and J. Choe. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *International Conference on Computer Vision*. 6

[58] Sergey Zagoruyko and Nikos Komodakis. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. *arXiv preprint arXiv:1612.03928*, 2016. 2

[59] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz. mixup: Beyond empirical risk minimization. 2017. 6

[60] Qiming Zhang, Yufei Xu, Jing Zhang, and Dacheng Tao. Vitaev2: Vision transformer advanced by exploring inductive bias for image recognition and beyond. *arXiv preprint arXiv:2202.10108*, 2022. 2

[61] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018. 8

[62] Yiman Zhang, Hanting Chen, Xinghao Chen, Yiping Deng, Chunjing Xu, and Yunhe Wang. Data-free knowledge distillation for image super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7852–7861, 2021. 3

[63] Z. Zhong, L. Zheng, G. Kang, S. Li, and Y. Yang. Random erasing data augmentation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(7), 2017. 6