ScaleKD: Strong Vision Transformers Could Be Excellent Teachers

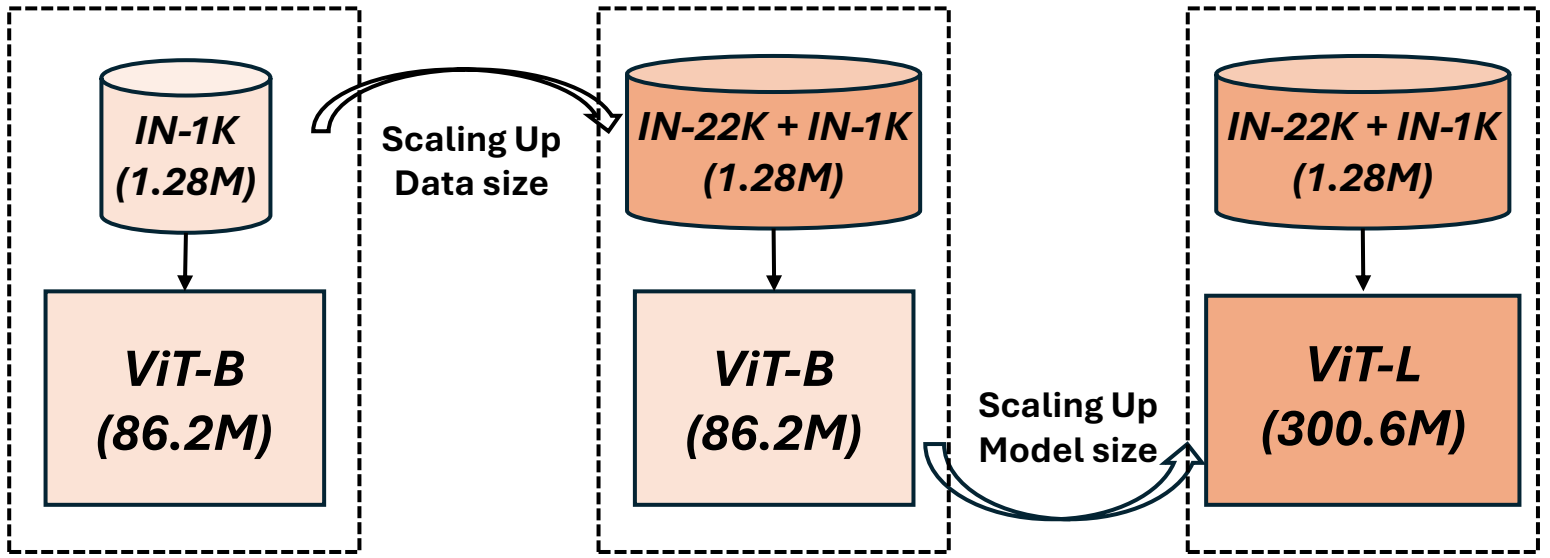
Jiawei Fan[1], Chao Li[1], Xiaolong Liu[2], and Anbang Yao[1]

[1] Intel Labs China

[2] iMotion Automotive Technology

Background

- **Scalable Properties of Vision Transformer:** When scaling up the data size and model size, the performance of vision transformer increases constantly.



Motivation of This Work

- Large ViT models with mass pre-training have attained state-of-the-art performance.
- **Motivation:** *Whether large pre-trained ViT models could be used as teachers that effectively transfer their scalable properties to target student models having different typed architectures such as* ***CNN and MLP or heterogeneous ViT structures***.

Problem Analysis

- To answer the question in our motivation, we think the knowledge transfer difficulties are rooted in the following three aspects of differences:
  - Difference in feature computing paradigm.
  - Difference in model scale.
  - Difference in knowledge density.
- The first difference is explicitly defined, while the other two are intertwined under the prevailing pre-training and fine-tuning paradigm and are finally encoded in teacher and student models' **feature space and parameter space**.

Two observations in Feature Space and Parameter space

- Feature Space: The frequency distributions of the features for the pre-trained ViTs are extremely imbalanced, where the direct component (zero frequency) response is dominant among all frequencies.
- Parameter Space: As the parameters of the pre-trained ViTs in the fine-tuning stage are slightly changed, their pre-training knowledge remains in the parameter space.

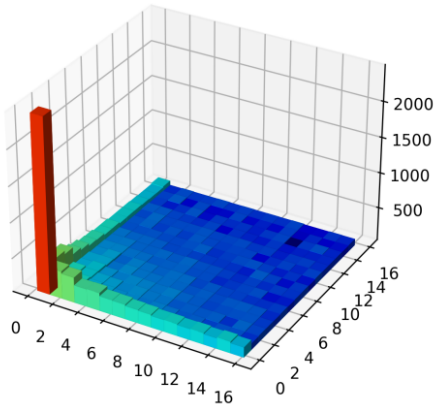
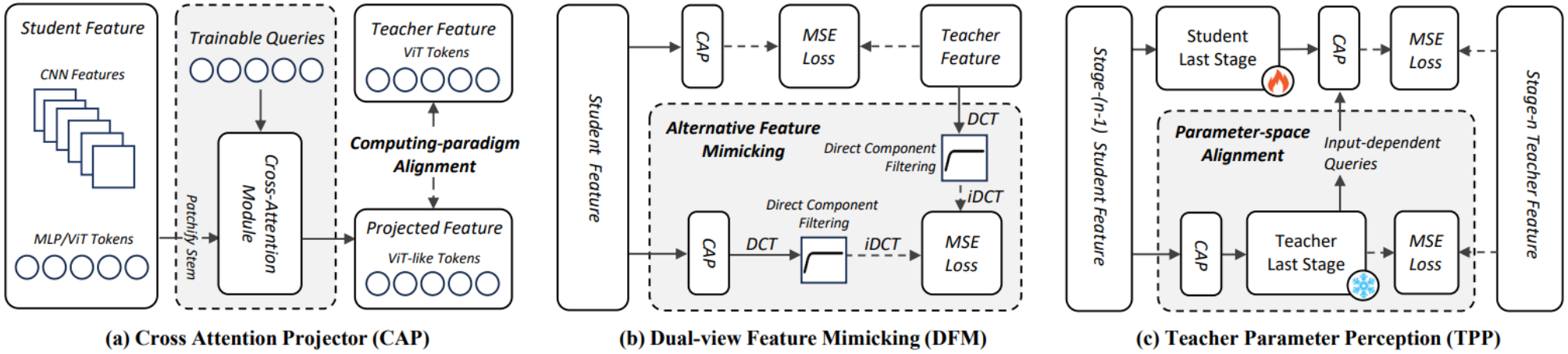


Figure 2: Feature distribution of BEiT-L/14 [41] in the frequency domain, where the direct component response is dominant. Details on drawing this figure are shown in Figure 5.

Three Core Components in ScaleKD

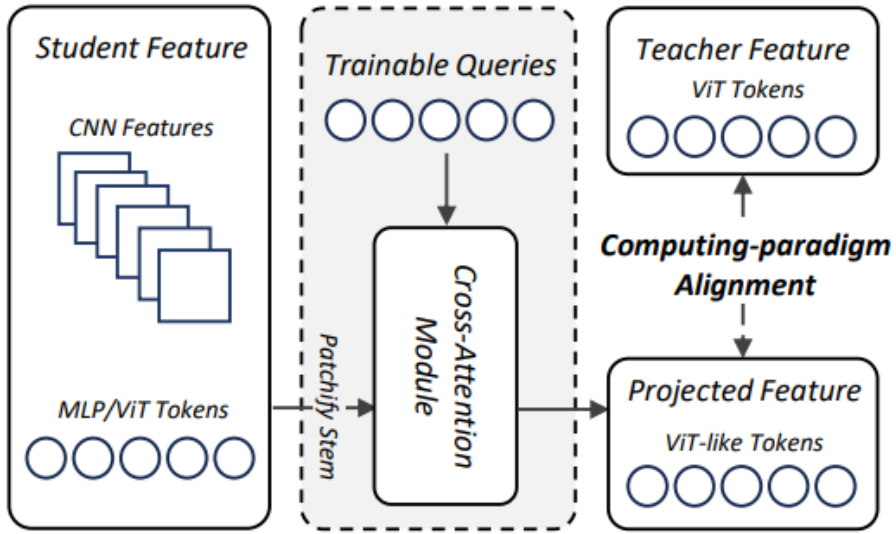
- Cross Attention Projector -> Aligning feature computing paradigm differences
- Dual-view Feature Mimicking -> Learning in Feature Space
- Teacher Parameter Perception -> Learning in Parameter Space



Cross Attention Projector

- CAP adopts the structure of a standard transformer decoder block, but incorporates three critical modifications (taking CNN as an example):
  - Patchifying regular grids of pixels in CNN
  - Adding positional embeddings
  - Trainable variables
- With CAP, the feature distillation loss is defined as:

$$\mathcal{L}_{CAP} = \alpha L(F^t, f_p(F^s; q)) = \alpha ||F^t - f_p(F^s; q)||_2^2,$$



(a) Cross Attention Projector (CAP)

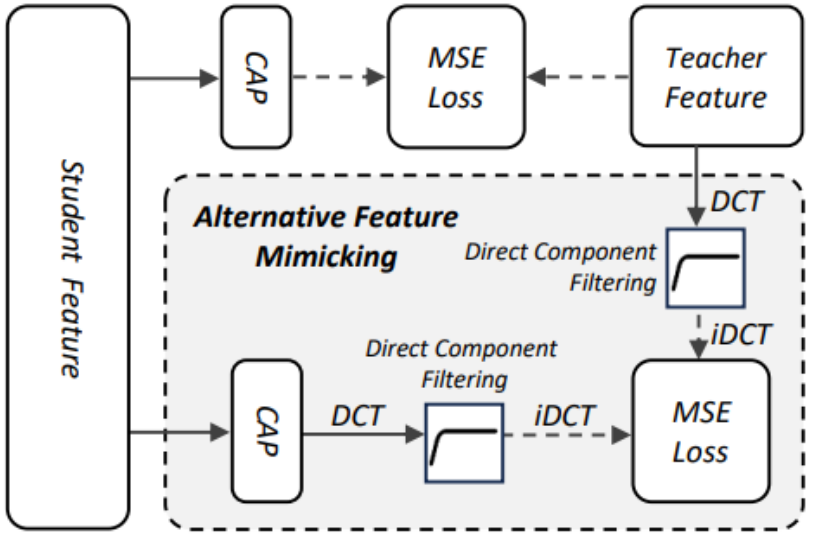
Dual-view Feature Mimicking

- In the first path, DFM conducts feature mimicking in the teacher's original feature space: $\mathcal{L}_{ori} = \alpha L(F^t, f_{p_1}(F^s, q_1))$
- In the second path, the dominant direct component should be removed. Thus, we define:

$$\phi(x) = DCT^{-1}(\sigma(DCT(x))) \qquad s.t.\ \sigma(z) = \begin{cases} 0, & z = 0 \\ z, & z \neq 0 \end{cases}.$$

- Next, feature mimicking in the second path is formulated as: $\mathcal{L}_{alt} = \alpha L(\phi(F^t), \phi(f_{p_2}(F^s; q_2)))$
- Now, the feature distillation loss of DFM is formulated as:

$$\mathcal{L}_{DFM} = \beta \mathcal{L}_{ori} + (1 - \beta)\mathcal{L}_{alt},$$

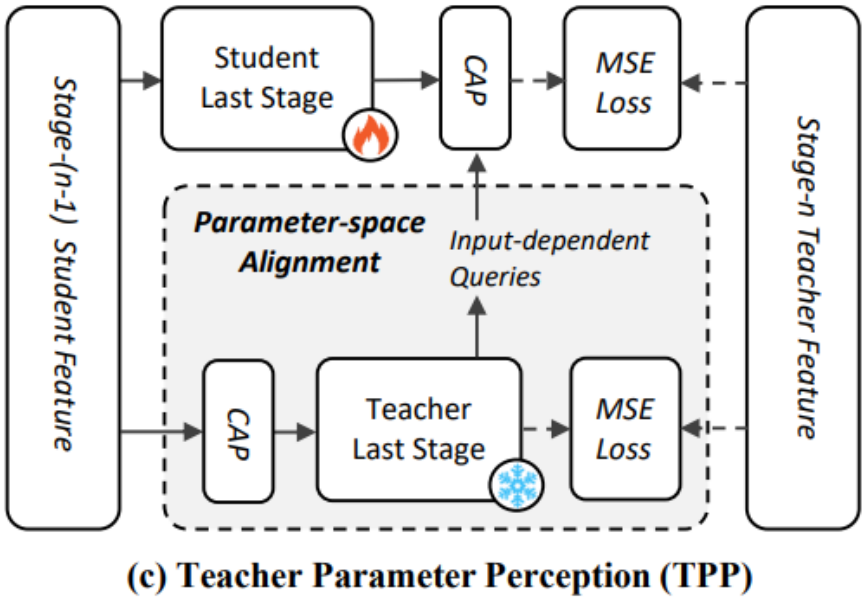


(b) Dual-view Feature Mimicking (DFM)

Teacher Parameter Perception

- TPP establishes a proxy feature processing path by connecting the student's early stages to the teacher's later stages through a CAP. The feature mimicking in the proxy path is formulated as: $\mathcal{L}^{st} = \alpha L(F^t, F^{st})$.
- With a simple principle of equal treatment to the two feature mimicking paths, the feature distillation loss of TPP is defined as:

$$\mathcal{L}^{TPP} = \mathcal{L}^{s} + \mathcal{L}^{st}.$$



(c) Teacher Parameter Perception (TPP)

The Overall Formulation of ScaleKD

- From a general perspective, the progressive designs of our above three components are naturally coupled.
- As CAP serves as the basic component in DFM and TPP, we further introduce how to apply DFM in TPP and get a neat formulation of our method, ScaleKD. Specifically, if treating DFM as an improved version of traditional feature mimicking, it can substitute the original feature mimicking in each path of TPP.

$$\mathcal{L}_{ScaleKD} = \mathcal{L}_{task} + \underbrace{\beta\mathcal{L}^{s}_{ori} + (1-\beta)\mathcal{L}^{s}_{alt}}_{\textit{DFM for TPP Student Path}} + \underbrace{\beta\mathcal{L}^{st}_{ori} + (1-\beta)\mathcal{L}^{st}_{alt}}_{\textit{DFM for TPP Teacher Path}} + \mathcal{L}_{kd},$$

Pilot Experiments under Basic Settings

Table 1: Pilot experiments on cross architecture distillation with ScaleKD and FD. s_i denotes the distillation is conducted on stage-i. To clearly show the performance gain, experiments in this table are conducted without L_{kd}.

Teacher	Student	Method	Top-1(%)	ΔTop-1(%)
Swin-S (83.02)	ResNet-50	*Baseline*	76.55	-
		FD (s_4)	77.43	+0.88
		FD (s_3,s_4)	77.74	+1.19
		ScaleKD	79.30	+2.75
	Mixer-S	*Baseline*	74.02	-
		FD (s_4)	74.88	+0.86
		FD (s_3,s_4)	75.32	+1.30
		ScaleKD	77.24	+3.22

Table 2: Pilot experiments on scaling up the teacher size. The advanced training strategy uses more sophisticated data augmentation and optimizer, and longer training epochs, as shown in Table 10.

Teacher	Student	Ratio of T/S Params	Top-1(%)	ΔTop-1(%)
ScaleKD with Traditional Training Strategy				
Baseline	ResNet-50	-	76.55	-
Swin-S (83.02)		1.94×	79.62	+3.07
Swin-B (85.16)		3.43×	79.80	+3.25
Swin-L (86.24)		7.68×	80.10	+3.55
ScaleKD with Advanced Training Strategy				
Baseline	ResNet-50	-	78.64	-
Swin-S (83.02)		1.94×	81.43	+2.79
Swin-B (85.16)		3.43×	81.77	+3.13
Swin-L (86.24)		7.68×	82.03	+3.39

Main Results

Table 3: Main results of ScaleKD on **11** teacher-student network pairs. † denotes the model pre-trained on IN-22K [45] and ‡ denotes the model pre-trained by EVA [41], which has the learned knowledge of LAION-2B [48].

Teacher	Student	Params (M)		FLOPs (G)		Accuracy (%)	
		T	S	T	S	Top-1	ΔTop-1
Swin-L† (86.24)	MobileNet-V1 (72.10)	196.53	4.23	34.04	0.58	75.15	+3.05
	ResNet-50 (78.64)		25.56		4.12	82.03	+3.39
	ConvNeXt-T (82.14)		28.59		4.46	84.16	+2.02
	Mixer-S/16 (74.02)	196.53	18.53	34.04	3.78	78.63	+4.61
	Mixer-B/16 (76.44)		59.88		12.61	81.96	+5.52
	ViT-S/16 (79.90)	196.53	22.05	34.04	4.61	83.93	+4.03
	Swin-T (81.18)		28.29		4.36	83.80	+2.62
	ViT-B/16 (81.80)		86.57		17.58	85.53	+3.73
BEiT-L/14‡ (88.58)	ResNet-50 (78.64)	304.14	25.56	81.06	4.12	82.34	+3.70
	Mixer-B/14 (76.62)		59.88		16.45	82.89	+6.27
	ViT-B/14 (82.02)		86.57		23.09	86.43	+4.41

The Scalable Properties from Teacher's Pre-training Data

Table 4: Experiments on exploring scalable properties from the teacher's pre-training data. We use the best reported models with different pre-training methods as our baselines to examine whether our student model has learned the teacher's pre-training knowledge. We use Swin-L as the teacher for the first two experiments and BEiT-L/14 as the teacher for the rest two experiments. $\Rightarrow$ denotes transfer learning and * denotes the model is trained and tested with 384$\times$ 384 sample resolution.

Model	Method	Training Dataset	Dataset Samples $\times$ Epochs (M)	Viewed Samples (M)	Top-1(%)
Supervised pre-training					
ViT-B/16	Pre-training [4]	IN-22K $\Rightarrow$ IN-1K	13.7$\times$90 + 1.28$\times$32	1274	83.97
		JFT-300 $\Rightarrow$ IN-1K	300$\times$7 + 1.28$\times$32	2141	84.15
	ScaleKD	IN-1K	1.28$\times$300	384	85.53
Self-supervised pre-training					
ViT-B/16	BEiT [40]	IN-22K $\Rightarrow$ IN-1K	13.7$\times$150 + 1.28$\times$100	2183	83.70
	iBOT [11]	IN-22K $\Rightarrow$ IN-1K	13.7$\times$320 + 1.28$\times$100	4512	84.40
	ScaleKD	IN-1K	1.28 $\times$ 300	384	85.64
Cross-modal pre-training					
ViT-B/16	CLIP [13]	LAION-2B $\Rightarrow$ IN-1K	2320$\times$32 + 1.28$\times$50	74304	85.47
		LAION-2B $\Rightarrow$ IN-12K $\Rightarrow$ IN-1K	2320$\times$32 + 12.1$\times$60 + 1.28$\times$50	75030	86.17
ViT-B/14	ScaleKD	IN-1K	1.28 $\times$ 300	384	86.43
EVA hybrid pre-training (MIM distillation from the cross-modal pre-trained teacher)					
EVA02-S/14*	EVA-02 [49]	IN-22K $\Rightarrow$ IN-1K	13.7$\times$240 + 1.28$\times$50	3352	85.80
	ScaleKD	IN-1K	1.28 $\times$ 300	384	86.22

Transferring to Downstream Tasks

- To further examine whether the performance gains from our method could be well preserved in transfer learning, we conduct comparative experiments on MS-COCO for object detection and instance segmentation, and on ADE20K for semantic segmentation.

Table 6: Transfer learning results (%) on ADE20K.

Framework	Backbone	Pre-training	IN-1K (Top-1)	ADE20K (mIOU)
UperNet	ResNet-50	*Baseline*	78.64	42.37
		Ours	82.03 (+3.39)	44.50 (+2.13)
	Swin-T	*Baseline*	81.18	44.41
		Ours	83.80 (+2.62)	46.33 (+1.92)
	ViT-B/16	*Baseline*	81.80	46.75
		Ours	85.53 (+3.73)	50.84 (+4.09)

Table 5: Transfer learning results (%) on MS-COCO.

Framework	Backbone	Pre-training	Classification (IN-1K)	Object Detection (COCO)				Instance Segmentation (COCO)			
			Top-1	AP	AP_S	AP_M	AP_L	AP	AP_S	AP_M	AP_L
Mask R-CNN	ResNet-50	*Baseline*	78.64	40.2	23.0	44.3	52.5	37.1	18.0	40.1	54.9
		Ours	82.03 (+3.39)	42.3	25.5	46.5	54.6	39.1	19.3	42.5	57.1
	Swin-T	*Baseline*	81.18	42.7	26.5	45.9	56.6	39.3	20.5	41.8	57.8
		Ours	83.80 (+2.62)	44.4	28.7	47.9	58.6	40.8	21.8	43.7	59.8

Conclusion

- In this paper, we present ScaleKD, a new cross architecture KD approach for transferring the scalable properties of pre-trained large ViTs to various CNNs, MLPs and heterogeneous ViTs.
- Our method consists of three tightly coupled components that rely on principled designs to align computing paradigm differences, model scale differences, and knowledge density differences between the teacher and the student.
- By conducting systematic experiments on several mainstream large-scale vision benchmarks, we broadly validate the effectiveness and generalization ability of our method.
- Benefiting from its novel motivation and design insights, ScaleKD is the first work which successfully verified that KD can be a more efficient alternative to the time-intensive pre-training, to the best of our knowledge. This extends the application scope of KD from model compression to training acceleration.