

VITKD: PRACTICAL GUIDELINES FOR ViT FEATURE KNOWLEDGE DISTILLATION

Zhendong Yang^{1,2} Zhe Li Ailing Zeng² Zexian Li³ Chun Yuan^{†1} Yu Li^{†2}

¹Tsinghua Shenzhen International Graduate School

²International Digital Economy Academy (IDEA) ³Beihang University

{yangzd21@mails, yuanc@sz}.tsinghua.edu.cn axel.li@outlook.com
{zengailing, liyu}@idea.edu.cn lizexian0427@gmail.com

ABSTRACT

Knowledge Distillation (KD) for Convolutional Neural Network (CNN) is extensively studied as a way to boost the performance of a small model. Recently, Vision Transformer (ViT) has achieved great success on many computer vision tasks and KD for ViT is also desired. However, besides the output logit-based KD, other feature-based KD methods for CNNs cannot be directly applied to ViT due to the huge structure gap. In this paper, we explore the way of feature-based distillation for ViT. Based on the nature of feature maps in ViT, we design a series of controlled experiments and derive three practical guidelines for ViT’s feature distillation. Some of our findings are even opposite to the practices in the CNN era. Based on the three guidelines, we propose our feature-based method ViTKD which brings consistent and considerable improvement to the student. On ImageNet-1k, we boost DeiT-Tiny from 74.42% to 76.06%, DeiT-Small from 80.55% to 81.95%, and DeiT-Base from 81.76% to 83.46%. Moreover, ViTKD and the logit-based KD method are complementary and can be applied together directly. This combination can further improve the performance of the student. Specifically, the student DeiT-Tiny, Small, and Base achieve 77.78%, 83.59%, and 85.41%, respectively.*

1 INTRODUCTION

Knowledge Distillation (KD) (Hinton et al., 2015) utilizes the output of the teacher model as soft labels to supervise the student model, bringing the lightweight models impressive improvements without extra costs for inference. It has been consistently explored for Convolutional Neural Network (CNN) models and applied successfully to many vision tasks successfully, including image classification (Zhou et al., 2020; Yang et al., 2020; Chen et al., 2021; Zhao et al., 2022; Lin et al., 2022), object detection (Li et al., 2022a; Yang et al., 2022d; Zheng et al., 2022; Yang et al., 2022a; Wang et al., 2022), semantic segmentation (Liu et al., 2019; He et al., 2019; Shu et al., 2021; Yang et al., 2022b).

Recently, Vision Transformer (ViT) (Dosovitskiy et al., 2021) has achieved great success for image classification and inspired various transformers (Yuan et al., 2021; Han et al., 2021; Touvron et al., 2021b; Liu et al., 2021). Compared with CNN-based models, the ViT-based methods generally need more parameters but can achieve better performance, making them harder to be deployed. Therefore, boosting the performance of small ViT models using KD is of great value. In this work, we look into *how to apply KD to ViT-based models*. A direct thought would be directly transferring the KD methods for CNN to ViT. In fact, some fundamental distillation works (Hinton et al., 2015; Romero et al., 2014) are inherently structure-independent. For example, the classic logit-based distillation directly use the model’s final output logit, thus it can apply for both CNNs and ViTs. DeiT (Touvron et al., 2021a) verifies this for ViT’s distillation.

However, the rest KD methods are mostly specially designed for CNN-based models and many of them work on the intermediate features. They are inapplicable to ViT-based models as there is a huge gap between these two architectures. The recent MiniViT (Zhang et al., 2022) adopts Self-Attention

[†]Corresponding authors

*The code is available at https://github.com/yzd-v/cls_KD.

Table 1: The results of different feature distillation methods, including distillation on the last layer like CNN’s distillation, the last 6 layers like BERT’s distillation, and all the 12 layers.

Distillation setting	DeiT Small (Teacher) - DeiT Tiny (Student)			
Methods	Baseline	Last layer	Last 6 layers	All 12 layers
Top-1 Accuracy	74.42	73.36 (-1.06)	73.76 (-0.66)	74.24 (-0.18)
Top-5 Accuracy	92.29	91.88	92.01	92.23

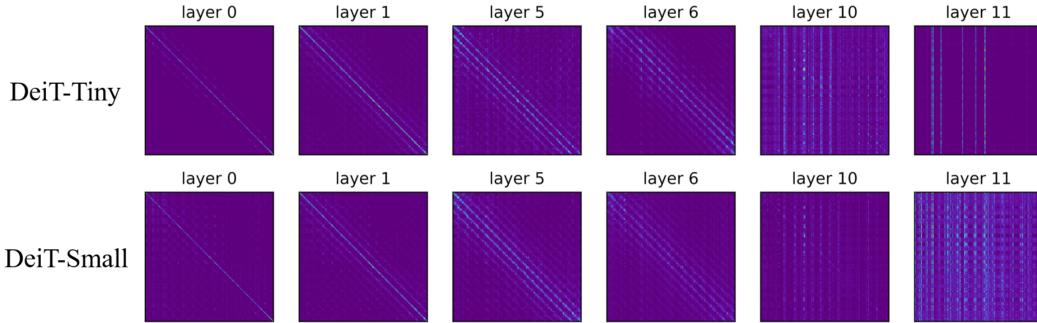


Figure 1: DeiT-Tiny’s and DeiT-Small’s attention maps from shallow to deep layers. The X-axis and Y-axis mean the key and query tokens, respectively. The attention map is obtained by *softmax* and reflects the response between the query and key tokens. The color is brighter with a larger response.

distillation and Hidden-State Distillation for feature-based distillation. Compared with the logit based distillation, its improvement is still quite limited.

Before developing new feature-based KD for ViT, we first conduct simple studies of transferring the knowledge from the last layer of a teacher (DeiT-Small) following the CNN’s distillation, from the last 6 layers like PKD (Sun et al., 2019) for BERT’s (Devlin et al., 2018) distillation, and from the whole 12 layers. Surprisingly, the results for all the intuitive feature distillations shown in Table 1 are not satisfactory which consistently degrade the performance of the student (DeiT-Tiny). Specifically, the Top-1 accuracy of the student is just 73.36% when distilling on the last layer. This distillation on the last layer is widely used for CNN’s distillation, but here it causes a 1.06% accuracy drop.

To further explore the features in ViT, we visualize the attention maps of the student and the teacher across different layers in Figure 1. For the shallow layers (*e.g.*, layers 0 and 1), the attention appears mainly on the diagonal, which indicates they focus on themselves. Both the student and teacher have similar patterns. While for the deep layers (*e.g.*, layers 10 and 11), the difference between student and teacher’s attention is greater. Their attention is decided by several sparse key tokens. Besides, they focus on completely different tokens. Such gap makes it hard for the student to mimic the teacher’s final feature directly. This phenomenon suggests different layers may need different methods.

Accordingly, we perform a series of controlled experiments to examine the effects of different distillation methods, different layers, and different modules. As a consequence, we derive three practical guidelines for ViT’s feature distillation in Section 2. Based on these principles, we propose a nontrivial way for feature-based ViT distillation, named **ViTKD**, and describe the details in Section 3. Extensive experiments demonstrate its effective-

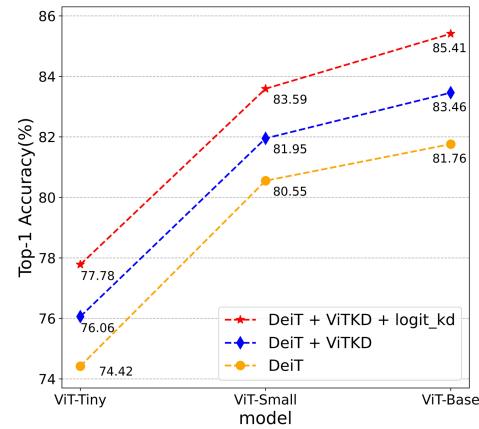


Figure 2: Comparison of training vision transformers with distillation on ImageNet-1k.

Table 2: The comparisons of Top-1 accuracy with different distillation methods on DeiT’s deep-layer feature on ImageNet-1K.

Type	Teacher Student	DeiT-Small (80.69) DeiT-Tiny (74.42)	DeiT III-Small (82.76) DeiT-Tiny (74.42)
<i>Mimicking</i>	Linear layer	73.36 (-1.06)	73.72 (-0.70)
	Correlation matrix	72.37 (-2.05)	72.20 (-2.22)
<i>Generation</i>	Cross-attention	73.77 (-0.65)	73.98 (-0.44)
	Self-attention	74.61 (+0.19)	74.65 (+0.23)
	Conv. projector	74.72 (+0.30)	74.79 (+0.37)

ness in Section 4. For instance, we boost the student DeiT-Tiny from 74.42% to 76.06%, DeiT-Small from 80.55% to 81.95% and DeiT-Base from 81.76% to 83.46% on ImageNet-1K. Besides, when combining ViTKD with the logit-based distillation, we can further advance their Top-1 accuracy to 77.78%, 83.59% and 85.41%. The comparison is shown in Figure 2. We also demonstrate the models trained with distillation are beneficial to other vision tasks like object detection.

2 PRACTICAL GUIDELINES FOR ViT’S FEATURE DISTILLATION

To explore the practical guidelines, we take larger DeiT (Touvron et al., 2021a) and DeiT III (Touvron et al., 2022) models as the teacher to distill lighter DeiT models on ImageNet-1k (Deng et al., 2009). The DeiT teacher is trained from scratch on ImageNet-1K, and DeiT III teacher is pre-trained on ImageNet-21K. As shown in Section 1, the attention maps vary greatly from different layers. Based on this observation, we analyze where and how to distill the student effectively and propose three practical guidelines for ViT’s feature distillation. Specifically, we conduct distillation experiments on features in different layers of DeiT with two strategies, namely *mimicking* and *generation*. When using *mimicking*, we align the embedding dimensions of the student and the teacher by a linear layer and correlation matrix, respectively. As for *generation*, we randomly mask the student’s tokens and utilize a generative block to restore the feature. Furthermore, we choose different generative blocks, including cross-attention block (Chen et al., 2022), self-attention block (He et al., 2022), and convolutional projector (Yang et al., 2022e). For both *mimicking* and *generation*, we calculate the square of L_2 distance as the distillation loss. The details about *mimicking* and *generation* are elaborated in Subsection 3.1 and 3.2, respectively.

G1) For distillation on the deep layer, *generation* is more suitable than *mimicking*. For CNN’s feature distillation, many works (Park et al., 2019; Tian et al., 2019; Yang et al., 2022e) transfer teachers’ semantic information from the last-stage feature. Most feature-based distillation methods (Romero et al., 2014; Zagoruyko & Komodakis, 2016; Heo et al., 2019) aim at making students get similar feature maps to the teacher. While MGD (Yang et al., 2022e) forces the student to generate the teacher’s full feature instead of mimicking it directly.

As our results shown in Table 2, the way to mimic the last layer feature of the teacher surprisingly impair the student’s performance noticeably. Specifically, the student’s Top-1 accuracy drops about 2% when mimicking the teacher’s correlation matrix. This trend is completely different from distillation for CNN-based models. Instead, the generation methods can improve the accuracy of the student mostly. The largest gains are obtained by using the convolutional projector as the generative block. These results reveal that *generation* is more suitable than *mimicking* for the deep layer.

G2) Distillation on the shallow layers also works for ViT with *mimicking*. For the CNN-based model’s feature distillation, the feature of shallow layers has a small receptive field and lacks semantic information, making it unsuitable for distillation. As the attention map shown in Figure 1, the shallow feature of DeiT also has a small receptive. That is, the tokens in the first two layers just have responses to themselves. Still, we believe such incipient attention knowledge is useful for distillation because it can teach the student how to form a better attention map at the beginning.

We pick the first two layers for distillation by either mimicking or generation in Table 3. Interestingly, the conclusion for feature distillation on the shallow layers and deep layers is the opposite. The

Table 3: The comparisons of Top-1 accuracy with different distillation methods on DeiT’s shallow-layer feature on ImageNet-1k.

Type	Teacher Student	DeiT-S (80.69) DeiT-T (74.42)	DeiT III-S (82.76) DeiT-T (74.42)	DeiT-B (81.76) DeiT-T (74.42)
<i>Mimicking</i>	Linear layer	75.12 (+0.70)	75.31 (+0.89)	75.15 (+0.73)
	Correlation matrix	75.27 (+0.85)	74.94 (+0.52)	75.01 (+0.59)
<i>Generation</i>	Conv. projector	74.69 (+0.27)	74.86 (+0.44)	74.71 (+0.29)

Table 4: The comparisons of Top-1 accuracy with different distillation modules on DeiT’s shallow layer feature on ImageNet-1k.

Type	Teacher Student	DeiT-Small (80.69) DeiT-Tiny (74.42)	DeiT III-Small (82.76) DeiT-Tiny (74.42)
Modules	MHA	75.06 (+0.64)	75.02 (+0.60)
	FFN	75.12 (+0.70)	75.31 (+0.89)

relations of different tokens and semantic information from the shallow layers are so weak that the student can not utilize its masked feature to generate the full feature from the teacher. It makes the generation way just bring a little improvement for the shallow-layer distillation. Moreover, different from distillation for CNN-based models, transferring the knowledge from teacher’s shallow layer by directly mimicking makes great progress. Mimicking by ‘correlation matrix’ performs a little better than the ‘linear layer’ way when using DeiT-S as the teacher. When the teacher performs better, the ‘linear layer’ way benefits the student much more than the ‘correlation matrix’ way. As we described above, the results validate that the shallow layer matters for distillation by mimicking. We fix to use the ‘linear layer’ strategy to distill the shallow layers.

G3) The FFN-out features are better than the MHA-out features for distillation. The ViT-based models are built by stacking several encoder layers. Each encoder layer consists of a multi-head attention (MHA) module and a feed-forward network (FFN) module. Based on the findings from **G1** and **G2**, we further conduct experiments on the first two layers of the student to explore how to choose the modules for ViT’s feature distillation.

We use the ‘linear layer’ way for the shallow-layer distillation on the MHA-out feature and FFN-out feature, respectively. Table 4 demonstrates that distilling on the MHA-out feature or FFN-out both bring the student improvements and transferring the knowledge from the FFN-out feature is better than the MHA-out feature.

3 METHODOLOGY

As mentioned in our guidelines in Section 2, we apply ‘linear layer’ and ‘correlation matrix’ for *mimicking*. For *generation*, we use ‘cross-attention’, ‘self-attention’, and ‘conv. projector’. In this section, we describe the details of these methods and the final formulation of our ViTKD. To begin, we recall the feature distillation of CNN is based on the L_2 distances between the feature maps. We also follow this paradigm. The general form of CNN’s feature distillation loss is as following:

$$\mathcal{L}_{fea} = \sum_{k=1}^C \sum_{i=1}^H \sum_{j=1}^W (\mathcal{F}_{k,i,j}^T - f(\mathcal{F}_{k,i,j}^S))^2, \quad (1)$$

where \mathcal{F}^T and \mathcal{F}^S denote the feature from the teacher and student, respectively, and $f(\cdot)$ is the adaptation layer to reshape the \mathcal{F}^S to the same dimension as \mathcal{F}^T . H and W denote the height and width of the feature, and C is the channel length. In the next, we shift to feature distillation for ViT.

3.1 MIMICKING FOR SHALLOW LAYERS

For each sample, we can denote student’s and teacher’s feature as $\mathcal{F}^S \in \mathcal{R}^{N \times D_S}$ and $\mathcal{F}^T \in \mathcal{R}^{N \times D_T}$, respectively. For the mimicking method, we utilize a linear layer to align the dimension of the student’s D_S and the teacher’s D_T . We term the strategy as ‘linear layer’ and summarize it as:

$$\mathcal{L}_{lr} = \sum_{i=1}^N \sum_{j=1}^D (\mathcal{F}_{i,j}^T - fc(\mathcal{F}^S)_{i,j})^2, \quad (2)$$

where $fc(\cdot)$ is a linear layer to reshape the \mathcal{F}^S to the same dimension as \mathcal{F}^T . N, D denote the number of patch tokens and the embedding dimension of the teacher’s feature.

Besides, we use a correlation matrix to describe the response among different patch tokens and force the student to learn the correlation matrix of the teacher’s features. In this case, we do not need the adaption layer to align the embedding dimension. The correlation matrix for each sample can be calculated as:

$$\mathcal{M} = \frac{\mathcal{F}\mathcal{F}^{Tr}}{\sqrt{D}}, \quad (3)$$

where $\mathcal{F} \in \mathcal{R}^{N \times D}$ denotes the student or teacher’s feature. D is their embedding dimension and Tr denotes transposition for the feature, so $\mathcal{F}^{Tr} \in \mathcal{R}^{D \times N}$. In this case, student’s and teacher’s relation matrices have the same shape $\mathcal{M} \in \mathcal{R}^{N \times N}$ and describe the response between different patch tokens. With ‘correlation matrix’, we calculate the distillation loss as:

$$\mathcal{L}_{rm} = \sum_{i=1}^N \sum_{j=1}^N (\mathcal{M}_{i,j}^T - \mathcal{M}_{i,j}^S)^2. \quad (4)$$

3.2 GENERATION FOR DEEP LAYERS

For *generation*, we first use a linear layer to align the feature dimension of the student and teacher. Then, we set a random mask $Mask \in \mathcal{R}^{N \times 1}$ and use masked tokens to replace the student’s original tokens, which can be formulated as:

$$\hat{\mathcal{F}}_i^S = \begin{cases} \text{masked token}, & \text{if } r_i < \lambda \\ \text{original token}, & \text{Otherwise,} \end{cases} \quad (5)$$

$$Mask_i = \begin{cases} 1, & \text{if } r_i < \lambda \\ 0, & \text{Otherwise,} \end{cases} \quad (6)$$

where r_i is a random number uniformly distributed in $[0, 1]$ and $i \in [0, N - 1]$ is the coordinates of the tokens dimension. λ is a hyper-parameter that is set as 0.5 for all the experiments. The *masked token* is the parameter to learn during training.

Finally, we use the new masked feature $\hat{\mathcal{F}}_i^S$ to generate teacher’s the full feature through a generative block \mathcal{G} , which can be formulated as follows:

$$\mathcal{G}(\hat{\mathcal{F}}^S) \longrightarrow \mathcal{F}^T. \quad (7)$$

We choose three ways to set the generative block \mathcal{G} . The first way is a ‘cross-attention’ block from CAE (Chen et al., 2022), which includes 6 transformer layers. The second way is a ‘self-attention’ block from MAE (He et al., 2022), which also includes 6 transformer layers. The difference between them is that cross-attention uses masked tokens as the query tokens. The third way is a ‘convolutional projector’ from MGD (Yang et al., 2022e), which includes two conventional layers. For all the three ways, we only calculate the distillation loss of masked tokens.

For *generation* method, we design the distillation loss \mathcal{L}_{gen} as:

$$\mathcal{L}_{gen} = \sum_{i=1}^N \sum_{j=1}^D Mask_i (\mathcal{F}_{i,j}^T - \mathcal{G}(\hat{\mathcal{F}}_{i,j}^S))^2. \quad (8)$$

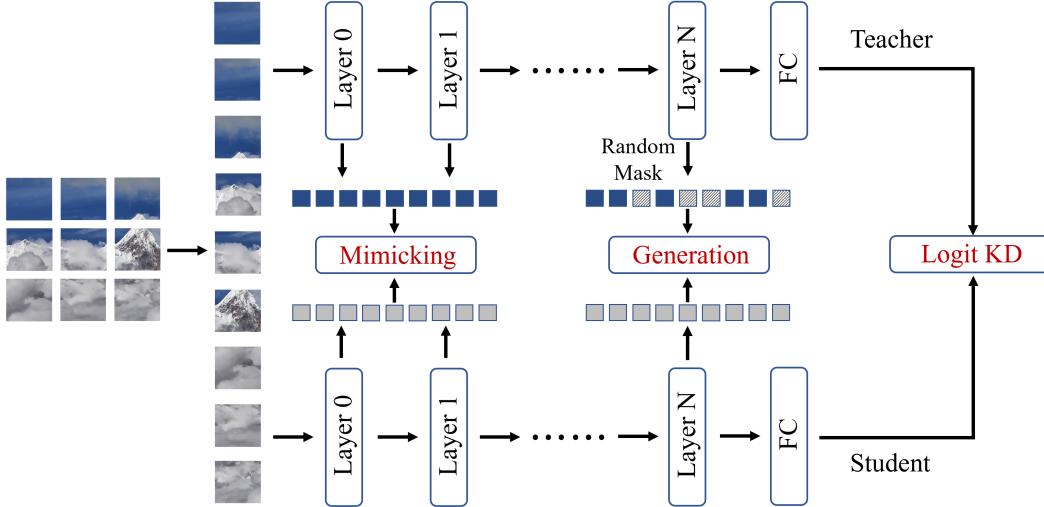


Figure 3: Illustration of the proposed ViTKD. ViTKD is a feature-based distillation method that includes *Mimicking* and *Generation*. It can be directly combined with the output logit-based distillation method together.

3.3 ViTKD

Based on the findings from **G1**, **G2** and **G3**, we finally propose our method ViTKD. We first use the ‘**linear layer**’ approach for the first two layers’ distillation, where the distillation loss is \mathcal{L}_{lr} . As for the last layer, we apply the ‘**conv. projector**’ for the generation distillation, where the distillation loss is \mathcal{L}_{gen} . The ViTKD we propose is shown in Figure 3. To sum up, we train the student model with the total loss as follows:

$$\mathcal{L} = \mathcal{L}_{ori} + \alpha \mathcal{L}_{lr} + \beta \mathcal{L}_{gen}, \quad (9)$$

where \mathcal{L}_{ori} is the original loss for the models, *e.g.*, the cross-entropy loss in DeiT-Tiny. α and β are two hyper-parameters to balance the loss.

4 EXPERIMENT

4.1 SETTINGS

Datasets. We explore the feature distillation for ViT-based models on ImageNet-1k (Deng et al., 2009), which contains 1000 object categories. We use the 1.2 million images to train the model and 50k images to evaluate the performance. For the downstream task, we evaluate our model on the COCO dataset (Lin et al., 2014), which contains 80 object classes. We use the 120k train images for training and 5k validation images for testing.

Implementation details. ViTKD uses the hyper-parameters α and β to balance the distillation loss in Equation 9. Another hyper-parameter λ is used to adjust the masked ratio for deep layer distillation in Equation 5. We adopt the hyper-parameters $\{\alpha = 3 \times 10^{-5}, \beta = 3 \times 10^{-6}, \lambda = 0.5\}$ for all the experiments. As for the logit distillation, we apply the distillation method NKD (Yang et al., 2022c) and set the hyper-parameters $\{\alpha = 1, \text{temperature} = 1\}$. Besides, to keep the model to be the same for the feature and logit distillation, we remove the extra distillation token which is used for logit distillation in DeiT. The image resolution for all the experiments is 224×224 . The other training details for distillation follow the setting from the baseline training setting in MMClassification (Contributors, 2020). All the experiments are conducted on 8 GPUs with MMClassification in Pytorch (Paszke et al., 2019). Unless specified, we evaluate the model with the performance of the last epoch.

Table 5: Main results on ImageNet-1k. * indicates the teacher is pre-trained on ImageNet-21K. We evaluate DeiT-B with the best performance because of the undulation when training the model.

Teacher	Student	Type	Top-1 Accuracy	Top-5 Accuracy
DeiT-Small (80.69)	DeiT-Tiny	-	74.42	92.29
	KD	logit	75.01 (+0.59)	92.52
	NKD	logit	75.48 (+1.06)	92.72
	Ours	feature	75.40 (+0.98)	92.66
	Ours+NKD	feature+logit	76.18 (+1.76)	93.14
DeiT III-Small* (82.76)	DeiT-Tiny	-	74.42	92.29
	KD	logit	76.01 (+1.59)	93.26
	NKD	logit	76.68 (+2.26)	93.51
	Ours	feature	76.06 (+1.64)	93.16
	Ours+NKD	feature+logit	77.78 (+3.36)	93.97
DeiT III-Base* (85.48)	DeiT-Small	-	80.55	95.12
	KD	logit	82.52 (+1.97)	96.30
	NKD	logit	82.74 (+2.19)	96.33
	Ours	feature	81.95 (+1.40)	95.64
	Ours+NKD	feature+logit	83.59 (+3.04)	96.69
DeiT III-Large* (86.81)	DeiT-Base	-	81.76	95.81
	KD	logit	84.06 (+2.30)	96.77
	NKD	logit	84.96 (+3.20)	97.17
	Ours	feature	83.46 (+1.70)	96.41
	Ours+NKD	feature+logit	85.41 (+3.65)	97.39

4.2 MAIN RESULTS

To evaluate our methods for ViT-based models, we utilize different teachers to distill different students. We train the student with the proposed ViTKD, the classic KD, the state-of-the-art logit method NKD (Yang et al., 2022c). We also combine ViTKD and NKD to explore the upper bound of the student’s performance. In Table 5, all the teachers bring the students remarkable performance improvements, *e.g.*, the DeiT III-Small teacher boosts the student’s Top-1 accuracy from 74.42% to 76.06% with our ViTKD method. The results of ViTKD even surpass the classic logit-based KD method. Comparing the results between different teachers, we find the student achieves better performance with a stronger teacher, *e.g.*, the student DeiT-Tiny achieves 75.40% and 76.06% Top-1 accuracy with the DeiT-Small and DeiT III-Small teacher, respectively. Furthermore, we also apply our method to a stronger student DeiT-Small and DeiT-Base. ViTKD can also bring them significant improvements, helping it to achieve 81.95% and 83.46%, respectively.

Besides, ViTKD is a feature-based knowledge distillation method and can be combined with other logit-based methods for image classification. Therefore, we try to add the state-of-the-art logit-based distillation loss NKD to our ViTKD. In this way, the students with different teachers all get another significant accuracy improvement, *e.g.*, the student DeiT-Small gets another 1.64% gains and achieves 83.59% Top-1 accuracy with a DeiT III-Base teacher. Surprisingly, the student DeiT-Small is just trained on ImageNet-1K, but its performance surpasses DeiT III-Small, which needs to be pre-trained on ImageNet-21k and then finetuned on ImageNet-1k.

4.3 DOWNSTREAM TASK

The model with ViTKD achieves significant improvements for the classification task on ImageNet. To further evaluate the effectiveness of the model with distillation, we try to apply the model to object detection. We use Mask-RCNN (He et al., 2017) as the detector and follow the training setting from ViTDet (Li et al., 2022b) on detectron2 (Wu et al., 2019).

As the results shown in Table 6, the backbone DeiT-Small trained with ViTKD brings the detector 1.21 mAP gains. When the backbone is trained with ViTKD and NKD, the mAP improvement can

Table 6: The detection results on COCO. We use Mask-RCNN as the detector.

Distillation	DeiT-Small		DeiT-Base	
	AP ^{box}	AP ^{mask}	AP ^{box}	AP ^{mask}
-	45.07	40.14	47.23	41.88
ViTKD	46.28 (+1.21)	41.05 (+0.91)	48.13 (+0.90)	42.82 (+0.94)
ViTKD+NKD	46.69 (+1.62)	41.38 (+1.24)	48.83 (+1.60)	43.19 (+1.31)

be boosted to 1.62. The results demonstrate the model trained with distillation has not only better performance for image classification but also stronger semantic information for the downstream task.

5 MORE ANALYSES

5.1 DO WE NEED TO DISTILL THE MIDDLE LAYERS?

We have discussed the distillation strategies for the shallow and deep layers of ViT-based models. As shown in Figure 1 and 4, the attention distributions of the middle layers are similar to that of shallow layers. Accordingly, we explore the effects of distillation on the middle layers (*e.g.*, the 6_{th} layer) in Table 7. In general, distillation on either the shallow or middle layers can benefit the student. The first layer’s knowledge boosts the student most. Besides comparing the improvements from different distillation layers, we find that the knowledge from the shallow layers is much more helpful than that from the middle layer for distillation. Furthermore, when combining the shallow and middle layers together, the accuracy improvement is just 0.02%. Considering the trade-offs between time consumption and performance, we do not distill on the middle layers eventually.

Table 7: The effect of distillation on different layers by a *Mimicking* way.

Layer	DeiT Small – DeiT Tiny						
0	-	✓	-	-	✓	✓	
1	-	-	✓	-	✓	✓	
6	-	-	-	✓	-	✓	
Top-1	74.42	75.01	74.97	74.72	75.12	75.14	

Table 8: The Top-1 accuracy comparisons of using different teachers to distill the student DeiT-Tiny’s shallow layers via mimicking.

Method	Teacher	Student
baseline	-	74.42
CaiT-S24	83.37	73.12 (-1.30)
DeiT-Small	80.69	75.12 (+0.70)
DeiT III-Small	82.76	75.31 (+0.89)

5.2 TEACHERS WITH THE SAME ARCHITECTURE AS THE STUDENT ARE APPROPRIATE.

We have chosen teachers with the same architecture to guide the student in Table 5, achieving significant improvements. In this subsection, we explore whether a teacher with different architecture is still suitable for distillation. Here we choose CaiT-S24 (Touvron et al., 2021b) as the teacher, which has a high performance of 83.37% and a different architecture with DeiT. Following the previous guidance, we utilize CaiT-S24’s shallow layers to distill the student’s shallow layers, which leads to a 1.29% accuracy drop (shown in Table 8). To further analyze what causes the degradation, we visualize the attention maps of the three used models in Figure 4. Interestingly, the shallow and deep layer’s attention distributions of DeiT and CaiT are quite different, making it hard for the student to learn such attention. This phenomenon is not consistent with the assumptions of our proposed ViTKD, which causes inevitable performance degradation. In contrast, when using a teacher with the same architecture for distillation, the student gains noticeably. This observation indicates that a teacher with the same architecture who generates similar attention, is more suitable for ViTKD.

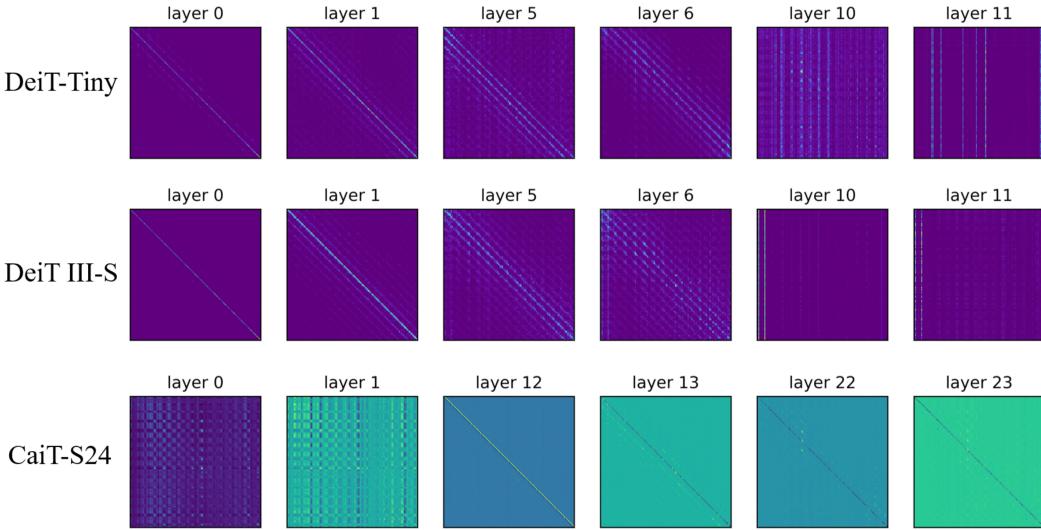


Figure 4: Visualization of the average attention map from the student (DeiT-Tiny) and two different teachers (DeiT III-Small and CaiT-S24).

Table 9: Ablation study of the losses of *Mimicking* and *Generation* distillation.

Losses	DeiT-Tiny (Student)			
\mathcal{L}_{lr}	-	✓	-	✓
\mathcal{L}_{gen}	-	-	✓	✓
DeiT-Small (Teacher)	74.42	75.12	74.72	75.40
DeiT III-Small (Teacher)	74.42	75.31	74.79	76.06

5.3 EFFECTS OF DIFFERENT LOSSES

As described in the practical guidance, we distill the shallow layers and deep layers by mimicking and generation, respectively. In this subsection, we conduct experiments of *Mimicking* loss \mathcal{L}_{lr} and *Generation* loss \mathcal{L}_{gen} to investigate their influences on the student with DeiT-Tiny. As shown in Table 9, both the knowledge from shallow and deep layers are helpful for the student. When just applying a single loss, the \mathcal{L}_{lr} on shallow layers benefits the student much more than \mathcal{L}_{gen} on the deep layer. This phenomenon shows that incipient attention knowledge really matters for ViT’s feature distillation, which is completely different from the CNN-based model’s feature distillation. Furthermore, these two losses are complementary to each other. For example, when combining \mathcal{L}_{lr} and \mathcal{L}_{gen} together, the student with a DeiT III-Small teacher achieves 76.06% Top-1 Accuracy, which is much higher than just applying \mathcal{L}_{lr} ’s 75.31% and \mathcal{L}_{gen} ’s 74.79%.

5.4 SENSITIVITY STUDY OF HYPER-PARAMETERS

In ViTKD, we use α and β in Equation 9 to balance the shallow layer’s distillation loss \mathcal{L}_{lr} and the deep layer’s distillation loss \mathcal{L}_{gen} , respectively. To explore the sensitivity of the hyper-parameters, we conduct experiments by adopting DeiT III-Small to distill DeiT-Tiny on ImageNet-1K. As shown in Figure 5, ViTKD is not sensitive to the hyper-parameters α or β which is just used for balancing the distillation loss. Specifically, when α varies from 2 to 6, the student’s worst accuracy is 76.03%, which is just 0.12% lower than the highest accuracy. Besides, it is still 1.61% higher than the baseline model, demonstrating our ViTKD is not sensitive to the hyper-parameters.

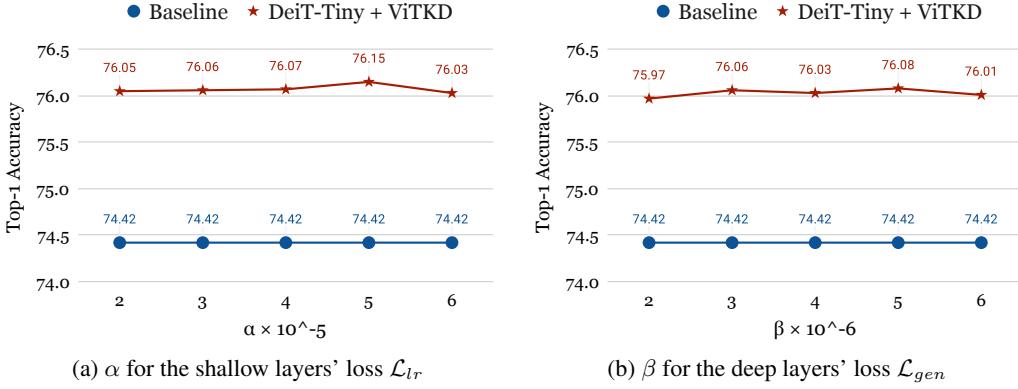


Figure 5: The sensitivity study of hyper-parameters α (a) and β (b).

6 CONCLUSION

In this paper, we explore a feature-based distillation method for ViT-based models. To this end, we design a series of experiments and discuss the effects of different distillation methods, layers, and modules. From the results, We derive three practical guidelines for ViT's feature distillation. We propose our method ViTKD based on the guidelines, which includes the distillation on shallow layers via *mimicking* and deep layers via *generation*. ViTKD brings the student significant improvements on the image classification task and also benefits other downstream task. Besides, ViTKD is truly a feature-based method that can be easily combined with logit-based distillation methods to further improve the student.

LIMITATIONS

We use the *mimicking* method for the shallow layer's distillation and the *generation* method for the deep layer's distillation. However, the way to achieve *mimicking* and *generation* is still simple and needs further exploration. Moreover, it is still interesting to transfer the feature knowledge to the student from a teacher with different architecture.

REFERENCES

- Pengguang Chen, Shu Liu, Hengshuang Zhao, and Jiaya Jia. Distilling knowledge via knowledge review. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5008–5017, 2021.
- Xiaokang Chen, Mingyu Ding, Xiaodi Wang, Ying Xin, Shentong Mo, Yunhao Wang, Shumin Han, Ping Luo, Gang Zeng, and Jingdong Wang. Context autoencoder for self-supervised representation learning. *arXiv preprint arXiv:2202.03026*, 2022.
- MMClassification Contributors. Openmmlab’s image classification toolbox and benchmark. <https://github.com/open-mmlab/mmclassification>, 2020.
- Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009.
- 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.
- 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. In *International Conference on Learning Representations*, 2021.
- Kai Han, An Xiao, Enhua Wu, Jianyuan Guo, Chunjing Xu, and Yunhe Wang. Transformer in transformer. *Advances in Neural Information Processing Systems*, 34:15908–15919, 2021.
- Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pp. 2961–2969, 2017.
- Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16000–16009, 2022.
- Tong He, Chunhua Shen, Zhi Tian, Dong Gong, Changming Sun, and Youliang Yan. Knowledge adaptation for efficient semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 578–587, 2019.
- Byeongho Heo, Jeesoo Kim, Sangdoo Yun, Hyojin Park, Nojun Kwak, and Jin Young Choi. A comprehensive overhaul of feature distillation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1921–1930, 2019.
- Geoffrey Hinton, Oriol Vinyals, Jeff Dean, et al. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2(7), 2015.
- Gang Li, Xiang Li, Yujie Wang, Shanshan Zhang, Yichao Wu, and Ding Liang. Knowledge distillation for object detection via rank mimicking and prediction-guided feature imitation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 1306–1313, 2022a.
- Yanghao Li, Hanzi Mao, Ross Girshick, and Kaiming He. Exploring plain vision transformer backbones for object detection. *arXiv preprint arXiv:2203.16527*, 2022b.
- Sihao Lin, Hongwei Xie, Bing Wang, Kaicheng Yu, Xiaojun Chang, Xiaodan Liang, and Gang Wang. Knowledge distillation via the target-aware transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10915–10924, 2022.
- Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pp. 740–755, 2014.
- 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*, pp. 2604–2613, 2019.

-
- 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. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10012–10022, 2021.
- Wonpyo Park, Dongju Kim, Yan Lu, and Minsu Cho. Relational knowledge distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3967–3976, 2019.
- Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32: 8026–8037, 2019.
- Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets. *arXiv preprint arXiv:1412.6550*, 2014.
- Changyong Shu, Yifan Liu, Jianfei Gao, Zheng Yan, and Chunhua Shen. Channel-wise knowledge distillation for dense prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5311–5320, 2021.
- Siqi Sun, Yu Cheng, Zhe Gan, and Jingjing Liu. Patient knowledge distillation for bert model compression. In *EMNLP/IJCNLP (1)*, 2019.
- Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive representation distillation. In *International Conference on Learning Representations*, 2019.
- 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*, pp. 10347–10357, 2021a.
- Hugo Touvron, Matthieu Cord, Alexandre Sablayrolles, Gabriel Synnaeve, and Hervé Jégou. Going deeper with image transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 32–42, 2021b.
- Hugo Touvron, Matthieu Cord, and Hervé Jégou. Deit iii: Revenge of the vit. *arXiv preprint arXiv:2204.07118*, 2022.
- Luting Wang, Xiaojie Li, Yue Liao, Zeren Jiang, Jianlong Wu, Fei Wang, Chen Qian, and Si Liu. Head: Hetero-assists distillation for heterogeneous object detectors. *arXiv preprint arXiv:2207.05345*, 2022.
- Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. <https://github.com/facebookresearch/detectron2>, 2019.
- Chenhongyi Yang, Mateusz Ochal, Amos Storkey, and Elliot J Crowley. Prediction-guided distillation for dense object detection. *arXiv preprint arXiv:2203.05469*, 2022a.
- Chuang Yang, Helong Zhou, Zhulin An, Xue Jiang, Yongjun Xu, and Qian Zhang. Cross-image relational knowledge distillation for semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12319–12328, 2022b.
- Jing Yang, Brais Martinez, Adrian Bulat, and Georgios Tzimiropoulos. Knowledge distillation via softmax regression representation learning. In *International Conference on Learning Representations*, 2020.
- Zhendong Yang, Zhe Li, Yuan Gong, Tianke Zhang, Shanshan Lao, Chun Yuan, and Yu Li. Rethinking knowledge distillation via cross-entropy. *arXiv preprint arXiv:2208.10139*, 2022c.
- Zhendong Yang, Zhe Li, Xiaohu Jiang, Yuan Gong, Zehuan Yuan, Danpei Zhao, and Chun Yuan. Focal and global knowledge distillation for detectors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4643–4652, 2022d.
- Zhendong Yang, Zhe Li, Mingqi Shao, Dachuan Shi, Zehuan Yuan, and Chun Yuan. Masked generative distillation. *arXiv preprint arXiv:2205.01529*, 2022e.

-
- Li Yuan, Yunpeng Chen, Tao Wang, Weihao Yu, Yujun Shi, Zi-Hang Jiang, Francis EH Tay, Jiashi Feng, and Shuicheng Yan. Tokens-to-token vit: Training vision transformers from scratch on imagenet. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 558–567, 2021.
- 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.
- Jinnian Zhang, Houwen Peng, Kan Wu, Mengchen Liu, Bin Xiao, Jianlong Fu, and Lu Yuan. Minivit: Compressing vision transformers with weight multiplexing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12145–12154, 2022.
- Borui Zhao, Quan Cui, Renjie Song, Yiyu Qiu, and Jiajun Liang. Decoupled knowledge distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11953–11962, 2022.
- Zhaohui Zheng, Rongguang Ye, Ping Wang, Dongwei Ren, Wangmeng Zuo, Qibin Hou, and Ming-Ming Cheng. Localization distillation for dense object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9407–9416, 2022.
- Helong Zhou, Liangchen Song, Jiajie Chen, Ye Zhou, Guoli Wang, Junsong Yuan, and Qian Zhang. Rethinking soft labels for knowledge distillation: A bias–variance tradeoff perspective. In *International Conference on Learning Representations*, 2020.