Supplementary Materials:
A Closer Look at Self-supervised Lightweight Vision Transformers

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-lite and MoCo-v3. For the fine-tuning evaluation without layer-wise *lr* decay, we decrease the base learning rate to 2.5e-4. Besides, we use global average pooling (GAP) for MAE-lite after the final block during fine-tuning, while using a class token for MoCo-v3 following their original practices.

Our linear probing evaluation follows (Chen et al., 2021a). Tab. A1 also summaries its details. We adopt an extra BatchNorm layer (Ioffe & Szegedy, 2015) without affine transformation between the output of the pre-trained encoder and the linear classifier following (He et al., 2021). Besides, the class token is used for both methods in linear probing evaluation.

Table A1: **Fine-tuning and linear probing evaluation settings**.

config	value (fine-tuning)	value (linear probing)
optimizer	AdamW	AdamW
base learning rate	1e-3	0.1
weight decay	0.05	0
optimizer momentum	$\beta_1, \beta_2 = 0.9, 0.999$	$\beta_1, \beta_2 = 0.9, 0.999$
layer-wise lr decay (Bao et al., 2021)	0.85 (MAE), 0.75 (MoCo-v3)	-
batch size	1024	4096
learning rate schedule	cosine decay (Loshchilov & Hutter, 2016)	cosine decay (Loshchilov & Hutter, 2016)
warmup epochs	5	10
training epochs	{100, 300, 1000}	90
augmentation	RandAug(10, 0.5) (Cubuk et al., 2020)	RandomResizedCrop
colorjitter	0.3	0
label smoothing	0	0
mixup (Zhang et al., 2018)	0.2	0
cutmix (Yun et al., 2019)	0	0
drop path (Huang et al., 2016)	0	0

A.2 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 best performance.

A.3 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. We adopt fine-tuning and linear probing evaluation on ImageNet (see Appendix A.1).

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 in both fine-tuning and linear probing evaluation with a learned patch projection layer. Besides, we observe that no matter whether this first layer is random or learned, it always reduces a large amount of high-frequency signals in the ultimate pre-trained models, as shown in Fig. A1.

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

config	value (fine-tuning)
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	10
training epochs	400
momentum coefficient	0.99
temperature	0.2

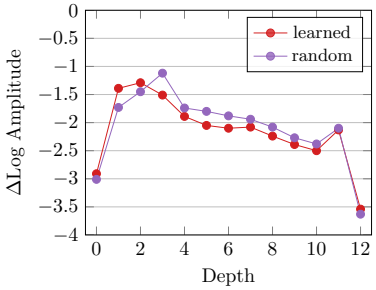


Figure A1: **Fourier analysis** for MoCo-v3 with learned or random patch projection.

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 and the momentum and batch size are set to 0.9 and 512 respectively. 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 with a linear warm-up. we resize images to 224×224. 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 depth regularization technique (Huang et al., 2016)). For iNat18, we follow the same training configurations as those on ImageNet.

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}

A.5 Transfer Evaluation Details on Dense Prediction Tasks

We reproduce the setup in (Li et al., 2021), except replacing the backbone with ViT-Tiny and decreasing the input image size from 1024 to 768 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.

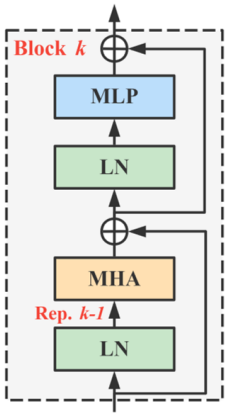


Figure A2: **Transformer block.**

A.6 Analysis Methods

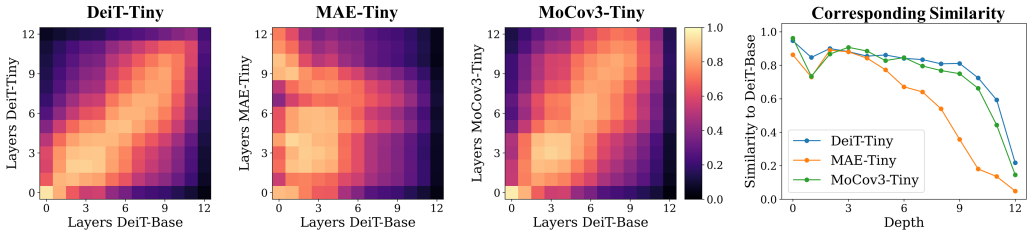
Representation similarity. We adopt the Centered Kernel Alignment (CKA) metric to analyze the representation similarity (S_{rep}) within and across networks. Specifically, CKA takes two feature maps (or representations) $\boldsymbol{X}$ and $\boldsymbol{Y}$ as input and computes their normalized similarity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

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

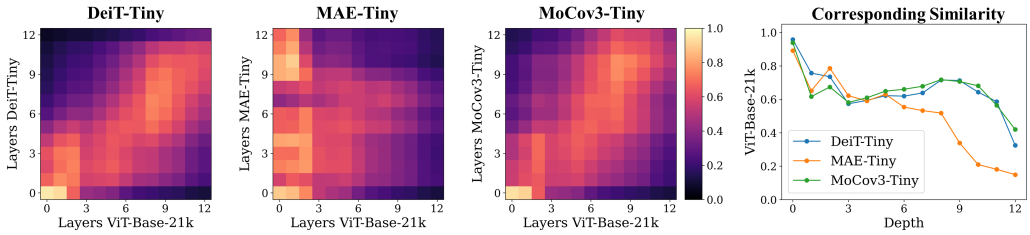
where $\boldsymbol{K} = \boldsymbol{X}\boldsymbol{X}^{\mathrm{T}}$ and $\boldsymbol{L} = \boldsymbol{Y}\boldsymbol{Y}^{\mathrm{T}}$ denote the Gram matrices for the two feature maps. A mini-batch version is adopted by using an unbiased estimator of HSIC (Nguyen et al., 2020) to work at scale with our networks. For the sake of conciseness, we only select the representation after 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) as the representation of the last Transformer block as depicted in Fig. A2.

B More Analyses on the Pre-training

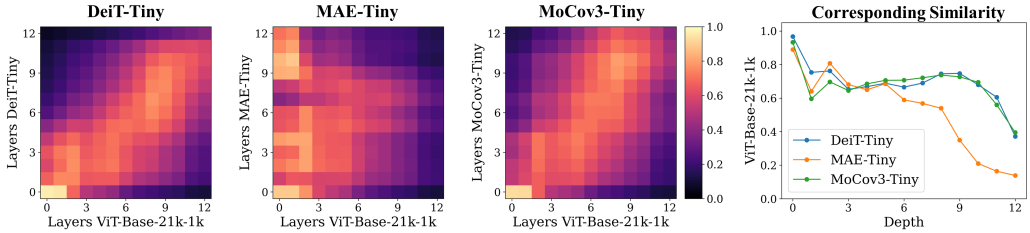
B.1 Analyses with More Models as Reference



(a) DeiT-Base (supervised trained on IN1K) as the reference model.



(b) ViT-Base-21k (supervised trained on IN21K) as the reference model.



(c) ViT-Base-21k-1k (supervised pre-trained on IN21K and fine-tuned on IN1K) as the reference model.

Figure A3: **Layer attention and representation similarity analyses** with more reference models.

In Sec. 4, the analyses are mainly conducted by adopting the supervised trained DeiT-Tiny as the reference model. Here, we additionally introduce more 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 (supervised trained on IN1K following Touvron et al. (2021a) and achieves 82.0 top1 accuracy on ImageNet), ViT-Base-21k (supervised 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 top1 accuracy on ImageNet). The layer representation similarity are presented in Fig. A3.

First, we observe that DeiT-Tiny is aligned well with these larger models (as shown in the left column of Fig. A3). We conjecture that the supervised trained ViTs generally have similar layer representation structures. Based on these stronger reference models, we observe similar phenomena 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 Fig. A4. We observe that MAE-Base still aligns relatively well to these much stronger recognition models, which supports 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.

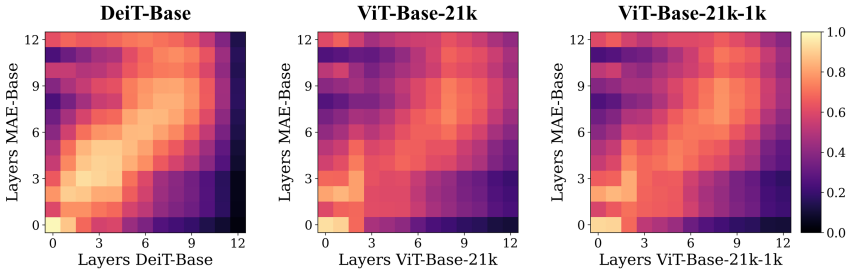


Figure A4: **Similarity analyses for MAE-Base** with more reference models.

B.2 Distilling with Larger Teachers

We further distill our lightweight pre-trained models with a larger teacher than our used MAE-Base, *i.e.*, MAE-Large (He et al., 2021). However, we find the large pre-trained model MAE-Large is not a good teacher though it achieves superior fine-tuning performance on ImageNet, as shown in Tab. A4.

Table A4: **Distilling with larger pre-trained teachers**. We report the achieved accuracy after fine-tuning on ImageNet. Top-1 accuracy for the teacher and top-1/5 accuracy for the student are presented.

Teacher Model	Teacher Top-1	Student Top-1	Student Top-5
-	-	76.2	93.1
MAE-Base He et al. (2021)	83.6	**77.1** (+0.9)	**93.5** (+0.4)
MAE-Large He et al. (2021)	**85.9**	76.7 (+0.5)	93.4 (+0.3)

B.3 Attention Map Analyses for the Distilled Pre-trained Models

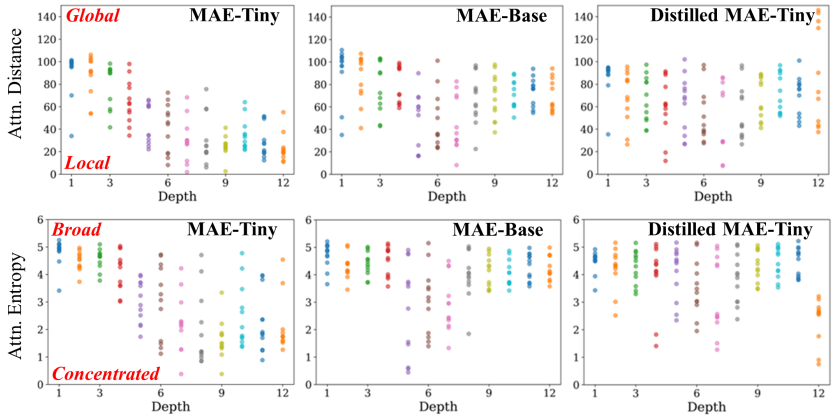


Figure A5: **Attention distance and entropy analyses for the distilled MAE-Tiny.**

we analyze the attention distance and entropy of the distilled MAE-Tiny introduced in Sec. 5, which is only applied distillation on the attention map of the last layer during the pre-training with MAE. As shown in Fig. A5, we observe more global and broad attention in the higher layers of the distilled MAE-Tiny compared with MAE-Tiny, which behaves more like the teacher, MAE-Base. 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.

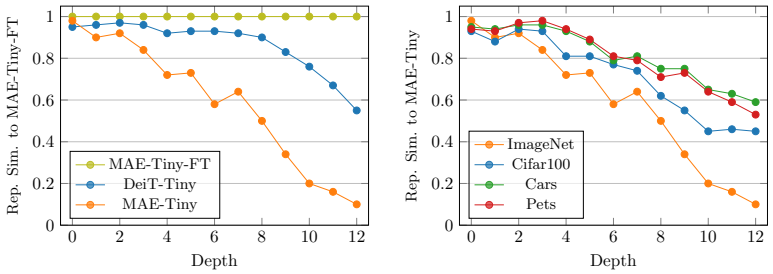


Figure A6: Analyses on the layer behaviors during the fine-tuning based on the CKA similarity.

B.4 Analyses on the Layer Behaviors During the Fine-tuning

In this section, the analyses on the layer behaviors during the fine-tuning of the pre-trained model, MAE-Tiny is present. First, we focus on the fine-tuning experiments on the IN1K, which is considered to have comparatively sufficient training data. We analyze the CKA representation similarity of corresponding layers between MAE-Tiny and MAE-Tiny-FT as shown in Fig. A6. We find higher layers are largely changed, which is in accordance with the analysis in Sec. 4.1, that the higher layers of MAE-Tiny are less relevant to recognition and thus can be further enhanced by fine-tuning as in MAE-Tiny-FT. Then, we compare MAE-Tiny-FT with DeiT-Tiny and find their difference mainly lies in the higher layers. We conjecture that by initializing the lower layers properly with MAE-Tiny, the model may be more concentrated on the training of higher layers during the fine-tuning phase, and can achieve better results if training data is sufficient.

Then, we focus on the fine-tuning process of MAE-Tiny on small-scale datasets. We compare the fine-tuned models on various downstream datasets with the pre-trained model, MAE-Tiny. The larger the downstream data scale, the more the higher layers change. We conjecture that the higher layers require more data to learn high-quality representation, considering the pre-training only provides initialization for higher layers with poor semantics for recognition. Thus, it may be hard for MAE-Tiny to adapt the higher layers to the data-insufficient downstream tasks, and then resulting in inferior transfer results for MAE-Tiny as shown in Tab. 4.

B.5 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. After pre-training, we obtain the pre-trained models 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 version 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.

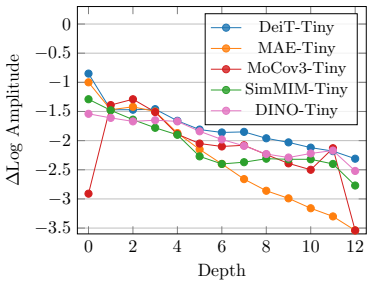


Figure A7: **Fourier analysis** for more pre-trained models.

Then we explore the layer representation of these models by CKA-based similarity analysis and Fourier analysis, as shown in Fig. A8 and Fig. A7. We observe similar layer representation structures for both MIM family and CL family. For instance, SimMIM-Tiny also learns poor semantics on higher layers. As for the Fourier analysis, we conjecture the multi-crop strategy used in DINO improves its frequency appearance, which forces the model to focus on some details.

Finally, we carry out the attention analyses for these models, as shown in Fig. A9. We also observe consistent properties for MIM family and CL family. SimMIM-Tiny also tends to focus on local

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	Fine-tuning 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.	73.7
MAE (He et al., 2021)	IN1K w/o labels	400	23	impr.	**78.0**
DINO Caron et al. (2021)	IN1K w/o labels	300	62	impr.	76.7
SimMIM Xie et al. (2022)	IN1K w/o labels	400	40	impr.	77.9

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**	**40.7**	**36.5**
self-supervised								
MoCov3-Tiny	94.8	87.8	**73.7**	83.9	83.9	54.5	40.0	36.0
MAE-Tiny	85.8	76.5	64.6	78.8	78.9	60.6	38.9	35.1
DINO-Tiny	95.6	89.3	73.6	84.5	84.7	58.7	40.2	36.1
SimMIM-Tiny	77.2	68.9	55.9	70.4	77.7	60.8	39.1	35.2

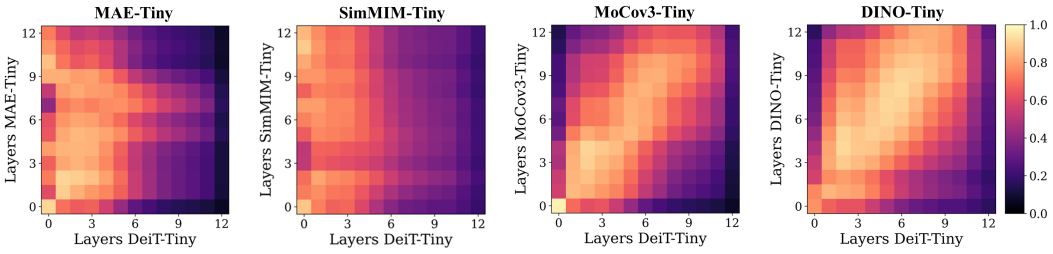


Figure A8: **Layer representation similarity analyses for more self-supervised pre-trained models.**

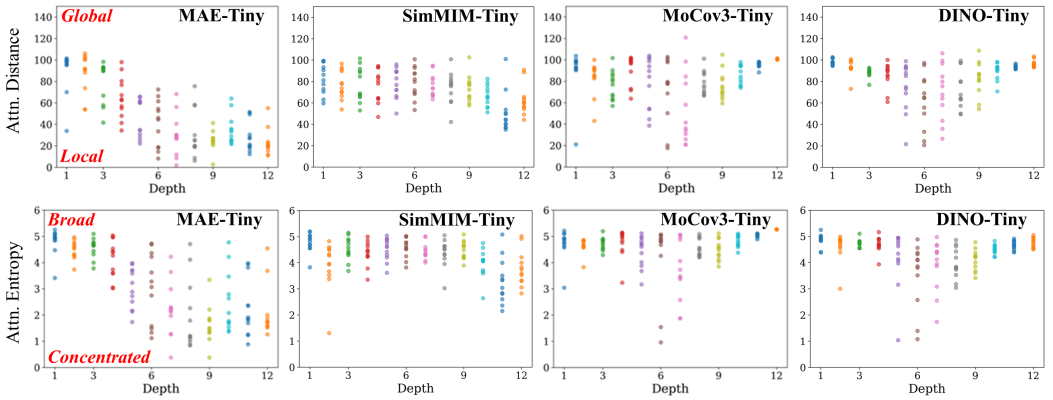


Figure A9: **Attention analyses for more self-supervised pre-trained models.**

pattern 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.