A Closer Look at Self-Supervised Lightweight Vision Transformers

Shaoru Wang [1 2] **Jin Gao** [1 2] **Zeming Li** [3] **Xiaoqin Zhang** [4] **Weiming Hu** [1 2 5]

Abstract

Self-supervised learning on large-scale Vision Transformers (ViTs) as pre-training methods has achieved promising downstream performance. Yet, how much these pre-training paradigms promote lightweight ViTs' performance is considerably less studied. In this work, we develop and benchmark several self-supervised pre-training methods on image classification tasks and some downstream dense prediction tasks. We surprisingly find that if proper pre-training is adopted, even vanilla lightweight ViTs show comparable performance to previous SOTA networks with delicate architecture design. It breaks the recently popular conception that vanilla ViTs are not suitable for vision tasks in lightweight regimes. We also point out some defects of such pre-training, *e.g.*, failing to benefit from large-scale pre-training data and showing inferior performance on data-insufficient downstream tasks. Furthermore, we analyze and clearly show the effect of such pre-training by analyzing the properties of the layer representation and attention maps for related models. Finally, based on the above analyses, a distillation strategy during pre-training is developed, which leads to further downstream performance improvement for MAE-based pre-training. Code is available at https://github.com/wangsr126/mae-lite.

1. Introduction

Self-supervised learning (SSL) has shown great progress in representation learning without heavy reliance on expensive labeled data. SSL focuses on various pretext tasks for pre-training. Among them, several works (He et al., 2020; Chen et al., 2020; Grill et al., 2020; Caron et al., 2020; Chen et al., 2021a; Caron et al., 2021) based on contrastive learning (CL) have achieved comparable or even better accuracy than supervised pre-training when transferring the learned representations to downstream tasks. Recently, another trend focuses on masked image modeling (MIM) (Bao et al., 2021; He et al., 2021; Zhou et al., 2022), which perfectly fits Vision Transformers (ViTs) (Dosovitskiy et al., 2020) for vision tasks, and achieves improved generalization performance. Most of these works, however, involve large networks with little attention paid to smaller ones. Some works (Fang et al., 2020; Abbasi Koohpayegani et al., 2020; Choi et al., 2021) focus on CL on small convolutional networks (ConvNets) and improve the performance by distillation. However, the pre-training of lightweight ViTs is considerably less studied.

Efficient neural networks are essential for modern on-device computer vision. Recent studies on achieving top-performing lightweight models mainly focus on designing network architectures (Sandler et al., 2018; Howard et al., 2019; Graham et al., 2021; Ali et al., 2021; Heo et al., 2021; Touvron et al., 2021b; Mehta & Rastegari, 2022; Chen et al., 2021b; Pan et al., 2022), while little attention is paid to how to optimize the training strategies for these models. We believe the latter is also of vital importance, and the utilization of pre-training is one of the most hopeful approaches along this way, since it has achieved great progress on large models. To this end, we develop and benchmark recently popular self-supervised pre-training methods, *e.g.*, CL-based MoCo-v3 (Chen et al., 2021a) and MIM-based MAE (He et al., 2021), along with fully-supervised pre-training for lightweight ViTs as the baselines on ImageNet and other classification tasks, as well as some dense prediction tasks, *e.g.*, object detection and segmentation. We surprisingly find that *if proper pre-training is adopted, even vanilla lightweight ViTs show comparable performance to previous SOTA networks with delicate design*, *e.g.*, we achieve 79.0% top-1 accuracy on ImageNet with vanilla ViT-Tiny (5.7M). The finding is intriguing since the result indicates that proper pre-training could bridge the performance gap between naive network architectures and delicately designed ones to a great extent, while naive architectures usually have

[1]State Key Laboratory of Multimodal Artificial Intelligence Systems, Institute of Automation, Chinese Academy of Sciences [2]School of Artificial Intelligence, University of Chinese Academy of Sciences [3]Megvii Research [4]Key Laboratory of Intelligent Informatics for Safety & Emergency of Zhejiang Province, Wenzhou University [5]School of Information Science and Technology, ShanghaiTech University. Correspondence to: Jin Gao <jin.gao@nlpr.ia.ac.cn>.

Proceedings of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023.

faster inference speed, by getting rid of some complicated operators. We also point out some defects of such pre-training, *e.g.*, *failing to benefit from large-scale pre-training data* and *showing inferior performance on data-insufficient downstream tasks*.

These findings motivate us to dive deep into the working mechanism of these pre-training methods for lightweight ViTs. More specifically, we introduce a variety of model analysis methods to study the pattern of layer behaviors during pre-training and fine-tuning, and investigate what really matters for downstream performance. First, we find that *lower layers of the pre-trained models matter more than higher ones if sufficient downstream data is provided, while higher layers matter in data-insufficient downstream tasks*. Second, we observe that *the pre-training with MAE makes the attention of the downstream models more local and concentrated,* i.e.*, introduces locality inductive bias, which may be the key to the performance gain*. Based on the above analyses, we also develop a distillation strategy for MAE-based pre-training, which significantly improves the pre-training of lightweight ViTs. Better downstream performance is achieved especially on data-insufficient classification tasks and detection tasks.

2. Preliminaries and Experimental Setup

ViTs. We use ViT-Tiny (Touvron et al., 2021a) in our study to examine the effect of the pre-training on downstream performance, which contains 5.7M parameters. We adopt the vanilla architecture, consisting of a patch embedding layer and 12 Transformer blocks with an embedding dimension of 192, except that the number of heads is increased to 12 as we find it can improve the model's expressive power. ViT-Tiny is chosen for study because it is an ideal experimental object, on which almost all existing pre-training methods can be directly applied. And it has a rather naive architecture: non-hierarchical, and with low human inductive bias in design. Thus the influence of the model architecture design on our analyses can be eliminated to a great extent.

Evaluation Metrics. We adopt *fine-tuning* as the default evaluation protocol considering that it is highly correlated with utility (Newell & Deng, 2020), in which all the layers are tuned by initializing them with the pre-trained models. By default, we do the evaluation on ImageNet (Deng et al., 2009) by fine-tuning on the training set and evaluating on the validation set. Several other downstream classification datasets (*e.g.*, Flowers (Nilsback & Zisserman, 2008), Aircraft (Maji et al., 2013), CIFAR100 (Krizhevsky et al., 2009), *etc.*) and object detection and segmentation tasks on COCO (Lin et al., 2014) are also exploited for comparison. For a more thorough study, analyses based on *linear probing* evaluation are presented in Appendix B.2.

Compared Methods. *Baseline*: We supervisedly train a ViT-Tiny from scratch for 300 epochs on the training set of ImageNet-1k (dubbed IN1K). It achieves 74.5% top-1 accuracy on the validation set of ImageNet-1k, surpassing that in the original architecture (72.2% (Touvron et al., 2021a)) through modifying the number of heads to 12 from 3, and further reaches 75.8% by adopting our improved training recipe (see Appendix A.1), which finally serves as our strong baseline to examine the pre-training. We denote this model from supervised training as DeiT-Tiny.

MAE: MAE (He et al., 2021) is selected as a representative for MIM-based pre-training methods, which has a simple framework with low training cost. We largely follow the design of MAE except that the encoder is altered to ViT-Tiny. Several basic factors and components are adjusted to fit the smaller encoder (see Appendix A.2). By default, we do pre-training on IN1K for 400 epochs, and denote the pre-trained model as MAE-Tiny.

MoCov3: We also implement a contrastive SSL pre-training counterpart, MoCo-v3 (Chen et al., 2021a), which is selected for its simplicity. We also do 400-epoch pre-training and denote the pre-trained model as MoCov3-Tiny. Details are provided in Appendix A.3.

Some other methods, *e.g.*, MIM-based SimMIM (Xie et al., 2022) and CL-based DINO (Caron et al., 2021) are also involved, but are moved to Appendix B.3 due to space limitation.

3. How Well Does Pre-Training Work on Lightweight ViTs?

In this section, we first benchmark the aforementioned pre-trained models on ImageNet, and then further evaluate their transferability to other datasets and tasks.

3.1. Benchmarks on ImageNet Classification Tasks

Which pre-training method performs best? We first develop and benchmark the pre-training methods on ImageNet, involving the baseline that does not adopt any pre-training, supervised pre-training on the training set of ImageNet-21k (a bigger and more diverse dataset, as roughly ten times the size of IN1K, dubbed IN21K) and the aforementioned self-supervised pre-training with MoCo-v3 and MAE. As reported in Tab. 1, most of these supervised and self-supervised pre-training methods improve the downstream performance, whilst *MAE outperforms others and consumes moderate training cost*. The results indicate that the vanilla ViTs have great potential, which can be unleashed via proper pre-training. It encourages us to further explore how the enhanced ViTs perform compared to recent SOTA ConvNets and ViT derivatives.

Table 1. **Comparisons on pre-training methods**. We report top-1 accuracy on the validation set of ImageNet-1k. IN1K and IN21K indicate the training set of ImageNet-1k and ImageNet-21k. The pre-training time is measured on 8×V100 GPU machine. 'ori.' represents the supervised training recipe from Touvron et al. (2021a) and 'impr.' represents our improved recipe (see Appendix A.1).

Methods	Pre-training Data	Epochs	Time (hour)	Fine-tuning recipe	Fine-tuning Top-1 Acc. (%)
-	-	-	-	ori.	74.5
-	-	-	-	impr.	75.8
Supervised (Steiner et al., 2021)	IN21K w/ labels	30	20	impr.	76.9
Supervised (Steiner et al., 2021)	IN21K w/ labels	300	200	impr.	77.8
MoCo-v3 (Chen et al., 2021a)	IN1K w/o labels	400	52	impr.	76.8†
MAE (He et al., 2021)	IN1K w/o labels	400	23	impr.	**78.0**

† Global average pooling is used instead of the default configuration based on the class token during the fine-tuning. See Appendix A.1 for details.

How do the enhanced ViTs with pre-training rank among SOTA lightweight networks? To answer the question, we further compare the enhanced ViT-Tiny with MAE pre-training to previous lightweight ConvNets and ViT derivatives. We report top-1 accuracy along with the model parameter count and the throughput in Tab. 3. We denote the fine-tuned model based on MAE-Tiny as MAE-Tiny-FT. Specifically, we extend the fine-tuning epochs to 1000 following Touvron et al. (2021a) and adopt relative position embedding. Under this strong fine-tuning recipe, the pre-training still contributes a 1.2 performance gain, ultimately reaching 79.0% top-1 accuracy. It sets a new record for lightweight vanilla ViTs, even without distillation during the supervised training phase on IN1K. It can also be seen that the pre-training can accelerate the downstream convergence, which helps to surpass that trained from scratch for 1000 epochs (77.8%) with only 300-epoch fine-tuning (78.5%).

We conclude that *the enhanced ViT-Tiny is on par with or even outperforms most previous ConvNets and ViT derivatives with comparable parameters or throughput*. This demonstrates that we can also achieve SOTA performance based on a naive network architecture by adopting proper pre-training, rather than designing complex ones. Significantly, naive architecture usually has faster inference speed and is friendly to deployment.

We also notice that there are some works applying supervised pre-training (Ridnik et al., 2021), CL-based self-supervised pre-training (Fang et al., 2020) and MIM-based self-supervised pre-training (Woo et al., 2023) on lightweight ConvNets. However, we find that ViT-Tiny benefits more from the pre-training (*e.g.*, +1.2 vs. +0.5 for ConvNeXt V2-F). We attribute it to that the plain architecture of ViT-Tiny with less artificial design may possess more model capacity.

Can the pre-training benefit from more data? One may be curious about whether it is possible to achieve better downstream performance by involving more pre-training data, as it does on large models. Unfortunately, the answer is no for the examined pre-training methods. We consider IN21K, a much larger dataset. The number of pre-training iterations is kept constant for a fair comparison. However, few improvements are observed for both MoCo-v3 and MAE as shown in Tab. 2. We further consider two subsets of IN1K containing 1% and 10% of the total examples (1% IN1K and 10% IN1K) balanced in terms of classes (Assran et al., 2021) and one subset with long-tailed class distribution (Liu et al., 2019) (IN1K-LT). Surprisingly, marginal performance declines are observed for MAE when pre-training on these subsets, showing more robustness than MoCo-v3 in terms of the pre-training data scale and class distribution.

Table 2. **Effect of pre-training data**. Top-1 accuracy is reported.

Datasets	MoCo-v3	MAE
IN1K	76.8	78.0
1% IN1K	76.2 (-0.6)	77.9 (-0.1)
10% IN1K	76.5 (-0.3)	78.0 (+0.0)
IN1K-LT	76.1 (-0.7)	77.9 (-0.1)
IN21K	76.9 (+0.1)	78.0 (+0.0)

3.2. Benchmarks on Transfer Performance

We further examine the transferability of these models pre-trained on IN1K, involving their transfer performance on some other classification tasks and dense prediction tasks. In addition to the self-supervised MAE-Tiny and MoCov3-Tiny, DeiT-Tiny is also involved, as a fully-supervised counterpart which is trained on IN1K for 300 epochs.

Can the pre-trained models transfer well on data-insufficient tasks? We introduce several classification tasks (Nilsback & Zisserman, 2008; Parkhi et al., 2012; Maji et al., 2013; Krause et al., 2013; Krizhevsky et al., 2009; Van Horn et al., 2018) to investigate their transferability. We conduct the transfer evaluation by fine-tuning these pre-trained models on these datasets (see Appendix A.4 for more details). As shown in Tab. 4, using various pre-training methods shows better performance than using random initialization, but the relative superiority and inferiority comparisons between these pre-training methods exhibit distinct characteristics from those on ImageNet. We find that *down-*

Table 3. **Comparisons with previous SOTA networks on ImageNet-1k.** We report top-1 accuracy along with throughput and parameter count. The throughput is borrowed from timm (Wightman, 2019), which is measured on a single RTX 3090 GPU with a batch size fixed to 1024 and mixed precision. '†' indicates that distillation is adopted during the supervised training (or fine-tuning). '⋆' indicates the original architecture of ViT-Tiny (the number of attention heads is 3).

Methods	pre-train data	fine-tuning epochs	#param.	throughput (image/s)	Accuracy Top-1 (%)
ConvNets					
ResNet-18 (He et al., 2016)	-	100	11.7M	8951	69.7
ResNet-50 (He et al., 2016; Wightman et al., 2021)	-	600	25.6M	2696	80.4
EfficientNet-B0 (Tan & Le, 2019)	-	450	5.3M	5369	77.7
EfficientNet-B0 (Fang et al., 2020)	IN1K w/o labels	450	5.3M	5369	77.2 (-0.5)
EfficientNet-B1 (Tan & Le, 2019)	-	450	7.8M	2953	78.8
MobileNet-v2 (Sandler et al., 2018)	-	480	3.5M	7909	72.0
MobileNet-v3 (Howard et al., 2019)	-	600	5.5M	9113	75.2
MobileNet-v3†(Ridnik et al., 2021)	IN21K	600	5.5M	9113	78.0
ConvNeXt V1-F (Liu et al., 2022)	-	600	5.2M	-	77.5
ConvNeXt V2-F (Woo et al., 2023)	-	600	5.2M	1816	78.0
ConvNeXt V2-F (Woo et al., 2023)	IN1K w/o labels	600	5.2M	1816	78.5 (+0.5)
Vision Transformers Derivative					
LeViT-128 (Graham et al., 2021)	-	1000	9.2M	13276	78.6
LeViT-192 (Graham et al., 2021)	-	1000	11.0M	11389	80.0
XCiT-T12/16†(Ali et al., 2021)	-	400	6.7M	3157	78.6
PiT-Ti†(Heo et al., 2021)	-	1000	5.1M	4547	76.4
CaiT-XXS-24†(Touvron et al., 2021b)	-	400	12.0M	1351	78.4
Swin-1G (Liu et al., 2021; Chen et al., 2021b)	-	450	7.3M	-	77.3
Mobile-Former-294M (Chen et al., 2021b)	-	450	11.4M	-	77.9
MobileViT-S (Mehta & Rastegari, 2022)	-	300	5.6M	1900	78.3
EdgeViT-XS (Pan et al., 2022)	-	300	6.7M	-	77.5
Vanilla Vision Transformers					
DeiT-Tiny⋆ (Touvron et al., 2021a)	-	300	5.7M	4844	72.2
DeiT-Tiny⋆†(Touvron et al., 2021a)	-	1000	5.7M	4764	76.6
DeiT-Tiny	-	300	5.7M	4020	76.2
MAE-Tiny-FT	IN1K w/o labels	300	5.7M	4020	78.5 (+2.3)
DeiT-Tiny	-	1000	5.7M	4020	77.8
MAE-Tiny-FT	IN1K w/o labels	1000	5.7M	4020	79.0 (+1.2)

Table 4. **Transfer evaluation on classification tasks and dense-prediction tasks**. Self-supervised pre-training approaches generally show inferior performance to the fully-supervised counterpart. Top-1 accuracy is reported for classification tasks and AP is reported for object detection (det.) and instance segmentation (seg.) tasks.The description of each dataset is represented as (train-size/test-size/#classes).

Init. \ Datasets	Flowers (2k/6k/102)	Pets (4k/4k/37)	Aircraft (7k/3k/100)	Cars (8k/8k/196)	CIFAR100 (50k/10k/100)	iNat18 (438k/24k/8142)	COCO(det.) (118k/50k/80)	COCO(seg.) (118k/50k/80)
Random	30.2	26.1	9.4	6.8	42.7	58.7	32.7	28.9
supervised								
DeiT-Tiny	**96.4**	**93.1**	73.5	**85.6**	**85.8**	**63.6**	**40.4**	**35.5**
self-supervised								
MoCov3-Tiny	94.8	87.8	**73.7**	83.9	83.9	54.5	39.7	35.1
MAE-Tiny	85.8	76.5	64.6	78.8	78.9	60.6	39.9	35.4

stream data scale matters. The self-supervised pre-training approaches achieve downstream performance far behind the fully-supervised counterpart, while the performance gap is narrowed more or less as the data scale of the downstream task increases. Moreover, MAE even shows inferior results to MoCo-v3. We conjecture that it is due to their different layer behaviors during pre-training and fine-tuning, which will be discussed in detail in the following section.

Can the pre-trained models transfer well on dense prediction tasks? For a more thorough study, we further conduct evaluations on downstream object detection and segmentation tasks on COCO (Lin et al., 2014), based on Li et al. (2021) (see Appendix A.5 for details) with different pre-trained models as initialization of the backbone. The results are shown in Tab. 4. The self-supervised pre-training also lags behind the fully-supervised counterpart.

4. Revealing the Secrets of the Pre-Training

In this section, we introduce some model analysis methods to study the pattern of layer behaviors during pre-training and fine-tuning, and investigate what matters for downstream performances.

4.1. Layer Representation Analyses

We first adopt Centered Kernel Alignment (CKA) method[1] (Cortes et al., 2012; Nguyen et al., 2020) to analyze the layer representation similarity across and within networks. Specifically, CKA computes the normalized similarity in terms of the Hilbert-Schmidt Independence Criterion (HSIC (Song et al., 2012)) between two feature maps or representations, which is invariant to the orthogonal transformation of representations and isotropic scaling (detailed in Appendix A.6).

Lower layers matter more than higher ones if sufficient downstream data is provided. We visualize the layer representation similarity between several pre-trained models and DeiT-Tiny as heatmaps in Fig. 1. We choose DeiT-Tiny, a classification model fully-supervisedly trained on IN1K, as the reference because we consider the higher similarity of the examined model's layer to that of DeiT-Tiny indicates its more relevance to recognition. Although the similarity does not directly indicate whether the downstream performance is good, it indeed reflects the pattern of layer representation to a certain extent. The similarity within DeiT-Tiny is also presented (the left column).

First, We observe a relatively high similarity between MAE-Tiny and DeiT-Tiny for lower layers, while low similarity for higher layers. In Appendix B.1, we observe similar phenomena with several additional supervisedly trained ViTs as the reference models. It indicates that fewer semantics are extracted for MAE-Tiny at a more abstract level in higher layers. In contrast, MoCov3-Tiny aligns DeiT-Tiny well across almost all layers. However, the fine-tuning evaluation in Tab. 1 shows that adopting the MAE-Tiny as initialization improves the performance more significantly than MoCov3-Tiny. Thus, we hypothesize that *lower layers matter much more than higher ones for the pre-trained models*. In order to verify the hypothesis, we design another experiment by only reserving several leading blocks of pre-trained models and randomly initializing the others, and then fine-tuning them on IN1K (for the sake of simplicity, we only fine-tune these models for 100 epochs). Fig. 2 shows that reserving only a certain number of leading blocks achieves a significant performance gain over randomly initializing all the blocks (*i.e.*, totally training from scratch) for both MAE-Tiny and MoCov3-Tiny. Whereas, further reserving higher layers leads to marginal gain for MAE-Tiny and MoCov3-Tiny, which demonstrates our hypothesis.

Higher layers matter in data-insufficient downstream tasks. Previous works (Touvron et al., 2021a; Raghu et al., 2021) demonstrate the importance of a relatively large dataset scale for fully-supervised high-performance ViTs with large model sizes. We also observe a similar phenomenon on lightweight ViTs even when the self-supervised pre-training is adopted as discussed in Sec. 3.2. It motivates us to study the key factor of downstream performance on data-insufficient tasks.

We conduct similar experiments as those in Fig. 2 on small-scale downstream datasets. The results are shown in Fig. 3. We observe consistent performance improvement as the number of reserved pre-trained models' blocks increases. And the smaller the dataset scale, the more the performance benefits from the higher layers. It demonstrates that higher layers are still valuable and matter in data-insufficient downstream tasks. Furthermore, we observe comparable performance for the transfer performance of MAE-Tiny and MoCov3-Tiny when only a certain number of lower layers are reserved, while MoCov3-Tiny surpasses when higher layers are further reserved. It indicates that the higher layers of MoCov3-Tiny work better than MAE-Tiny on data-insufficient downstream tasks, which is also consistent with our CKA-based analyses shown in Fig. 1, that MoCov3-Tiny learns more semantics at an abstract level relevant to recognition in higher layers (high similarity to reference recognition models in higher layers) than MAE-Tiny.

4.2. Attention Map Analyses

The attention maps reveal the behaviors for aggregating information in the attention mechanism of ViTs, which are computed from the compatibility of queries and keys by dot-product operation. We introduce two metrics for further analyses on the pre-trained models, *i.e.*, *attention distance* and *attention entropy*. The attention distance for the j-th token of h-th head is calculated as:

$$\boldsymbol{D}_{h,j} = \sum_{i} \mathrm{softmax}(\boldsymbol{A}_h)_{i,j} \boldsymbol{G}_{i,j}, \tag{1}$$

where $\boldsymbol{A}_h \in \mathbb{R}^{l \times l}$ is the attention map for the h-th attention head, and $\boldsymbol{G}_{i,j}$ is the Euclidean distance between the spatial locations of the i-th and j-th tokens. l is the number of

[1] https://github.com/AntixK/PyTorch-Model-Compare

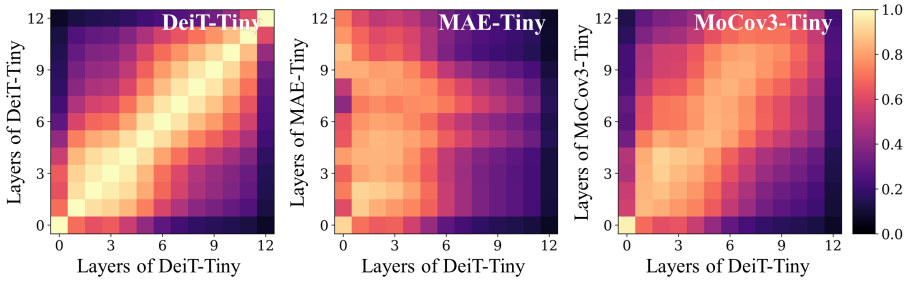

Figure 1. **Layer representation similarity** within and across models as heatmaps, with x and y axes indexing the layers (the 0 index indicates the patch embedding layer), and higher values indicate higher similarity.

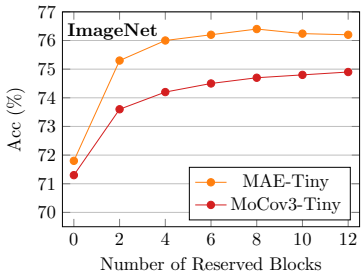

Figure 2. Lower layers of pre-trained models contribute to most gains on downstream ImageNet dataset.

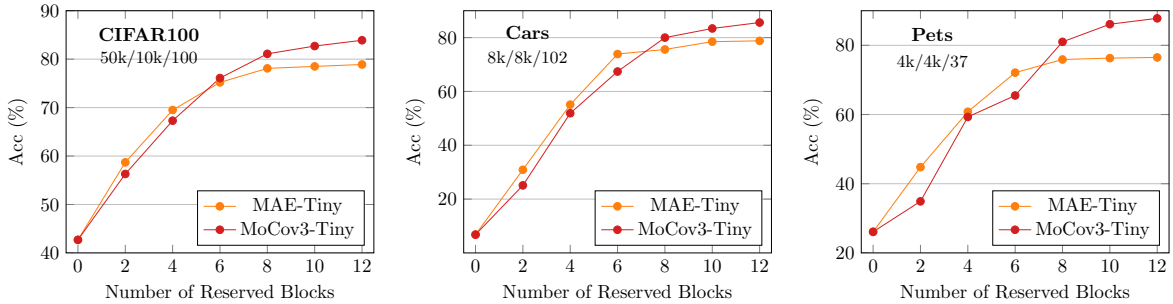

Figure 3. The contributions on performance gain from higher layers of pre-trained models increase as the downstream dataset scale shrinks, which indicates that higher layers matter in data-insufficient downstream tasks.

tokens. And the attention entropy is calculated as:

$$\boldsymbol{E}_{h,j} = -\sum_{i} \text{softmax}(\boldsymbol{A}_h)_{i,j} \log(\text{softmax}(\boldsymbol{A}_h)_{i,j}), \quad (2)$$

Specifically, the attention distance reveals how much local *vs*. global information is aggregated, and a lower distance indicates that each token focuses more on neighbor tokens. The attention entropy reveals the concentration of the attention distribution, and lower entropy indicates that each token attends to fewer tokens. We analyze the distributions of the average attention distance and entropy across all the tokens in different attention heads, as shown in Fig. 4.

The pre-training with MAE makes the attention of the downstream models more local and concentrated. First, we compare MAE-Tiny-FT with DeiT-Tiny. The former adopts MAE-Tiny as initialization and then is fine-tuned on IN1K, and the latter is supervisedly trained from scratch (Random Init.) on IN1K. As shown in Fig. 4, we observe very similar attention behaviors between them, except that the attention of MAE-Tiny-FT (the purple box-whisker) is more local (with lower attention distance) and concentrated (with lower attention entropy) in middle layers compared with DeiT-Tiny (the red box-whisker). We attribute it to the introduction of the MAE-Tiny as pre-training (the orange box-whisker), which has lower attention distance and entropy, and may bring locality inductive bias compared with random initialization (the blue box-whisker). It is noteworthy that the locality inductive bias does not mean that tokens in all attention heads attend to solely a few nearby tokens. The attention distance and entropy for different heads are still distributed in a wide range (except for several last layers), which indicates that the heads have diverse specializations, making the models aggregate both local and global tokens with both concentrated and broad focuses.

Then, we focus on the comparison between MAE-Tiny and MoCov3-Tiny, trying to give some explanations for their diverse downstream performances observed in Sec. 3. As shown in Fig. 4, we observe that MoCov3-Tiny (the green box-whisker) generally has more global and broad attention than MAE-Tiny (the orange box-whisker). Even several leading blocks have a narrower range of attention distance and entropy than MAE-Tiny. We think this characteristic of MoCov3-Tiny makes the downstream fine-tuning with it as initialization take "shortcuts", *i.e*., directly paying attention to global features and overlooking local patterns, which may be unfavorable for fine-grained recognition. It leads to inferior downstream performance on ImageNet, but fair on Flowers, CIFAR100, *etc*., for which the "shortcuts" may be barely adequate. As for MAE-Tiny, its distinct behaviors in higher layers with rather low attention distance and entropy may make it hard to transfer to data-insufficient downstream tasks, thus resulting in inferior performance on these tasks.

5. Distillation Improves Pre-Trained Models

In the previous section, we have conjectured that it is hard for MAE to learn good representation relevant to recognition

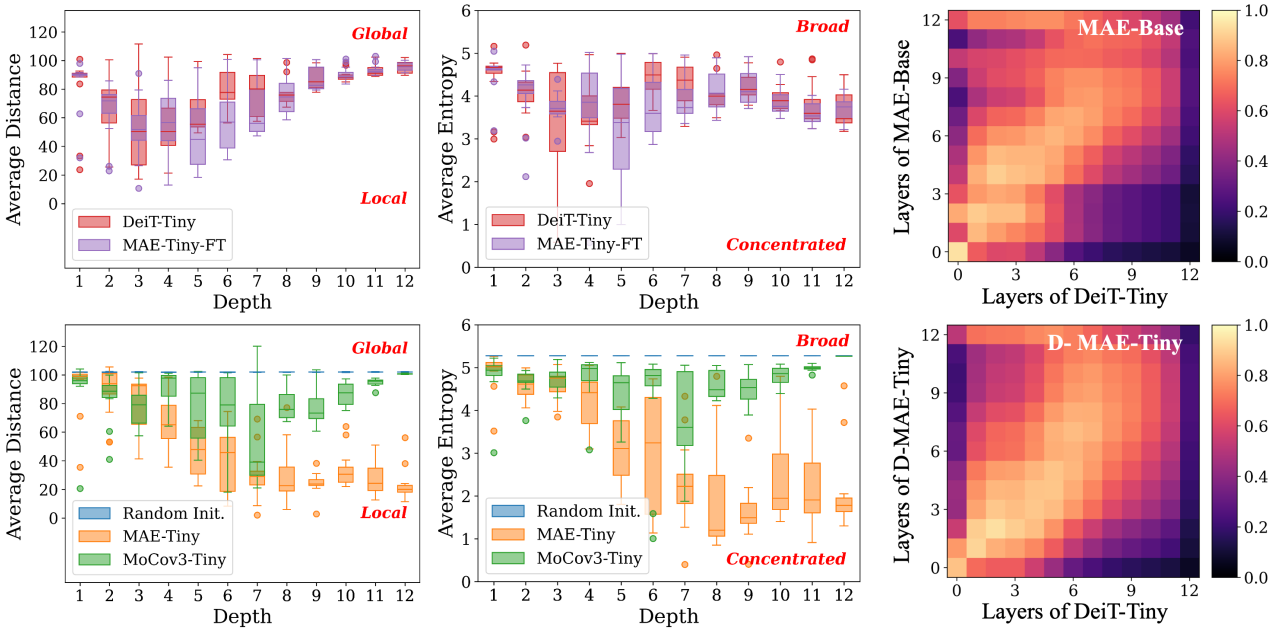

Figure 4. **Attention distance and entropy analyses**. We visualize the distributions of the average attention distance and entropy across all tokens in different attention heads w.r.t. the layer number with box-whisker plots.

Figure 5. Distillation compresses the good representation of the teacher (MAE-Base) to the student (D-MAE-Tiny).

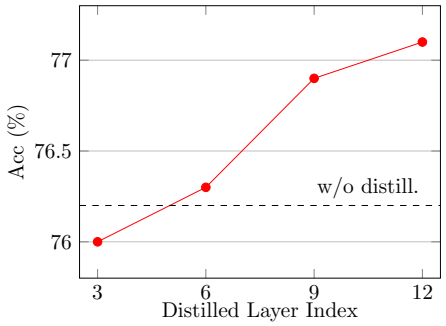

Figure 6. Distillation on attention maps of higher layers improves performance most.

in higher layers, which results in unsatisfactory performance on data-insufficient downstream tasks. A natural question is that can it gain more semantic information by scaling up the models. We further examine a large pre-trained model, MAE-Base (He et al., 2021), and find it achieves a better alignment with the reference model, as shown in the top subfigure of Fig. 5. It indicates that *it is possible to extract features relevant to recognition in higher layers for the scaled-up encoder in MAE pre-training.* These observations motivate us to compress the knowledge of large pre-trained models to tiny ones with knowledge distillation under the MIM framework.

Distillation methods. Specifically, a pre-trained MAE-Base (He et al., 2021) is introduced as the teacher network. The distillation loss is constructed based on the similarity between the attention maps of the corresponding teacher's and student's layers. It is formulated as:

$$L_{\text{attn}} = \text{MSE}(\boldsymbol{A}^T, \boldsymbol{M}\boldsymbol{A}^S), \tag{3}$$

where $\boldsymbol{A}^T \in \mathbb{R}^{h\times l\times l}$ and $\boldsymbol{A}^S \in \mathbb{R}^{h'\times l\times l}$ refer to the attention maps of the corresponding teacher's and student's layers, with h and h' attention heads respectively. l is the number of tokens. A learnable mapping matrix $\boldsymbol{M} \in \mathbb{R}^{h\times h'}$ is introduced to align the number of heads. MSE denotes mean squared error.

During the pre-training, the teacher processes the same unmasked image patches as the student encoder. The parameters of the student network are updated based on the joint backward gradients from the distillation loss and the original MAE's reconstruction loss, while the teacher's parameters remain frozen throughout the pre-training process.

Distill on lower or higher layers? We first examine applying the above layer-wise distillation on which pair of teacher's and student's layers contributes to the most performance gain. We conduct experiments by constructing the above attention-based distillation loss between pair of layers at $1/4$, $2/4$, $3/4$, or $4/4$ depth of the teacher and student respectively, *i.e.*, the 3rd, 6th, 9th, or 12th layer for both the teacher (MAE-Base) and the student (MAE-Tiny). As shown in Fig. 6, distilling on the attention maps of the last transformer blocks promotes the performance most, surpassing those distilling on lower layers (for the sake of simplicity, we only fine-tune the pre-trained models on IN1K for 100 epochs). It is consistent with the analyses

Table 5. **Distillation improves downstream performance** on classification tasks and object detection and segmentation tasks. Top-1 accuracy is reported for classification tasks and AP is reported for object detection (det.) and instance segmentation (seg.) tasks.

Init. \ Datasets	Flowers	Pets	Aircraft	Cars	CIFAR100	iNat18	ImageNet	COCO(det.)	COCO(seg.)
supervised									
DeiT-Tiny	**96.4**	**93.1**	73.5	85.6	**85.8**	**63.6**	-	40.4	35.5
self-supervised									
MAE-Tiny	85.8	76.5	64.6	78.8	78.9	60.6	78.0	39.9	35.4
D-MAE-Tiny	95.2 (+9.4)	89.1 (+12.6)	**79.2** (+14.6)	**87.5** (+8.7)	85.0 (+6.1)	**63.6** (+3.0)	**78.4** (+0.4)	**42.3** (+2.4)	**37.4** (+2.0)

in Sec. 4. Specifically, the lower layers learn good representation themselves during the pre-training with MAE, and thus distilling on these layers contributes to marginal improvement, while the higher layers rely on a good teacher to guide them to capture rich semantic features.

Distillation improves downstream performance. We further evaluate the distilled pre-trained model on several downstream tasks. For simplicity, we only apply distillation on the last layers. The resulting model is denoted as D-MAE-Tiny. The visualization result at the bottom of Fig. 5 shows that the good representation relevant to the recognition of the teacher is compressed to the student. Especially the quality of higher layers is improved. The distillation contributes to better downstream performance as shown in Tab. 5, especially on data-insufficient classification tasks and dense prediction tasks. In Appendix C.3, we also show that our distillation technique can help other ViT students beyond ViT-Tiny to achieve better downstream performance.

6. Related Works

Self-supervised learning (SSL) focuses on different pretext tasks (Gidaris et al., 2018; Zhang et al., 2016; Noroozi & Favaro, 2016; Dosovitskiy et al., 2014) for pre-training without using manually labeled data. Among them, contrastive learning (CL) has been popular and shows promising results on various convolutional networks (ConvNets) (He et al., 2020; Chen et al., 2020; Grill et al., 2020; Caron et al., 2020) and ViTs (Chen et al., 2021a; Caron et al., 2021). Recently, methods based on masked image modeling (MIM) achieve state-of-the-art on ViTs (He et al., 2021; Bao et al., 2021; Zhou et al., 2022). It has been demonstrated that these methods can scale up well on larger models, while their performance on lightweight ViTs is seldom investigated.

Vision Transformers (ViTs) (Dosovitskiy et al., 2020) apply a Transformer architecture (a stack of attention modules (Vaswani et al., 2017)) on image patches and show very competitive results in various visual tasks (Touvron et al., 2021a; Liu et al., 2021; Li et al., 2022). The performance of ViTs has been largely improved thanks to better training recipes (Touvron et al., 2021a; Steiner et al., 2021; Touvron et al., 2022). As for lightweight ViTs, most works focus on integrating ViTs and ConvNets (Graham et al., 2021; Heo et al., 2021; Mehta & Rastegari, 2022; Chen et al., 2021b), while few works focus on how to optimize the networks.

Knowledge Distillation is a mainstream approach for model compression (Buciluǎ et al., 2006), in which a large teacher network is trained first and then a more compact student network is optimized to approximate the teacher (Hinton et al., 2015; Romero et al., 2014). Touvron et al. (2021a) achieves better accuracy on ViTs by adopting a ConvNet as the teacher. With regard to the compression of the pre-trained networks, some works (Sanh et al., 2019; Jiao et al., 2020; Wang et al., 2021; Sun et al., 2020) attend to distill large-scale pre-trained language models. In the area of computer vision, a series of works (Fang et al., 2020; Abbasi Koohpayegani et al., 2020; Choi et al., 2021) focus on transferring knowledge of large pre-trained networks based on CL to lightweight ConvNets. There are few works focusing on improving the quality of lightweight pre-trained ViTs based on MIM by distillation thus far.

7. Discussions

Limitations. Our study is restricted to classification tasks and some dense-prediction tasks. We leave the exploration of more tasks for further work.

Conclusions. We investigate the self-supervised pre-training of lightweight ViTs, and demonstrate the usefulness of the advanced lightweight ViT pre-training strategy in improving the performance of downstream tasks, even comparable to most delicately-designed SOTA networks on ImageNet. Some properties about the pre-training are revealed, *e.g.*, these methods fail to benefit from large-scale pre-training data, and show more dependency on the downstream dataset scale. We also present some insights on what matters for downstream performance. They may indicate potential future directions in improving pre-training on lightweight models, the value of which has also been demonstrated as it guides the design of our proposed distillation strategy and helps to achieve much better downstream performance. We expect our research may provide useful experience and advance the study of self-supervised learning on lightweight ViTs.

Acknowledgment. The authors would like to thank the anonymous reviewers for their valuable comments and suggestions. This work was supported in part by the National Key R&D Program of China (Grant No. 2020AAA0105802, 2020AAA0105800), the Natural Science Foundation of China (Grant No. U22B2056, 61972394, U2033210, 62036011, 62192782, 61721004, 62172413), the Beijing Natural Science Foundation (Grant No. L223003, JQ22014), the Major Projects of Guangdong Education Department for Foundation Research and Applied Research (Grant No. 2017KZDXM081, 2018KZDXM066), the Guangdong Provincial University Innovation Team Project (Grant No. 2020KCXTD045), the Zhejiang Provincial Natural Science Foundation (Grant No. LDT23F02024F02). Jin Gao was also supported in part by the Youth Innovation Promotion Association, CAS.

References

Abbasi Koohpayegani, S., Tejankar, A., and Pirsiavash, H. Compress: Self-supervised learning by compressing representations. *Adv. Neural Inform. Process. Syst.*, 33:12980–12992, 2020.

Ali, A., Touvron, H., Caron, M., Bojanowski, P., Douze, M., Joulin, A., Laptev, I., Neverova, N., Synnaeve, G., Verbeek, J., et al. Xcit: Cross-covariance image transformers. *Adv. Neural Inform. Process. Syst.*, 34, 2021.

Assran, M., Caron, M., Misra, I., Bojanowski, P., Joulin, A., Ballas, N., and Rabbat, M. Semi-supervised learning of visual features by non-parametrically predicting view assignments with support samples. In *Int. Conf. Comput. Vis.*, pp. 8443–8452, 2021.

Ba, J. L., Kiros, J. R., and Hinton, G. E. Layer normalization. *ArXiv*, abs/1607.06450, 2016.

Bao, H., Dong, L., and Wei, F. Beit: Bert pre-training of image transformers. *ArXiv*, abs/2106.08254, 2021.

Buciluǎ, C., Caruana, R., and Niculescu-Mizil, A. Model compression. In *ACM Int. Conf. on Knowledge Discovery and Data Mining*, pp. 535–541, 2006.

Caron, M., Misra, I., Mairal, J., Goyal, P., Bojanowski, P., and Joulin, A. Unsupervised learning of visual features by contrasting cluster assignments. In *Adv. Neural Inform. Process. Syst.*, 2020.

Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., and Joulin, A. Emerging properties in self-supervised vision transformers. In *Int. Conf. Comput. Vis.*, pp. 9650–9660, 2021.

Chen, X., Fan, H., Girshick, R., and He, K. Improved baselines with momentum contrastive learning. *ArXiv*, abs/2003.04297, 2020.

Chen, X., Xie, S., and He, K. An empirical study of training self-supervised vision transformers. In *Int. Conf. Comput. Vis.*, pp. 9640–9649, 2021a.

Chen, Y., Dai, X., Chen, D., Liu, M., Dong, X., Yuan, L., and Liu, Z. Mobile-former: Bridging mobilenet and transformer. *ArXiv*, abs/2108.05895, 2021b.

Cho, J. H. and Hariharan, B. On the efficacy of knowledge distillation. In *Int. Conf. Comput. Vis.*, pp. 4794–4802, 2019.

Choi, H. M., Kang, H., and Oh, D. Unsupervised representation transfer for small networks: I believe i can distill on-the-fly. In *Adv. Neural Inform. Process. Syst.*, 2021.

Cortes, C., Mohri, M., and Rostamizadeh, A. Algorithms for learning kernels based on centered alignment. *The Journal of Machine Learning Research*, 13:795–828, 2012.

Cubuk, E. D., Zoph, B., Shlens, J., and Le, Q. Randaugment: Practical automated data augmentation with a reduced search space. In *Adv. Neural Inform. Process. Syst.*, volume 33, pp. 18613–18624, 2020.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pp. 248–255, 2009.

Dosovitskiy, A., Springenberg, J. T., Riedmiller, M., and Brox, T. Discriminative unsupervised feature learning with convolutional neural networks. *Adv. Neural Inform. Process. Syst.*, 27:766–774, 2014.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *Int. Conf. Learn. Represent.*, 2020.

Fang, Z., Wang, J., Wang, L., Zhang, L., Yang, Y., and Liu, Z. Seed: Self-supervised distillation for visual representation. In *Int. Conf. Learn. Represent.*, 2020.

Gidaris, S., Singh, P., and Komodakis, N. Unsupervised representation learning by predicting image rotations. In *Int. Conf. Learn. Represent.*, 2018.

Goyal, P., Dollár, P., Girshick, R., Noordhuis, P., Wesolowski, L., Kyrola, A., Tulloch, A., Jia, Y., and He, K. Accurate, large minibatch sgd: Training imagenet in 1 hour. *ArXiv*, abs/1706.02677, 2017.

Graham, B., El-Nouby, A., Touvron, H., Stock, P., Joulin, A., Jégou, H., and Douze, M. Levit: A vision transformer in convnet's clothing for faster inference. In *Int. Conf. Comput. Vis.*, pp. 12259–12269, 2021.

Grill, J.-B., Strub, F., Altché, F., Tallec, C., Richemond, P., Buchatskaya, E., Doersch, C., Pires, B., Guo, Z., Azar, M., et al. Bootstrap your own latent: A new approach to self-supervised learning. In *Adv. Neural Inform. Process. Syst.*, 2020.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pp. 770–778, 2016.

He, K., Fan, H., Wu, Y., Xie, S., and Girshick, R. Momentum contrast for unsupervised visual representation learning. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pp. 9729–9738, 2020.

He, K., Chen, X., Xie, S., Li, Y., Dollár, P., and Girshick, R. Masked autoencoders are scalable vision learners. *ArXiv*, abs/2111.06377, 2021.

Heo, B., Yun, S., Han, D., Chun, S., Choe, J., and Oh, S. J. Rethinking spatial dimensions of vision transformers. In *Int. Conf. Comput. Vis.*, pp. 11936–11945, 2021.

Hinton, G., Vinyals, O., and Dean, J. Distilling the knowledge in a neural network. *ArXiv*, abs/1503.02531, 2015.

Howard, A., Sandler, M., Chu, G., Chen, L.-C., Chen, B., Tan, M., Wang, W., Zhu, Y., Pang, R., Vasudevan, V., Le, Q. V., and Adam, H. Searching for mobilenetv3. In *Int. Conf. Comput. Vis.*, 2019.

Huang, G., Sun, Y., Liu, Z., Sedra, D., and Weinberger, K. Q. Deep networks with stochastic depth. In *Eur. Conf. Comput. Vis.*, pp. 646–661, 2016.

Jiao, X., Yin, Y., Shang, L., Jiang, X., Chen, X., Li, L., Wang, F., and Liu, Q. Tinybert: Distilling bert for natural language understanding. In *Findings of Empirical Methods in Natural Language Process.*, pp. 4163–4174, 2020.

Jin, X., Peng, B., Wu, Y., Liu, Y., Liu, J., Liang, D., Yan, J., and Hu, X. Knowledge distillation via route constrained optimization. In *Int. Conf. Comput. Vis.*, pp. 1345–1354, 2019.

Krause, J., Stark, M., Deng, J., and Fei-Fei, L. 3d object representations for fine-grained categorization. In *Int. Conf. Comput. Vis. Worksh.*, pp. 554–561, 2013.

Krizhevsky, A. et al. Learning multiple layers of features from tiny images. *Technical Report*, 2009.

Li, Y., Xie, S., Chen, X., Dollar, P., He, K., and Girshick, R. Benchmarking detection transfer learning with vision transformers. *ArXiv*, abs/2111.11429, 2021.

Li, Y., Mao, H., Girshick, R., and He, K. Exploring plain vision transformer backbones for object detection. *ArXiv*, abs/2203.16527, 2022.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. Microsoft coco: Common objects in context. In *Eur. Conf. Comput. Vis.*, pp. 740–755, 2014.

Liu, Z., Miao, Z., Zhan, X., Wang, J., Gong, B., and Yu, S. X. Large-scale long-tailed recognition in an open world. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019.

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., and Guo, B. Swin transformer: Hierarchical vision transformer using shifted windows. In *Int. Conf. Comput. Vis.*, pp. 10012–10022, 2021.

Liu, Z., Mao, H., Wu, C.-Y., Feichtenhofer, C., Darrell, T., and Xie, S. A convnet for the 2020s. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pp. 11966–11976, 2022.

Loshchilov, I. and Hutter, F. Sgdr: Stochastic gradient descent with warm restarts. *ArXiv*, abs/1608.03983, 2016.

Maji, S., Rahtu, E., Kannala, J., Blaschko, M., and Vedaldi, A. Fine-grained visual classification of aircraft. *ArXiv*, abs/1306.5151, 2013.

Mehta, S. and Rastegari, M. Mobilevit: Light-weight, general-purpose, and mobile-friendly vision transformer. In *Int. Conf. Learn. Represent.*, 2022.

Mirzadeh, S. I., Farajtabar, M., Li, A., Levine, N., Matsukawa, A., and Ghasemzadeh, H. Improved knowledge distillation via teacher assistant. In *AAAI Conf. on Artificial Intelligence*, volume 34, pp. 5191–5198, 2020.

Newell, A. and Deng, J. How useful is self-supervised pretraining for visual tasks? In *IEEE Conf. Comput. Vis. Pattern Recog.*, pp. 7345–7354, 2020.

Nguyen, T., Raghu, M., and Kornblith, S. Do wide and deep networks learn the same things? uncovering how neural network representations vary with width and depth. In *Int. Conf. Learn. Represent.*, 2020.

Nilsback, M.-E. and Zisserman, A. Automated flower classification over a large number of classes. In *Indian Conference on Computer Vision, Graphics & Image Processing*, pp. 722–729, 2008.

Noroozi, M. and Favaro, P. Unsupervised learning of visual representations by solving jigsaw puzzles. In *Eur. Conf. Comput. Vis.*, pp. 69–84, 2016.

Pan, J., Bulat, A., Tan, F., Zhu, X., Dudziak, L., Li, H., Tzimiropoulos, G., and Martinez, B. Edgevits: Competing light-weight cnns on mobile devices with vision transformers. *ArXiv*, abs/2205.0343, 2022.

Parkhi, O. M., Vedaldi, A., Zisserman, A., and Jawahar, C. V. Cats and dogs. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pp. 3498–3505, 2012.

Raghu, M., Unterthiner, T., Kornblith, S., Zhang, C., and Dosovitskiy, A. Do vision transformers see like convolutional neural networks? *Adv. Neural Inform. Process. Syst.*, 34, 2021.

Ridnik, T., Ben-Baruch, E., Noy, A., and Zelnik-Manor, L. Imagenet-21k pretraining for the masses. *ArXiv*, abs/2104.10972, 2021.

Romero, A., Ballas, N., Kahou, S. E., Chassang, A., Gatta, C., and Bengio, Y. Fitnets: Hints for thin deep nets. *ArXiv*, abs/1412.6550, 2014.

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., and Chen, L.-C. Mobilenetv2: Inverted residuals and linear bottlenecks. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pp. 4510–4520, 2018.

Sanh, V., Debut, L., Chaumond, J., and Wolf, T. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *ArXiv*, abs/1910.01108, 2019.

Song, L., Smola, A., Gretton, A., Bedo, J., and Borgwardt, K. Feature selection via dependence maximization. *The Journal of Machine Learning Research*, 13(5), 2012.

Steiner, A., Kolesnikov, A., Zhai, X., Wightman, R., Uszkoreit, J., and Beyer, L. How to train your vit? data, augmentation, and regularization in vision transformers. *ArXiv*, abs/2106.10270, 2021.

Sun, Z., Yu, H., Song, X., Liu, R., Yang, Y., and Zhou, D. Mobilebert: a compact task-agnostic bert for resource-limited devices. In *Association for Computational Linguistics*, pp. 2158–2170, 2020.

Tan, M. and Le, Q. Efficientnet: Rethinking model scaling for convolutional neural networks. In *Int. Conf. Machine Learning.*, pp. 6105–6114, 2019.

Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., and Jegou, H. Training data-efficient image transformers & distillation through attention. In *Int. Conf. Machine Learning.*, volume 139, pp. 10347–10357, 2021a.

Touvron, H., Cord, M., Sablayrolles, A., Synnaeve, G., and Jégou, H. Going deeper with image transformers. In *Int. Conf. Comput. Vis.*, pp. 32–42, 2021b.

Touvron, H., Cord, M., and Jégou, H. Deit iii: Revenge of the vit. *ArXiv*, abs/2204.07118, 2022.

Van Horn, G., Mac Aodha, O., Song, Y., Cui, Y., Sun, C., Shepard, A., Adam, H., Perona, P., and Belongie, S. The inaturalist species classification and detection dataset. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. *Adv. Neural Inform. Process. Syst.*, 30, 2017.

Wang, W., Bao, H., Huang, S., Dong, L., and Wei, F. Minilmv2: Multi-head self-attention relation distillation for compressing pretrained transformers. In *Findings of Int. Joint Conf. on Natural Language Process.*, pp. 2140–2151, 2021.

Wightman, R. Pytorch image models. `https://github.com/rwightman/pytorch-image-models`, 2019.

Wightman, R., Touvron, H., and Jégou, H. Resnet strikes back: An improved training procedure in timm. *ArXiv*, abs/2110.00476, 2021.

Woo, S., Debnath, S., Hu, R., Chen, X., Liu, Z., Kweon, I.-S., and Xie, S. Convnext v2: Co-designing and scaling convnets with masked autoencoders. *ArXiv*, abs/2301.00808, 2023.

Xie, Z., Zhang, Z., Cao, Y., Lin, Y., Bao, J., Yao, Z., Dai, Q., and Hu, H. Simmim: A simple framework for masked image modeling. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2022.

Yun, S., Han, D., Oh, S. J., Chun, S., Choe, J., and Yoo, Y. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Int. Conf. Comput. Vis.*, pp. 6023–6032, 2019.

Zhang, H., Cisse, M., Dauphin, Y. N., and Lopez-Paz, D. mixup: Beyond empirical risk minimization. In *Int. Conf. Learn. Represent.*, 2018.

Zhang, R., Isola, P., and Efros, A. A. Colorful image colorization. In *Eur. Conf. Comput. Vis.*, pp. 649–666, 2016.

Zhou, J., Wei, C., Wang, H., Shen, W., Xie, C., Yuille, A., and Kong, T. ibot: Image bert pre-training with online tokenizer. *Int. Conf. Learn. Represent.*, 2022.

A. Experimental Details

A.1. Evaluation Details for MAE and MoCo-v3 on ImageNet

We follow the common practice of supervised ViT training (Touvron et al., 2021a) for fine-tuning evaluation except for some hyper-parameters of augmentation. The default setting is in Tab. A1. We use the linear *lr* scaling rule (Goyal et al., 2017): *lr* = base *lr*×batchsize / 256. We use layer-wise *lr* decay following (Bao et al., 2021; He et al., 2021), and the decay rate is tuned respectively for MAE and MoCo-v3.

Besides, we use global average pooling (GAP) after the final block during the fine-tuning of both the MAE and MoCo-v3-based pre-trained models, which is, however, not the common practice for MoCo-v3 (Chen et al., 2021a). We adopt it as it significantly helps to surpass the model using the original configuration based on a class token (76.8% *vs*. 73.7% top-1 accuracy) for the lightweight ViT-Tiny.

Table A1. **Fine-tuning evaluation settings**.

config	value
optimizer	AdamW
base learning rate	1e-3
weight decay	0.05
optimizer momentum	$\beta_1, \beta_2 = 0.9, 0.999$
layer-wise *lr* decay (Bao et al., 2021)	0.85 (MAE), 0.75 (MoCo-v3)
batch size	1024
learning rate schedule	cosine decay (Loshchilov & Hutter, 2016)
warmup epochs	5
training epochs	{100, 300, 1000}
augmentation	RandAug(10, 0.5) (Cubuk et al., 2020)
colorjitter	0.3
label smoothing	0
mixup (Zhang et al., 2018)	0.2
cutmix (Yun et al., 2019)	0
drop path (Huang et al., 2016)	0

Table A2. **Pre-training setting for MoCo-v3**.

config	value
optimizer	AdamW
base learning rate	1.5e-4
weight decay	0.1
optimizer momentum	$\beta_1, \beta_2 = 0.9, 0.999$
batch size	1024
learning rate schedule	cosine decay
warmup epochs	40
training epochs	400
momentum coefficient	0.99
temperature	0.2

A.2. Pre-Training Details of MAE

Our experimental setup on MAE largely follows those of MAE (He et al., 2021), including the optimizer, learning rate, batch size, argumentation, *etc*. But several basic factors and components are adjusted to fit the smaller encoder. We find MAE prefers a much more lightweight decoder when the encoder is small, thus a decoder with only one Transformer block is adopted by default and the width is 192. We sweep over 5 masking ratios {0.45, 0.55, 0.65, 0.75, 0.85} and find 0.75 achieves the best performance.

A.3. Pre-Training Details of MoCo-v3

We reimplement MoCo-v3 (Chen et al., 2021a) with ViT-Tiny as encoder and largely follow the original setups. The default setting is in Tab. A2.

Chen et al. (2021a) observes that instability is a major issue that impacts self-supervised ViT training and causes mild degradation in accuracy, and a simple trick by adopting fixed random patch projection (the first layer of a ViT model) is proposed to improve stability in practice. However, we find that stability is not the main issue for small networks. Higher performance is achieved with a learned patch projection layer. Thus, this technique is not used by default.

A.4. Transfer Evaluation Details on Classification Tasks

We evaluate several pre-trained models with transfer learning in order to measure the generalization ability of these models. We use 6 popular vision datasets: Flowers-102 (Flowers for short) (Nilsback & Zisserman, 2008), Oxford-IIIT Pets (Pets) (Parkhi et al., 2012), FGVC-Aircraft (Aircraft) (Maji et al., 2013), Stanford Cars (Cars) (Krause et al., 2013), Cifar100 (Krizhevsky et al., 2009), iNaturalist 2018 (iNat18) (Van Horn et al., 2018). For all these datasets except iNat18, we fine-tune with SGD (momentum=0.9), and the batch size is set to 512. The learning rates are swept over 3 candidates and the training epochs are swept over 2 candidates per dataset as detailed in Tab. A3. We adopt a cosine decay learning rate schedule (Loshchilov & Hutter, 2016) with a linear warm-up. we resize images to 224 $\times$ 224. We adopt random resized crop and random horizontal flipping as augmentations and do not use any regularization (*e.g*., weight decay, dropout, or the stochastic

Table A3. **Transfer evaluation details.**

Dataset	Learning rate	Total epochs and warm-up epochs	layer-wise *lr* decay
Flowers	{0.01, 0.03, 0.1}	{(150,30),(250,50)}	{1.0, 0.75}
Pets	{0.01, 0.03, 0.1}	{(70,14),(150,30)}	{1.0, 0.75}
Aircraft	{0.01, 0.03, 0.1}	{(50,10),(100,20)}	{1.0, 0.75}
Cars	{0.01, 0.03, 0.1}	{(50,10),(100,20)}	{1.0, 0.75}
CIFAR100	{0.03, 0.1, 0.3}	{(25, 5),(50,10)}	{1.0, 0.75}

depth regularization technique (Huang et al., 2016)). For iNat18, we follow the same training configurations as those on ImageNet.

A.5. Transfer Evaluation Details on Dense Prediction Tasks

We reproduce the setup in (Li et al., 2021), except for replacing the backbone with ViT-Tiny and decreasing the input image size from 1024 to 640 to make it trainable on a single machine with 8 NVIDIA V100. We fine-tune for up to 100 epochs on COCO (Lin et al., 2014), with different pre-trained models as initialization of the backbone. We do not use layer-wise *lr* decay since we find it useless for the tiny backbone on the detection tasks. The weight decay is 0.05 and the stochastic depth regularization (Huang et al., 2016) is not used.

A.6. Analysis Methods

We adopt the Centered Kernel Alignment (CKA) metric to analyze the representation similarity (S_{rep}) within and across networks. Specifically, CKA takes two feature maps (or representations) $\boldsymbol{X}$ and $\boldsymbol{Y}$ as input and computes their normalized similarity in terms of the Hilbert-Schmidt Independence Criterion (HSIC) as

$$S_{rep}(\boldsymbol{X}, \boldsymbol{Y}) = \mathrm{CKA}(\boldsymbol{K}, \boldsymbol{L}) = \frac{\mathrm{HSIC}(\boldsymbol{K}, \boldsymbol{L})}{\sqrt{\mathrm{HSIC}(\boldsymbol{K}, \boldsymbol{K})\mathrm{HSIC}(\boldsymbol{L}, \boldsymbol{L})}}, \tag{A1}$$

where $\boldsymbol{K} = \boldsymbol{X}\boldsymbol{X}^{\mathrm{T}}$ and $\boldsymbol{L} = \boldsymbol{Y}\boldsymbol{Y}^{\mathrm{T}}$ denote the Gram matrices for the two feature maps. A minibatch version is adopted by using an unbiased estimator of HSIC (Nguyen et al., 2020) to work at scale with our networks. We select the normalized version of the output representation of each Transformer block (consisting of a multi-head self-attention (MHA) block and an MLP block). Specifically, we select the feature map after the first LayerNorm (LN) (Ba et al., 2016) of the next block as the representation of this Transformer block as depicted in Fig. A1.

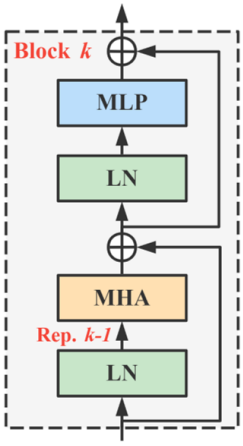

Figure A1. **Transformer block.**

B. More Analyses on the Pre-Training

B.1. Analyses with More Models as Reference

In Sec. 4, the analyses are mainly conducted by adopting the supervisedly trained DeiT-Tiny as the reference model. Here, we additionally introduce stronger recognition models as references to demonstrate the generalizability of our analyses. Specifically, we use ViT-Base models trained with various recipes as references, *e.g.*, DeiT-Base (supervisedly trained on IN1K following Touvron et al. (2021a) and achieves 82.0% top-1 accuracy on ImageNet), ViT-Base-21k (supervisedly trained on IN21K following Steiner et al. (2021)), ViT-Base-21k-1k (first pre-trained on IN21K and then fine-tuned on IN1K following Steiner et al. (2021), achieving 84.5% top-1 accuracy on ImageNet). The layer representation similarity is presented in Fig. A2.

First, we observe that our default reference model, DeiT-Tiny, is aligned well with these larger models (as shown in the left column of Fig. A2). We conjecture that the supervisedly trained ViTs generally have similar layer representation structures. Based on these stronger reference models, we observe similar phenomena for MAE-Tiny and MoCov3-Tiny as discussed in Sec. 4, which demonstrates the robustness of our analyses and conclusions w.r.t. different reference models.

Then, we analyze the larger MAE-Base with these newly introduced models as references, as shown in the last column of Fig. A2. We observe that MAE-Base still aligns relatively well with these much stronger recognition models, which supports

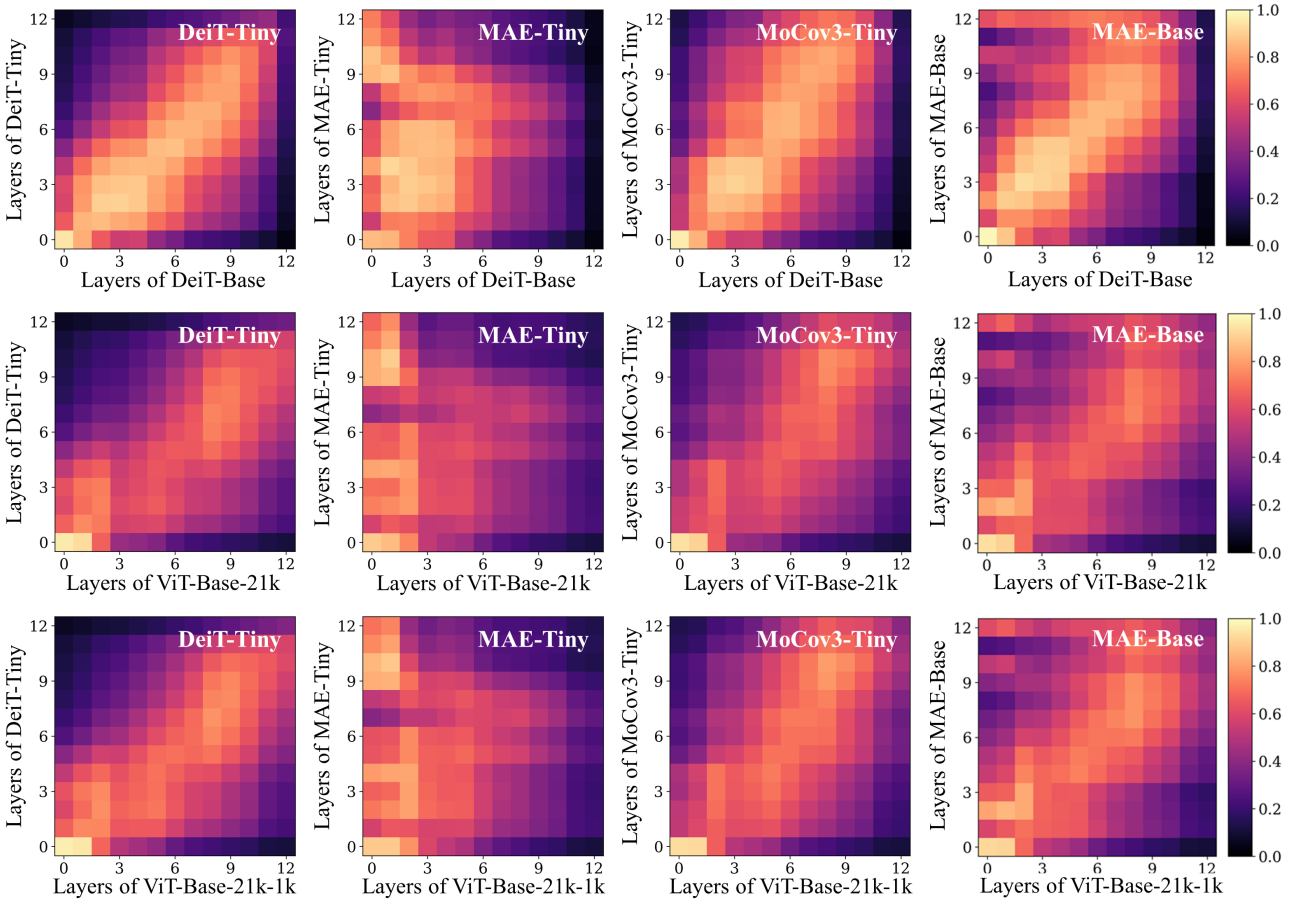

Figure A2. **Layer representation analyses** with DeiT-Base (supervisedly trained on IN1K, the top row), ViT-Base-21k (supervisedly trained on IN21K, the middle row), and ViT-Base-21k-1k (supervisedly pre-trained on IN21K and fine-tuned on IN1K, the bottom row) as the reference models.

our claim in Sec. 5 that *it is possible to extract features relevant to recognition in higher layers for the scaled-up encoder in MAE pre-training*. It is the prerequisite for the improvement of the pre-trained models from the proposed distillation.

B.2. Analyses Based on Linear Probing Evaluation

Our analyses are mainly based on the *fine-tuning* evaluation. In this section, we present some experimental results based on *linear probing* evaluation, in which only a classifier is tuned during the downstream training while the pre-trained representations are kept frozen. It reflects how the representations obtained by the pre-trained models are linearly separable w.r.t. semantic categories.

As shown in Tab. A4, the *linear probing* performance is consistently lower than the *fine-tuning* performance. Coupled with the case that *linear probing* does not save much training time for evaluating lightweight models, it is not a proper way to utilize the pre-trained models compared to the *fine-tuning* setting.

Furthermore, the *linear probing* evaluation results do not reflect fine-tuned performance according to Tab. A4 and Tab. 4, especially for those downstream tasks with relatively sufficient labeled data, *e.g.*, iNat18, ImageNet, thus may lead to an underestimation of the value of some pre-trained models in the practical utility on downstream tasks. We attribute it to that *linear probing* only evaluates the final representation of the pre-trained models, which makes it overlook the value of providing good initialization for lower layers. For instance, MAE-Tiny is better at it than MoCov3-Tiny.

Additionally, the inferior *linear probing* results of MAE-Tiny to MoCov3-Tiny also support our analyses in Sec. 4.1 that MoCov3-Tiny learns more semantics at an abstract level relevant to recognition in higher layers than MAE-Tiny. But our proposed distillation technique can improve the results to a certain extent.

Table A4. **Linear probing evaluation** of pre-trained models on downstream classification tasks. Top-1 accuracy is reported.

Init. \ Datasets	Flowers	Pets	Aircraft	Cars	CIFAR100	iNat18	ImageNet
supervised							
DeiT-Tiny	91.0	**92.0**	41.2	**47.9**	**73.6**	**39.8**	-
self-supervised							
MoCov3-Tiny	**93.2**	83.5	**44.8**	44.5	73.4	36.2	**62.1**
MAE-Tiny	48.9	25.0	12.8	8.8	31.0	1.4	23.3
D-MAE-Tiny	77.1	55.5	20.1	16.4	58.4	10.7	42.0

Table A5. **Comparisons on more pre-training methods**. It is a revised version of Tab. 1 in the main paper with more self-supervised pre-training methods.

Methods	Pre-training Data	Epochs	Time (hour)	Fine-tuning recipe	Top-1 Acc. (%)
from scratch	-	-	-	ori.	74.5
from scratch	-	-	-	impr.	75.8
Supervised (Steiner et al., 2021)	IN21K w/ labels	30	20	impr.	76.9
Supervised (Steiner et al., 2021)	IN21K w/ labels	300	200	impr.	77.8
MoCo-v3 (Chen et al., 2021a)	IN1K w/o labels	400	52	impr.	76.8
MAE (He et al., 2021)	IN1K w/o labels	400	23	impr.	78.0
DINO (Caron et al., 2021)	IN1K w/o labels	400	83	impr.	77.2
SimMIM (Xie et al., 2022)	IN1K w/o labels	400	40	impr.	77.9
D-MAE-Tiny (ours)	IN1K w/o labels	400	26	impr.	**78.4**

Table A6. **Transfer evaluation on classification tasks and dense-prediction tasks for more pre-training methods**. It is a revised version of Tab. 4 in the main paper with more self-supervised pre-training methods.

Init. \ Datasets	Flowers (2k/6k/102)	Pets (4k/4k/37)	Aircraft (7k/3k/100)	Cars (8k/8k/196)	CIFAR100 (50k/10k/100)	iNat18 (438k/24k/8142)	COCO(det.) (118k/50k/80)	COCO(seg.) (118k/50k/80)
supervised								
DeiT-Tiny	**96.4**	**93.1**	73.5	85.6	**85.8**	63.6	40.4	35.5
self-supervised								
MoCov3-Tiny	94.8	87.8	73.7	83.9	83.9	54.5	39.7	35.1
MAE-Tiny	85.8	76.5	64.6	78.8	78.9	60.6	39.9	35.4
DINO-Tiny	95.6	89.3	73.6	84.5	84.7	58.7	41.4	36.7
SimMIM-Tiny	77.2	68.9	55.9	70.4	77.7	60.8	39.3	34.8
D-MAE-Tiny (ours)	95.2	89.1	**79.2**	**87.5**	85.0	**63.6**	**42.3**	**37.4**

B.3. Analyses for More Self-Supervised Pre-Training Methods

In the main paper, our analyses mainly focus on MAE (He et al., 2021) and MoCov3 (Chen et al., 2021a). In this section, more self-supervised pre-training methods are involved. Specifically, another MIM-based method, SimMIM (Xie et al., 2022), and another CL-based method, DINO (Caron et al., 2021), are evaluated based on the lightweight ViT-Tiny. The 400-epoch pre-trained models are denoted as SimMIM-Tiny and DINO-Tiny respectively.

We first evaluate their downstream performance on ImageNet and other classification tasks, and object detection and segmentation tasks, as shown in Tab. A5 and Tab. A6. They are also revised versions of Tab. 1 and Tab. 4 in the main paper. According to the results, we find that MIM-based methods are generally superior to CL-based methods on data-sufficient tasks, *e.g.*, ImageNet and iNat18, while inferior on data-insufficient tasks. Downstream data scale matters for all these methods and none of them achieve consistent superiority on all downstream tasks.

Then we explore the layer representation of these models by CKA-based similarity analyses, as shown in Fig. A3. We observe similar layer representation structures for both MIM family and CL family. For instance, SimMIM-Tiny also learns poor semantics on higher layers.

Finally, we carry out the attention analyses for these models, as shown in Fig. A4. We also observe consistent properties for MIM family and CL family. SimMIM-Tiny also tends to focus on local patterns with concentrated attention in higher layers like MAE-Tiny, while DINO-Tiny behaves like MoCov3-Tiny and has broad and global attention in higher layers.

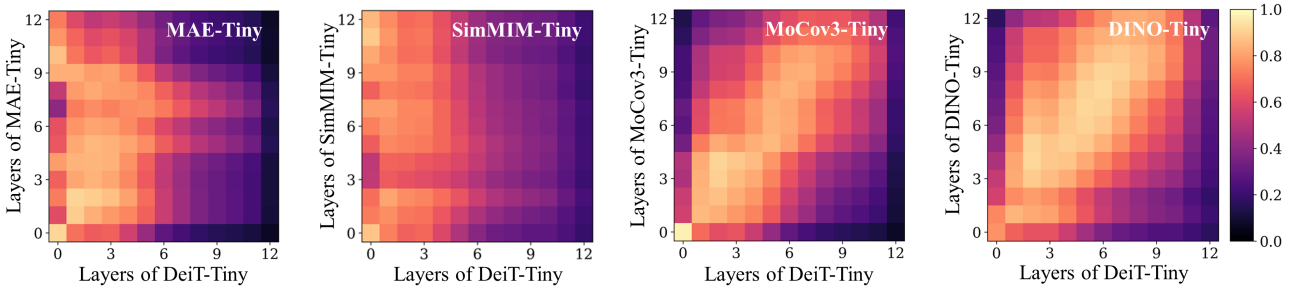

Figure A3. **Layer representation analyses** for more self-supervised pre-trained models.

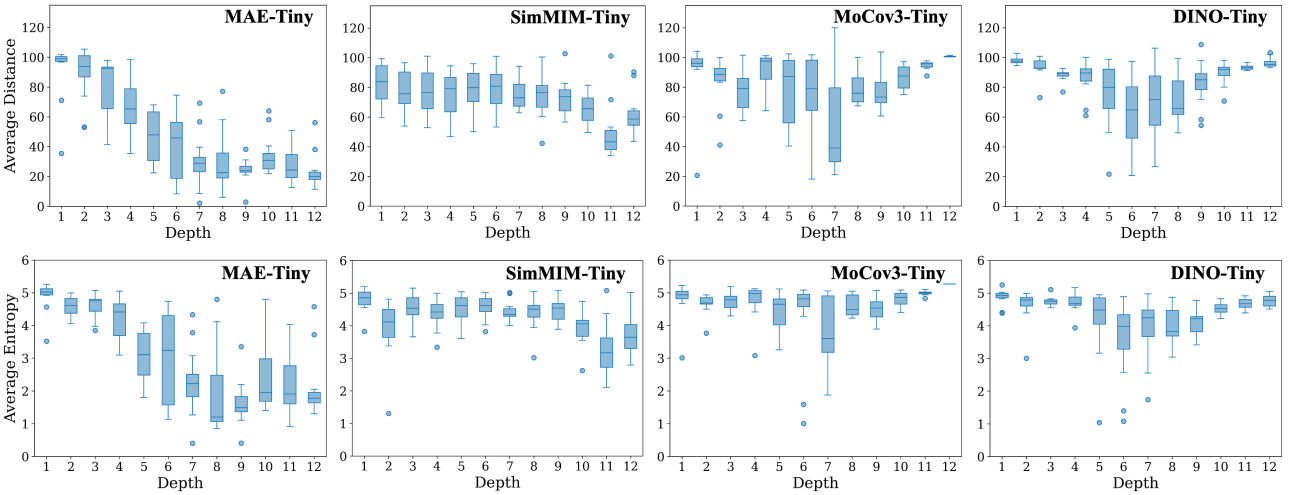

Figure A4. **Attention analyses** for more self-supervised pre-trained models.

C. More Analyses on Distillation

C.1. Illustration of the Distillation Process

We illustrate our distillation process in Fig. A5 for a better presentation and explanation.

Based on the mask auto-encoder, we introduce a teacher ViT, which is pre-trained with MAE. During pre-training, the teacher processes the same visible image patches as the student encoder, and the attention-based distillation loss is calculated between the attention maps of the corresponding teacher's and student's layers. The parameters of the student are updated based on the joint backward gradients from the distillation loss and the original MAE's reconstruction loss, while the teacher's parameters remain frozen throughout the pre-training process.

C.2. Attention Map Analyses for the Distilled Pre-trained Models

we analyze the attention distance and entropy of the distilled MAE-Tiny introduced in Sec. 5 (D-MAE-Tiny), which is only applied distillation on the attention map of the last layer during the pre-training with MAE. As shown in Fig. A6, we observe more global and broad attention in the higher layers of D-MAE-Tiny compared with MAE-Tiny, which behaves more like the teacher, MAE-Base. We attribute it to that the distillation on the final layer (*i.e.*, the 12th layer) forces the distilled layer of the student to imitate the teacher's attention and also requires the several preceding layers to make changes to meet the imitation. We reckon that it may be useful to capture semantic features and improve downstream performance.

We also find the attention distance of the last layer shows more diversity: some attention heads are rather global and the others are local, and all of them are concentrated. We reckon that it shows odd behaviors for the reason that the layer can not handle both training targets from the reconstruction task and distillation restricted to the model size. But the more plentiful supervision indeed improves the quality of previous layers and thus achieves better downstream performance.

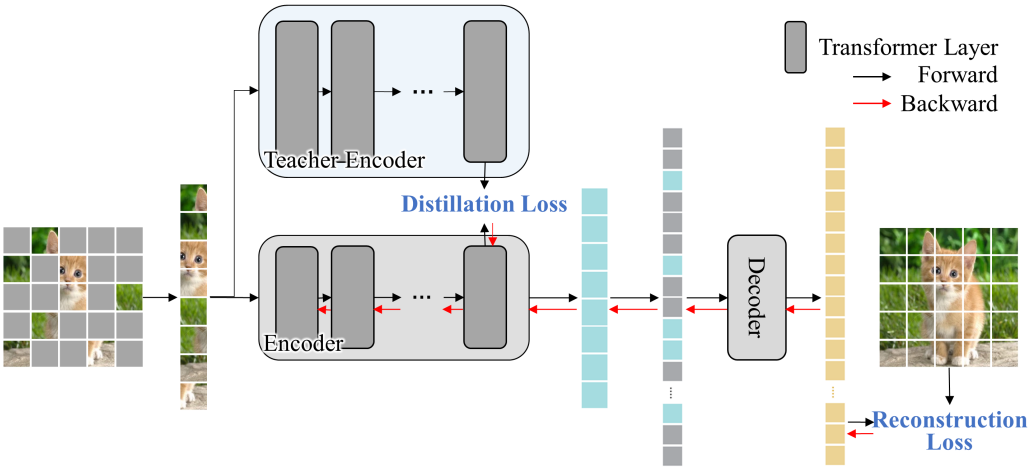

Figure A5. **Illustration of the distillation process.**

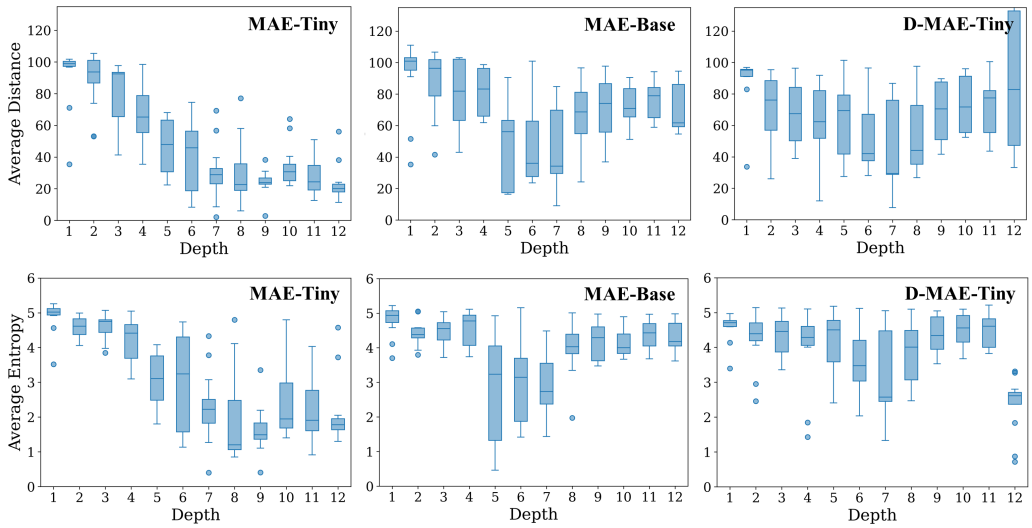

Figure A6. **Attention distance and entropy analyses** for the distilled MAE-Tiny.

C.3. Applying Distillation on More Networks

To further evaluate our proposed distillation method, we additionally apply it to the pre-training of ViT-Small also with MAE-Base as the teacher. The configurations of these models are presented in Tab. A7. The transfer evaluation results are presented in Tab. A8. The transfer performance of the distilled MAE-Small (D-MAE-Small) surpasses the baseline model, MAE-Small by a large margin, which shows the efficacy of the distillation.

C.4. Distilling with Larger Teachers

We further conduct additional experiments with various models as teachers and compared their performance on various downstream tasks (see Tab. A9). The configurations of the student model (ViT-Tiny) and teacher models are presented in Tab. A7. The results indicate that *an appropriately sized teacher model provides the most improvement gains in distillation*, which is a common finding in the area of knowledge distillation (Cho & Hariharan, 2019; Jin et al., 2019; Mirzadeh et al., 2020). To further investigate the impact of teacher size, we conducted CKA-based layer representation analyses of these teachers, as shown in Fig. A7. It can be seen that a teacher that is too small (MAE-Small) also suffers from degraded representation on higher layers and can not provide sufficient knowledge, while a teacher that is too large (MAE-Large) would result in a mismatch of capacity with the tiny student model, considering it has over 50 times more parameters than ViT-Tiny with different depths and attention head numbers, which leads to a little distinct learned pattern compared to the reference tiny model, and may not be suitable for the student.

Table A7. **Configurations of ViTs**.

Model	channel dimension	#heads	#layers	#params
ViT-Tiny	192	12	12	6M
ViT-Small	384	12‡	12	22M
ViT-Base	768	12	12	86M
ViT-Large	1024	16	24	304M

‡ Our ViT-Small is with heads=12 following Chen et al. (2021a).

Table A8. **Distillation on MAE-Small**. Top-1 accuracy for the transfer performance on downstream classification tasks of pre-trained models w. or w/o. distillation is reported.

Init. \ Datasets	Flowers	Pets	Aircraft	Cars	CIFAR100	iNat18	ImageNet
supervised							
DeiT-Small	**97.4**	**94.2**	77.6	88.2	**89.2**	66.5	80.2
self-supervised							
MAE-Small	91.2	82.0	65.8	79.2	80.8	63.2	82.1
D-MAE-Small	95.8 (+4.6)	91.4 (+9.4)	**80.7** (+14.9)	**88.3** (+9.1)	87.8 (+7.0)	**66.9** (+3.7)	**82.5** (+0.4)

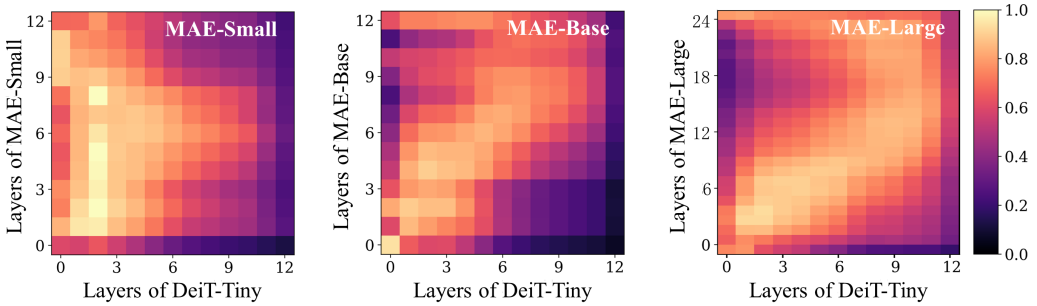

Figure A7. **Layer representation analyses** of the teachers for distillation.

Table A9. **Distillation with different sized teachers**. Top-1 accuracy for the transfer performance on downstream classification tasks of the distilled pre-trained models is reported.

Pre-training		Fine-tuning						
Student	Teacher	Flowers	Pets	Aircraft	Cars	CIFAR100	iNat18	ImageNet
MAE-Tiny	-	85.8	76.5	64.6	78.8	78.9	60.6	78.0
MAE-Tiny	MAE-Small	89.4	78.6	65.2	78.9	79.6	61.5	78.1
MAE-Tiny	MAE-Base	**95.2**	**89.1**	**79.2**	**87.5**	**85.0**	**63.6**	**78.4**
MAE-Tiny	MAE-Large	94.0	87.3	77.1	85.2	84.2	63.1	78.3