TinyViT: Fast Pretraining Distillation for Small Vision Transformers

Kan Wu[1,3,*], Jinnian Zhang[2,4,*], Houwen Peng[3,*,†], Mengchen Liu[4], Bin Xiao[4], Jianlong Fu[3], Lu Yuan[4]

[1] Sun Yat-sen University, [2] University of Wisconsin-Madison, [3] Microsoft Research, [4] Microsoft Cloud+AI

Abstract. Vision transformer (ViT) recently has drawn great attention in computer vision due to its remarkable model capability. However, most prevailing ViT models suffer from huge number of parameters, restricting their applicability on devices with limited resources. To alleviate this issue, we propose TinyViT, a new family of tiny and efficient small vision transformers pretrained on large-scale datasets with our proposed fast distillation framework. The central idea is to transfer knowledge from large pretrained models to small ones, while enabling small models to get the dividends of massive pretraining data. More specifically, we apply distillation during pretraining for knowledge transfer. The logits of large teacher models are sparsified and stored in disk in advance to save the memory cost and computation overheads. The tiny student transformers are automatically scaled down from a large pretrained model with computation and parameter constraints. Comprehensive experiments demonstrate the efficacy of TinyViT. It achieves a top-1 accuracy of 84.8% on ImageNet-1k with only 21M parameters, being comparable to Swin-B pretrained on ImageNet-21k while using 4.2 times fewer parameters. Moreover, increasing image resolutions, TinyViT can reach 86.5% accuracy, being slightly better than Swin-L while using only 11% parameters. Last but not the least, we demonstrate a good transfer ability of TinyViT on various downstream tasks. Code and models are available at here.

Keywords: Pretraining, Knowledge Distillation, Small Transformer

1 Introduction

Transformer [62] has taken computer vision domain by storm and are becoming increasingly popular in both research and practice [20,8,75]. One of the recent trends for vision transforms (ViT) is to continue to grow in model size while yielding improved performance on standard benchmarks [75,41,54]. For example, V-MoE [54] uses 305 million images to train an extremely large model with 14.7 billion parameters, achieving state-of-the-art performance on image classification. Meanwhile, the Swin transformer uses 3 billion parameters with 70

*Equal contribution. Work done when Kan and Jinnian were interns of Microsoft.
†Corresponding author: houwen.peng@microsoft.com.

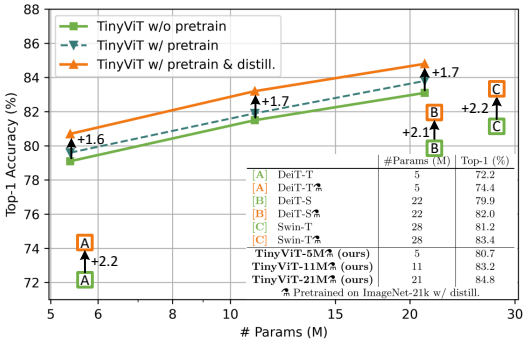


Fig. 1: Comparison of our TinyViT with other small vision transformer models [61,42] on ImageNet-1k in terms of w/ and w/o ImageNet-21k pretraining and distillation. Pretraining with distillation can effectively improve the performance of all these small transformer models, further unveiling their capacities. Best viewed in color.

million pretraining images, to attain promising results on downstream detection and segmentation tasks [42,41]. Such large model sizes and the accompanying heavy pretraining costs make these models unsuitable for applications involving limited computational budgets, such as mobile and IoT edge devices [77].

In contrast to scaling up models to large scales, this work turns attention to downsizing vision transformers, aiming to generate a new family of tiny models and elevate their transfer capacities in downstream tasks. In particular, we explore the following key issue: *how to effectively transfer the knowledge of existing large-scale transformers to small ones, as well as unleash the power of large-scale data to elevate the representability of small models?* In computer vision, it has long been recognized that large models pretrained on large datasets often achieve better results, while small models easily become saturated (or underfitting) as the growth of data [75,41]. Is there any possible way for small models to absorb knowledge from massive data and further unveil their capacities?

To answer this question, we introduce a fast knowledge distillation method to pretrain small models, and show that small models can also get the dividends of massive pretraining data with the guidance of large models. More specifically, we observe that direct pretraining of small models suffers from performance saturation, especially when the data scale increases. But if we impose distillation during pretraining, using a powerful model as the teacher, the potentials of large-scale pretraining data can be unlocked for small models, as demonstrated in Fig. 1. Meanwhile, the distilled small models can be transferred well to downstream tasks, since they have learned a great deal of knowledge about how to generalize from the large model as well as the large-scale pretraining data. We give a detailed discussion in Sec. 4 exploring the underlying reasons why pretraining distillation is able to further unveil the capacities of small models.

Pretraining models with distillation is inefficient and costly, because a considerable proportion of computing resources is consumed on passing training data through the large teacher model in each iteration, rather than training the target small student. Also, a giant teacher may occupy the most GPU memory, significantly slowing down the training speed of the students (due to limited batch size). To address this issue, we propose a fast and scalable distillation strategy. More concretely, we propose to generate a sparse probability vector as the soft

label of each input image in advance, and store it into label files together with the corresponding data augmentation information like random cropping, RandAugment [17], CutMix [73], *etc.* During training, we reuse the stored sparse soft labels and augmentations to precisely replicate the distillation procedure, successfully omitting the forward computation and storage of large teacher models. Such strategy has two advantages: 1) Fast. It largely saves the memory cost and computation overheads of generating teachers' soft labels during training. Thus, the distillation of small models can be largely speed up because it is able to use much larger batch size. Besides, since the teacher logits per epoch are independent, they can be saved in parallel, instead of epoch-by-epoch in conventional methods. 2) Scalable. It can mimic any kind of data augmentation and generate the corresponding soft labels. We just need to forward the large teacher model for only once, and reuse the soft labels for arbitrary student models.

We verify the efficacy of our fast pretraining distillation framework not only on existing small vision transformers, such as DeiT-T [61] and Swin-T [42], but also over our new designed tiny architectures. Specifically, following [21], we adopt a progressive model contraction approach to scale down a large model and generate a family of tiny vision transformers (TinyViT). With our fast pretraining distillation on ImageNet-21k [18], TinyViT with 21M parameters achieves 84.8% top-1 accuracy on ImageNet-1k, being 4.2 times smaller than the pretrained Swin-B (85.2% accuracy with 88M parameters). With higher resolution, our model can reach 86.5% top-1 accuracy, establishing new state-of-the-art performance on ImageNet-1k under aligned settings. Moreover, TinyViT models demonstrate good transfer capacities on downstream tasks. For instance, TinyViT-21M gets an AP of 50.2 on COCO object detection benchmark, being 2.1 points superior to Swin-T using 28M parameters.

In summary, the main cotributions of this work are twofold.

- We propose a fast pretraining distillation framework to unleash the capacity of small models by fully leveraging the large-scale pretraining data. To our best knowledge, this is the first work exploring small model pretraining.
- We release a new family of tiny vision transformer models, which strike a good trade-off between computation and accuracy. With pretraining distillation, such models demonstrate good transfer ability on downstream tasks.

2 Related Work

In this section, we review the related work on large-scale pretraining, small vision transformers, and knowledge distillation. It is notable that our work is orthogonal to existing literature on model compression techniques such as quantization [43,35,26,43] and pruning [81,69,70,38]. These techniques can be used as a post-processing for our TinyViT to further improve model efficiency.

Large-scale pretraining. Bommasani *et al.* [6] first coined the concept of foundation models that are pretrained from large-scale data and have outstanding performance in various downstream tasks. For example, BERT [19] and GPT-3 [51] have been demonstrated to be effective foundation models in

natural language processing. Recently, there are some research efforts in developing foundation models in computer vision, including CLIP [50], Align [36] and Florence [72]. They have shown impressive transfer and zero-shot capabilities. However, these large models are unsuitable for downstream applications with limited computational budgets. By contrast, our work investigates the pretraining method for small models and improves their transferability to various downstream tasks.

Small vision transformers. Lightweight CNNs have powered many mobile vision tasks [33,59]. Recently, there are several attempts developing light vision transformers (ViTs). Mehta *et al.* [44] combined standard convolutions and transformers to develop MobileViT, which outperforms the prevailing MobileNets [33] and ShuffleNet [79]. Gong *et al.* [22] employed NAS and identified a family of efficient ViTs with MACs ranging from 200M to 800M, surpassing the state-of-the-art. Graham *et al.* [24] optimized the inference time of small and medium-sized ViTs and generated a family of throughput-efficient ViTs. Different from these manually designed or automatically searched small models, our work explores model contraction to generate small models by progressively slimming a large seed model, which can be considered as a complementary work to existing literature on scaling-up large vision transformers [12,75,54,41].

Knowledge distillation. Distillation in a teacher-student framework [31] is widely used to leverage knowledge from large teacher models. It has been extensively studied in convolutional networks [23]. Recently, there are several research works in developing distillation techniques for ViTs [70,80]. For example, Touvron et al. [61] introduced a distillation token to allow the transformer to learn from a ConvNet teacher, while Jia et al. [37] proposed to excavate knowledge from the teacher transformer via the connection between images and patches. Distillation for ViTs is still under-explored, especially for pretraining distillation.

In knowledge distillation, the mostly related work to ours is the recent FKD [56]. Both methods share a similar spirit on saving teacher logits to promote training efficiency, but our framework has two advantages. 1) More efficient. Instead of saving the explicit information of each transformation in data augmentation using hundreds of bytes, such as crop coordinates and rotation degree, our framework only needs 4 bytes to store a random seed. The seed will be used as the initial state of the random number generator to reproduce the number sequence that controls the transformations in data augmentation to generate crop coordinates and rotation degree, *etc.* 2) More general. Our framework supports all existing types of data augmentation including the complex Mixup [76] and Cutmix [73], which are not explored in FKD. Moreover, the studied problem in [56] is different to ours. We focus on pretraining-stage distillation for transformers, while FKD explores finetune-stage distillation for CNN models.

3 TinyViT

This section proposes TinyViT, a new family of tiny and efficient models with fast pretraining distillation on large-scale data. We first introduce the fast knowl-

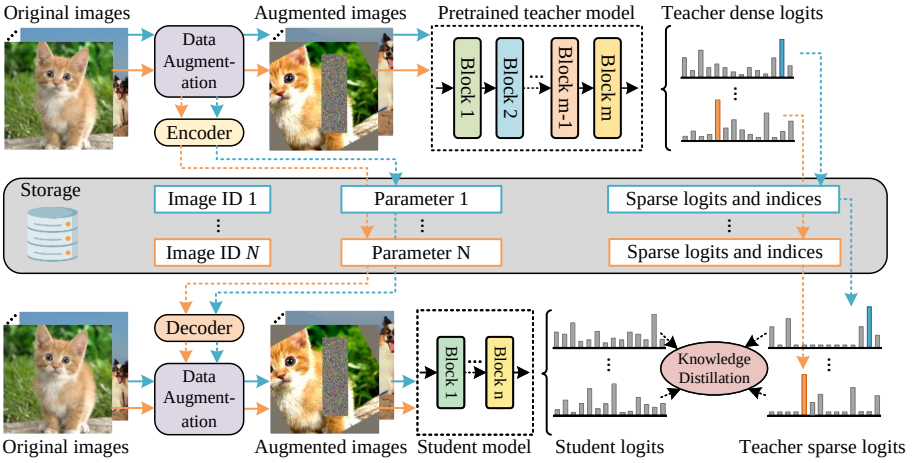


Fig. 2: Our fast pretraining distillation framework. **Top:** the branch for saving teacher logits. Encoded data augmentation and sparsified teacher logits are saved. **Middle:** the disk for storing information. **Bottom:** the branch for training the student. The decoder reconstructs the data augmentation, and distillation is conducted between the teacher logits and student outputs. Note that the two branches are independent and asynchronous.

edge distillation framework for small model pretraining in Sec. 3.1. Then we design a new tiny model family with good computation/accuracy trade-off by progressivley scaling down a large seed model in Sec. 3.2.

3.1 Fast Pretraining Distillation

We observe that direct pretraining of small models on massive data does not bring much gains, especially when transferring them to downstream tasks, as presented in Fig. 1. To address this issue, we resort to knowledge distillation to further unveil the power of pretraining for small models. Different from prior work that pays most attention to finetune-stage distillation [61], we focus on pretraining distillation, which not only allows small models to learn from large-scale model, but also elevates their transfer capacities for downstream tasks.

Pretraining with distillation is inefficient and costly, because a considerable proportion of computing resources is consumed on passing training data through the large teacher model in each iteration, rather than training the target small student. Also, a giant teacher may occupy the most GPU memory, slowing down the training speed of the target students (due to limited batch size). To solve this problem, we propose a fast pretraining distillation framework. As depicted in Fig. 2, we store the information of data augmentation and teacher predictions in advance. During training, we reuse the stored information to precisely replicate the distillation procedure, successfully omitting the forward computation and memory occupation of the large teacher model.

Mathematically, for an input image x with strong data augmentation $\mathcal{A}$, such as RandAugment [17] and CutMix [73], we store both $\mathcal{A}$ and teacher prediction $\hat{\mathbf{y}} = T(\mathcal{A}(x))$, where $T(\cdot)$ and $\mathcal{A}(x)$ are the teacher model and the augmented image. It is notable that passing the same image through the same data augmentation pipeline multiple times will generate different augmented images due to the inherent randomness in data augmentation. Therefore, the pair $(\mathcal{A}, \hat{\mathbf{y}})$ needs to be saved for each image in each iteration, as illustrated in Fig. 2.

In the training process, we only need to recover the pairs $(\mathcal{A}, \hat{\mathbf{y}})$ from stored files, and optimize the following objective function for student model distillation:

$$\mathcal{L} = CE\left(\hat{\mathbf{y}}, S(\mathcal{A}(x))\right), \tag{1}$$

where $S(\cdot)$ and $CE(\cdot)$ are the student model and cross entropy loss, respectively. Note that our framework is label-free, *i.e.*, with no need for ground-truth labels, because we only use the soft labels generated by teacher models for training. Therefore, it can utilize numerous off-the-shelf web data without labels for large-scale pretraining. Such a label-free strategy is workable in practice because the soft labels are accurate enough while carrying a lot of discriminative information for classification such as category relations. We also observe that distillation with ground-truth would cause slight performance drops. The reason may be that not all the labels in ImageNet-21k [18] are mutually exclusive [53], including correlative pairs like "chair" and "furniture", "horse" and "animal". Therefore, the one-hot ground-truth label could not describe an object precisely, and in some cases it suppresses either child classes or parent classes during training. Moreover, our distillation framework is as fast as training models without distillation since the cumbersome teacher $T(\cdot)$ is removed during training in Eq. (1).

Besides, our distillation framework is fast due to two key components: sparse soft labels and data augmentation encoding. They can largely reduce the storage consumption while improving memory efficiency during training.

Sparse soft labels. Let's consider the teacher model outputs C logits for the prediction. It often consumes much storage space to save the whole dense logits of all augmented images if C is large, *e.g.*, $C = 21,841$ for ImageNet-21k. Therefore, we just save the most important part of the logits, *i.e.*, sparse soft labels. Formally, we select the top-K values in $\hat{\mathbf{y}}$, *i.e.*, $\{\hat{y}_{\mathcal{I}(k)}\}_{k=1}^{K} \in \hat{\mathbf{y}}$, and store them along with their indices $\{\mathcal{I}(k)\}_{k=1}^{K}$ into our label files. During training, we only reuse the stored sparse labels for distillation with label smoothing [58,55], which is defined as

$$\hat{y}_c = \begin{cases} \hat{y}_{\mathcal{I}(k)} & \text{if } c = \mathcal{I}(k), \\ \frac{1-\sum_{k=1}^{K} \hat{y}_{\mathcal{I}(k)}}{C-K} & \text{otherwise}, \end{cases} \tag{2}$$

where $\hat{y}_c$ is the recovered teacher logits for student model distillation, *i.e.*, $\hat{\mathbf{y}} = [\hat{y}_1, \ldots, \hat{y}_c, \ldots, \hat{y}_C]$. When the sparsity factor K is small, *i.e.* $K \ll C$, it can reduce logits' storage by orders of magnitude. Moreover, we empirically show that such sparse labels can achieve comparable performance to the dense labels for knowledge distillation, as presented in Sec. 5.2.

Data augmentation encoding. Data augmentation involves a set of parameters $\mathbf{d}$, such as the rotation degree and crop coordinates, to transform the input image. Since $\mathbf{d}$ is different for each image in each iteration, saving it directly becomes memory-inefficient. To solve this problem, we encode $\mathbf{d}$ by a single parameter $d_0 = \mathcal{E}(\mathbf{d})$, where $\mathcal{E}(\cdot)$ is the encoder in Fig. 2. Then in the training process, we recover $\mathbf{d} = \mathcal{E}^{-1}(d_0)$ after loading d_0 in the storage files, where $\mathcal{E}^{-1}(\cdot)$ is viewed as the decoder. Therefore, the data augmentation can be accurately reconstructed. In practice, a common choice for the decoder is the pseudo-random number generator (i.e. PCG [47]). It takes a single parameter as the input and generates a sequence of parameters. As for the encoder, we simply implement it by a generator for d_0 and reusing the decoder $\mathcal{E}^{-1}(\cdot)$. It outputs $\mathbf{d} = \mathcal{E}^{-1}(d_0)$ for the teacher model. d_0 is saved for the decoder to reproduce $\mathbf{d}$ when training the student. Thus, the implementation becomes more efficient.

3.2 Model Architectures

In this subsection, we present a new family of tiny vision transformers by scaling down a large model seed with a progressive model contraction approach [21]. Specifically, we start with a large model and define a basic set of contraction factors. Then in each step, smaller candidate models are generated around the current model by adjusting the contraction factors. We select models that satisfy both constraints on the number of parameters and throughput. The model with the best validation accuracy will be utilized for further reduction in the next step until the target is achieved. This is a form of *constrained local search* [32] in the model space spanned by the contraction factors.

We adopt a hierarchical vision transformer as the basic architecture, for the convenience of dense prediction downstream tasks like detection that require multi-scale features. More concretely, our base model consists of four stages with a gradual reduction in resolution similar to Swin [42] and LeViT [24]. The patch embedding block consists of two convolutions with kernel size 3, stride 2 and padding 1. We apply lightweight and efficient MBConvs [33] in Stage 1 and down sampling blocks, since convolutions at earlier layers are capable of learning low-level representation efficiently due to their strong inductive biases [24,67]. The last three stages are constructed by transformer blocks, with window attention to reduce computational cost. The attention biases [24] and a 3×3 depthwise convolution between attention and MLP are introduced to capture local information [66,15]. Residual connection [28] is applied on each block in Stage 1, as well as attention blocks and MLP blocks. All activation functions are GELU [30]. The normalization layers of convolution and linear are BatchNorm [34] and LayerNorm [3], respectively.

Contraction factors. We consider the following factors to form a model:

- $\gamma_{D_{1-4}}$: embeded dimension of four stages respectively. Decreasing them results in a thinner network with fewer heads in multi-head self-attention.
- $\gamma_{N_{1-4}}$: the number of blocks in four stages respectively. The depth of the model is decreased by reducing these values.

- $\gamma_{W_{2-4}}$: window size in the last three stages respectively. As these values become smaller, the model has fewer parameters and higher throughput.
- γ_R: channel expansion ratio of the MBConv block. We can obtain a smaller model size by reducing this factor.
- γ_M: expansion ratio of MLP for all transformer blocks. The hidden dimension of MLP will be smaller if scaling down this value.
- γ_E: the dimension of each head in multi-head attention. The number of heads will be increased when scaling it down, bringing lower computation cost.

We scale down the above factors with a progressive model contraction approach [21] and generate a new family of tiny vision transformers: All models share the same factors: $\{\gamma_{N_1}, \gamma_{N_2}, \gamma_{N_3}, \gamma_{N_4}\} = \{2, 2, 6, 2\}$, $\{\gamma_{W_2}, \gamma_{W_3}, \gamma_{W_4}\} = \{7, 14, 7\}$ and $\{\gamma_R, \gamma_M, \gamma_E\} = \{4, 4, 32\}$. For the embeded dimensions $\{\gamma_{D_1}, \gamma_{D_2}, \gamma_{D_3}, \gamma_{D_4}\}$, TinyViT-21M: {96, 192, 384, 576}, TinyViT-11M: {64, 128, 256, 448} and TinyViT-5M: {64, 128, 160, 320}.

4 Analysis and Discussions

In this section, we provide analysis and discussions on two key questions: 1) What are the underlying factors limiting small models to fit large data? 2) Why distillation can unlock the power of large data for small models? To answer the above questions, we conduct experiments on the widely used large-scale benchmark ImageNet-21k [18], which contains 14M images with 21,841 categories

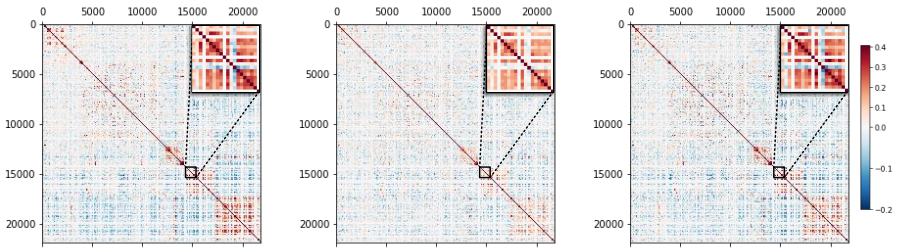
What are the underlying factors limiting small models to fit large data? We observe that there are many hard samples existing in IN-21k, *e.g.*, images with wrong labels and similar images with different labels due to the existence of multiple equally prominent objects in the images. This is also recognized by existing literature [74,5,53] and approximately 10% images in ImageNet are considered as hard samples. Small models struggle to fit these hard samples, leading to low training accuracy compared to large models (TinyViT-21M: 53.2% *vs.* Swin-L-197M [42]: 57.1%) and limited transferability on ImageNet-1k (TinyViT-21M w/ pretraining: 83.8% *vs.* w/o pretraining: 83.1%).

To verify the impact of hard samples, we resort to two techniques. 1) Inspired by [5], we exploit the powerful pretrained model Florence [72] finetuned on ImageNet-21k to identify the images whose labels lie outside the top-5 predictions of Florence. Through this procedure, we remove 2M images from ImageNet-21k, approximately 14%, and then pretrain TinyViT-21M and Swin-T on the cleaned dataset. 2) We perform distillation to pretrain TinyViT-21M/Swin-T using Florence as the teacher model, which generates soft labels to replace the polluted groundtruth labels in ImageNet-21k. The results of the pretrained models with finetuning on ImageNet-1k are reported in Tab. 1.

We obtain several insights from the results. 1) Pretraining small models on the original ImageNet-21k dataset brings limited performance gains on ImageNet-1k (0.7% for both Swin-T and TinyViT-21M). 2) After removing parts of the hard samples in ImageNet-21k, both models can better leverage the large data and achieve higher performance gains (1.0%/1.1% for Swin-T/TinyViT-21M).

Table 1: Impact of hard samples. Models are pretrained on IN-21k and then finetuned on IN-1k.

#	Model	Pretraining Dataset	IN-1k Top-1(%)	IN-Real [5] Top-1(%)	IN-V2 [52] Top-1(%)
0	Swin-T [42]	Train from scratch on IN-1k	81.2	86.7	69.7
1		Original IN-21k	81.9(+0.7)	87.0(+0.3)	70.6(+0.9)
2		Cleaned IN-21k	82.2(+1.0)	87.3(+0.6)	71.1(+1.4)
3		Original IN-21k w/ distillation	83.4(+2.2)	88.0(+1.3)	72.6(+2.9)
4	TinyViT-21M (**ours**)	Train from scratch on IN-1k	83.1	88.1	73.1
5		Original IN-21k	83.8(+0.7)	88.4(+0.3)	73.8(+0.7)
6		Cleaned IN-21k	84.2(+1.1)	88.5(+0.4)	73.8(+0.7)
7		Original IN-21k w/ distillation	84.8(+1.7)	88.9(+0.8)	75.1(+2.0)



(a) Teacher (b) TinyViT w/o distill. (c) TinyViT w/ distill.

Fig. 3: Pearson correlations of output predictions on ImageNet-21k.

3) Distillation is able to avoid the defects of hard samples, because it does not use the groundtruth labels that are the main cause of hard samples. Thus, it gets higher improvements (2.2%/1.7% for Swin-T and TinyViT-21M).

Why can distillation improve the performance of small models on large datasets? The answer is that the student models can directly learn domain knowledge from teachers. Namely, the teacher injects class relationship prior when training the student, while filtering noisy labels (hard samples) for small student models.

To analyze the class relationships of teacher predictions, we select 8 images per class from IN-21k with totally 21,841 classes. These images are then fed into Florence [72] to extract prediction logits. Following [60], we can generate the heatmap of Pearson correlation coefficients between classes on the prediction logits. In Fig. 3(a), simialr or related classes clearly have a high correlations with each other (red), illustrated by the block diagonal structure. In addition, the teacher model can also capture uncorrelated classes (shown in blue). This observation verifies that teacher predictions indeed reveal class relationships.

We compare the Pearson correlations on the predictions of TinyViT-21M w/o and w/ distillation, as shown in Fig. 3(b) and Fig. 3(c) respectively. The block diagonal structure is less obvious without distillation, indicating that the small model is difficult to capture more class relations. However, distillation can guide the student model to imitate the teacher behaviors, leading to better excavating

Table 2: Ablation study on different pretraining strategies for Swin [42] and DeiT [61]. The performance on IN-1k is reported.

Model	#Params (M)	Train on IN-1k	Pretrain on IN-21k	
			w/o distill.	w/ distill.
DeiT-Ti [61]	5	72.2	73.0(+0.8)	74.4(+2.2)
DeiT-S [61]	22	79.9	80.5(+0.6)	82.0(+2.1)
Swin-T [42]	28	81.2	81.9(+0.7)	83.4(+2.2)

knowledge from large datasets. As shown in Fig. 3(c), the Pearson correlations of TinyViT with distillation are closer to the teacher.

5 Experiments

In this section, we first provide ablation studies on our proposed fast pretraining distillation framework. Next, we compare our TinyViT with other state-of-the-art models. At last, we demonstrate the transferability on downstream tasks.

5.1 Implementation Details

Pretraining on ImageNet. For the pretraining on IN-21k [18], we pretrain TinyViT for 90 epochs, then finetune the pretrained models for 30 epochs on IN-1k. For the training from scratch on IN-1k, we train our models for 300 epochs. More details are shown in *Supplementary Materials* (Sec. B).

Knowledge distillation. We pre-store the top-100 logits of teacher models for IN-21k, including Swin-L [42], BEiT-L [4], CLIP-ViT-L/14 [50,20] and Florence [72] for all 90 epochs. Note that CLIP-ViT-L/14 and Florence are finetuned on IN-21k for 30 epochs to serve as teachers. Then, we distill the student models using the stored teacher logits with the same hyper-parameters as the distillation involving the teacher model. The distillation temperature is set to 1.0. We disable Mixup [76] and Cutmix [73] for pretraining distillation on TinyViT. All models are implemented using PyTorch [49] with timm library [65].

5.2 Ablation Study

Impact of pretraining distillation on existing small ViTs. We study the effectiveness of our proposed fast pretraining distillation framework on two popular vision transformers: DeiT [61] and Swin [42]. As shown in Tab. 2, comparing to training from scratch on IN-1k, pretraining without distillation on IN-21k can only bring limited gains, *i.e.* 0.8%/0.6%/0.7% for DeiT-Ti/DeiT-S/Swin-T, respectively. However, our proposed fast pretraining distillation framework increases the accuracy by 2.2%/2.1%/2.2% respectively. It indicates that pretraining distillation allows small models to benefit more from large-scale datasets.

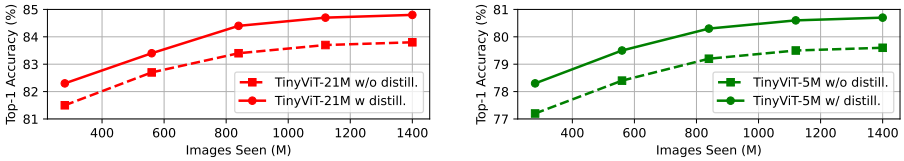


Fig. 4: Comparison on pretrained TinyViT-21M/5M over training data size.

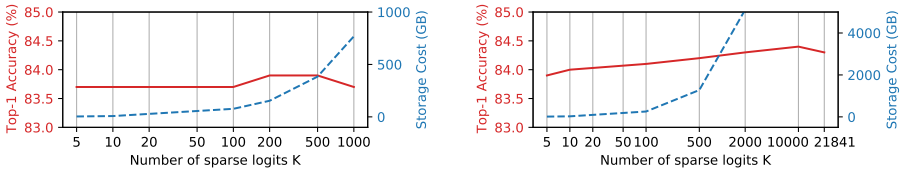


Fig. 5: The accuracy on IN-1k and storage cost of TinyViT-21M along different saved logits K. **Left:** distill TinyViT-21M on IN-1k **Right:** distill TinyViT-21M on IN-21k then finetune it on IN-1k.

Impact of pretraining data scale. We investigate the representation quality of TinyViT-5M/21M with respect to the total number of images "seen" (batch size times number of steps) during pretraining on IN-21k, following the strategies in [75]. We use CLIP-ViT-L/14 [50,20] as the teacher. The results on IN-1k after finetuning are shown in Fig. 4. We have the following observations. 1) For both models, pretraining distillation can consistently brings performance gains over different data size. 2) All models tends to saturate as the number of epochs increase, which may be bottlenecked by the model capacities.

Impact of the number of saved logits. The effects of sparse logits on distilling TinyViT-21M by using Swin-L [42] as the teacher model are shown in Fig. 5. On both IN-1k and IN-21k, we observe that the accuracy increases as the number of sparse logits K grows until saturation, meanwhile the storage cost grows linearly.

This observation is aligned with existing work on knowledge distillation [60,56], where teacher logtis capture class relationships but also contain noise. This makes it possible to sparsify teacher logits such that the class relationships are reserved while reducing noise. Moreover, memory consumption also impose constraints on the choice of K. To obtain comparable accuracy under limited storage space, we select the slightly larger K, where K=10 (1.0% logits) on IN-1k for 300 epochs and K=100 (0.46% logits) on IN-21k for 90 epochs using 16 GB/481 GB storage cost, respectively.

Impact of teacher models. We evaluate the impact of teacher models for pretraining distillation. As shown in Tab. 3, a better teacher can yield better student models (#1 *vs.* #2 *vs.* #3 and #4). TinyViT-21M distilled by Florence on IN-21k is 1.0%/0.6%/1.0% higher in top-1 accuracy on three benchmark datasets than trained from scratch on IN-21k (#0 *vs.* #4). However, better teacher models are often large in model size, resulting in high GPU memory consumption and long training time, e.g., Florence (#4) with 682M parameters occupies 11GB GPU memory and leads to 2.4 times longer training time.

Table 3: Ablation study on different teacher models for pretraining distillation. Teacher performance are listed in the brackets: (the number of parameters, linear probe performance on IN-1k). We report the training time cost and memory consumption of teacher models on NVIDIA V100 GPUs without using our proposed fast pretraining distillation.

#	IN-21k Pretrained Teacher	IN-1k Top-1(%)	IN-Real Top-1(%)	IN-V2 Top-1(%)	Training Time (GPU Hours)	Memory (GB)
0	w/o distill.	83.8	88.4	73.8	3,360	0
1	BEiT-L (326M, 84.1) [4]	84.1	88.4	73.8	6,415 (1.9×)	3.9
2	Swin-L (229M, 84.4) [42]	84.2	88.6	73.9	5,804 (1.7×)	6.8
3	CLIP-ViT-L/14 (321M, 85.2) [50]	84.8	88.9	75.1	7,087 (2.1×)	2.7
4	Florence (682M, 86.2) [72]	84.8	89.0	74.8	7,942 (2.4×)	10.7

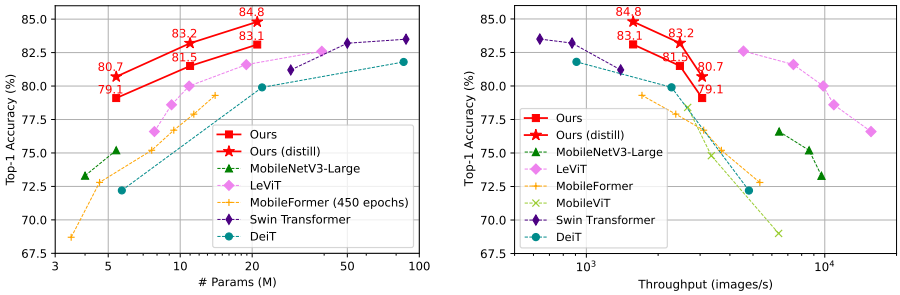


Fig. 6: Comparison with state-of-the-art methods on IN-1k.

Note that our fast pretraining distillation framework simply loads the teacher logits from a hard disk during training. Therefore, it does not require additional GPU memory and has the same training time as #0. Moreover, the framework is compatible with all types of teacher models. Therefore, the performance of TinyViT can be further improved by introducing more powerful teachers.

5.3 Results on ImageNet

In this section, we compare our scaled TinyViT models with state-of-the-art methods on IN-1k [18]. The performance is reported in Fig. 6. The models with ⚗ indicates pretraining on IN-21k with the proposed fast distillation framework using CLIP-ViT-L/14 [50,20] as the teacher. It shows that, without distillation, our TinyViT models achieve comparable performance to current prevailing methods, such as Swin transformer [42] and LeViT [24], with similar parameters. This indicates the effectiveness of the proposed new architectures and the model scaling techniques. Moreover, with the fast pretraining distillation, the performance of TinyViT can be largely improved, outperforming the state-of-the-art CNN, transformer and hybrid models. In particular, using only 21M parameters, TinyViT trained from scratch on IN-1k gets 1.9%/3.2% higher top-1 accuracy than Swin-T [42] and DeiT-S [61] respectively, while after pretraining with distillation on IN-21k, the improvements arise to 3.6% and 4.9%. With higher resolution, TinyViT-21M reaches a top-1 accuracy of 86.5%, establishing

Table 4: TinyViT performance on IN-1k [18] with comparisons to state-of-the-art models. MACs (multiply–accumulate operations) and Throughput are measured using the GitHub repository of [1,24] and a V100 GPU. ⚗: pretrain on IN-21k with the proposed fast distillation; ↑: finetune with higher resolution.

	Model	Top-1 (%)	Top-5 (%)	#Params (M)	MACs (G)	Throughput (images/s)	Input	Arch.
5-10M #Params	MoblieViT-S [44]	78.4	-	6	1.8	2,661	256	Hybrid
	ViTAS-DeiT-A [57]	75.5	92.4	6	1.3	3,504	224	Trans
	GLiT-Tiny [9]	76.3	-	7	1.5	3,262	224	Trans
	Mobile-Former-214M [14]	76.7	-	9	0.2	3,105	224	Hybrid
	CrossViT-9 [10]	77.1	-	9	2.0	2,659	224	Trans
	TinyViT-5M (ours)	**79.1**	94.8	5.4	1.3	3,060	224	Hybrid
	TinyViT-5M⚗ (ours)	**80.7**	95.6	5.4	1.3	3,060	224	Hybrid
11-20M	ResNet-18 [28]	70.3	86.7	12	1.8	8,714	224	CNN
	PVT-Tiny [63]	75.1	-	13	1.9	2,791	224	Trans
	ResT-Small [78]	79.6	94.9	14	2.1	2,037	224	Trans
	LeViT-256 [24]	81.6	-	19	1.1	7,386	224	Hybrid
	CoaT-Lite Small [68]	81.9	95.6	20	4.0	1,138	224	Trans
	TinyViT-11M (ours)	**81.5**	95.8	11	2.0	2,468	224	Hybrid
	TinyViT-11M⚗ (ours)	**83.2**	96.5	11	2.0	2,468	224	Hybrid
>20M	DeiT-S [61]	79.9	95.0	22	4.6	2,276	224	Trans
	T2T-ViT-14 [71]	81.5	95.7	21	4.8	1,557	224	Trans
	AutoFormer-S [11]	81.7	95.7	23	5.1	1,341	224	Trans
	Swin-T [42]	81.2	95.5	28	4.5	1,393	224	Trans
	CrossViT-15 [10]	82.3	-	28	6.1	1,306	224	Trans
	EffNet-B5 [59]	83.6	96.7	30	9.9	330	456	CNN
	TinyViT-21M (ours)	**83.1**	96.5	21	4.3	1,571	224	Hybrid
	TinyViT-21M⚗ (ours)	**84.8**	97.3	21	4.3	1,571	224	Hybrid
	TinyViT-21M⚗ ↑384 (ours)	**86.2**	97.8	21	13.8	394	384	Hybrid
	TinyViT-21M⚗ ↑512 (ours)	**86.5**	97.9	21	27.0	167	512	Hybrid

new state-of-the-art performance on IN-1k for small models. Besides, TinyViT surpasses automatically searched models, such as AutoFormer [11] and GLiT [9].

5.4 Transfer Learning Results

Linear Probe. For linear probe, we follow the same setting as in MOCO v3 [13], *i.e.*, replacing the head of TinyViT models with a linear layer, while only finetuning the linear layer on downstream datasets and frozing other weights. We consider five classification benchmarks: CIFAR-10 [39], CIFAR-100 [39], Flowers [46], Cars [2] and Pets [48]. The results are reported in Tab. 5.

We compare the performance of TinyViT-21M with 4 different training settings. It is clear that distillation can improve the linear probe performance of TinyViT (#0 *vs.* #1, #2 *vs.* #3). Besides, when trained on larger datasets (*i.e.*, IN-21k), TinyViT gets more than 10% gains over CIFAR-100, Flowers and Cars (#0,#1 *vs.* #2, #3), indicating better representability. Thus, pretraining with distillation on large-scale datasets achieves the best representability (#3).

Few-shot Learning. We also evaluate the transferability of TinyViT with different training settings on few-shot learning benchmark [25]. The benchmark datasets include: CropDisease [45], EuroSAT [29], ISIC 2018 [16] and

Table 5: Performance of TinyViT-21M w/ and w/o pretraining for linear probe and few-shot image classification.

		Linear probe					5-shot				20-shot				50-shot			
#	Training dataset	CIFAR-10	CIFAR-100	Flowers	Cars	Pets	ISIC	EuroSAT	CropD	ChestX	ISIC	EuroSAT	CropD	ChestX	ISIC	EuroSAT	CropD	ChestX
0	IN-1k	91.7	75.2	80.9	56.3	86.5	42.9	82.4	92.2	24.8	56.7	91.0	97.4	29.0	63.7	94.2	98.6	31.8
1	IN-1k⚗	91.7	74.5	82.4	61.7	85.5	43.0	83.0	94.2	24.4	58.5	91.8	97.9	28.6	66.2	94.3	98.9	31.8
2	IN-21k	96.3	84.7	99.7	67.7	92.6	52.5	87.4	97.4	24.6	66.5	93.7	99.1	29.4	73.4	95.5	99.5	33.4
3	IN-21k⚗	96.9	86.6	99.7	75.1	93.8	53.5	88.1	98.0	24.7	67.3	93.9	99.3	29.5	74.2	96.0	99.5	33.2

Table 6: Comparison on COCO [40] object detection using Cascade Mask R-CNN [7,27] for 12 epochs. We report the number of parameters of the backbone.

#	Backbone	#Params	IN-1k	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
0	Swin-T [42]	28M	81.2	48.1	67.1	52.1	31.1	51.2	63.5
1	TinyViT-21M	21M	83.1	49.6 (+1.5)	68.5	54.2	32.3	53.2	64.8
2	TinyViT-21M⚗	21M	84.8	50.2 (+2.1)	69.4	54.4	32.9	53.9	65.2

ChestX [64]. The average accuracy over 600 episodes is reported in Tab. 5. We obtain same observations as the linear probe results, except of ChestX, where gray-scale medical images are the least similar to natural images, as well as few in the training dataset for the teacher models and the student models. In combination of these results, we can conclude that pretraining distillation is significant in improving the representability of small models, and thus our proposed fast pretraining distillation framework is effective.

Object Detection. We also investigate the transfer ability of our TinyViT on object detection task [40]. We use Cascade R-CNN [7] with Swin-T [42] as our baseline. We follow the same training settings used in Swin transformer [42]. The results on COCO 2017 validation set are reported in Tab. 6. Under the same training recipe, our TinyViT architecture achieves better performance than Swin-T, getting 1.5% AP improvements. Furthermore, after applying pretraining distillation, TinyViT gets another 0.6% AP improvements, being 2.1% higher than Swin-T. This clearly demonstrates our fast pretraining distillation framework is effective and capable of improving the transfer ability of small models.

6 Conclusions

We have proposed a new family of tiny and efficient vision transformers pretrained on large-scale datasets with our proposed fast distillation framework, named TinyViT. Extensive experiments demonstrate the efficacy of TinyViT on ImageNet-1k, and its superior transferability on various downstream benchmarks. In future work, we will consider using more data to further unlock the representability of small models with the assistance of more powerful teacher models. Designing a more effective scaling down method to generate small models with better computation/accuracy is another interesting research direction.

References

1. fvcore library. https://github.com/facebookresearch/fvcore/
2. 3d object representations for fine-grained categorization. In: 3dRR (2013)
3. Ba, J.L., Kiros, J.R., Hinton, G.E.: Layer normalization. arXiv (2016)
4. Bao, H., Dong, L., Wei, F.: Beit: Bert pre-training of image transformers. In: ICLR (2022)
5. Beyer, L., Hénaff, O.J., Kolesnikov, A., Zhai, X., Oord, A.v.d.: Are we done with imagenet? arXiv (2020)
6. Bommasani, R.e.a.: On the opportunities and risks of foundation models (2021)
7. Cai, Z., Vasconcelos, N.: Cascade r-cnn: Delving into high quality object detection. In: CVPR (2018)
8. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: ECCV (2020)
9. Chen, B., Li, P., Li, C., Li, B., Bai, L., Lin, C., Sun, M., Yan, J., Ouyang, W.: Glit: Neural architecture search for global and local image transformer. In: ICCV (2021)
10. Chen, C.F., Fan, Q., Panda, R.: Crossvit: Cross-attention multi-scale vision transformer for image classification. ICCV (2021)
11. Chen, M., Peng, H., Fu, J., Ling, H.: Autoformer: Searching transformers for visual recognition. In: ICCV (2021)
12. Chen, W., Huang, W., Du, X., Song, X., Wang, Z., Zhou, D.: Auto-scaling vision transformers without training. In: ICLR (2021)
13. Chen, X., Xie, S., He, K.: An empirical study of training self-supervised vision transformers. In: ICCV (2021)
14. Chen, Y., Dai, X., Chen, D., Liu, M., Dong, X., Yuan, L., Liu, Z.: Mobile-former: Bridging mobilenet and transformer. In: CVPR (2022)
15. Chu, X., Tian, Z., Zhang, B., Wang, X., Wei, X., Xia, H., Shen, C.: Conditional positional encodings for vision transformers. arXiv (2021)
16. Codella, N., Rotemberg, V., Tschandl, P., Celebi, M.E., Dusza, S., Gutman, D., Helba, B., Kalloo, A., Liopyris, K., Marchetti, M., et al.: Skin lesion analysis toward melanoma detection 2018: A challenge hosted by the international skin imaging collaboration (isic). arXiv (2019)
17. Cubuk, E.D., Zoph, B., Shlens, J., Le, Q.V.: Randaugment: Practical automated data augmentation with a reduced search space. In: CVPR (2020)
18. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: CVPR (2009)
19. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. In: NAACL-HLT (1) (2019)
20. 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. ICLR (2021)
21. Feichtenhofer, C.: X3d: Expanding architectures for efficient video recognition. CVPR (2020)
22. Gong, C., Wang, D., Li, M., Chen, X., Yan, Z., Tian, Y., liu, q., Chandra, V.: NASVit: Neural architecture search for efficient vision transformers with gradient conflict aware supernet training. In: ICLR (2022)
23. Gou, J., Yu, B., Maybank, S.J., Tao, D.: Knowledge distillation: A survey. IJCV (2021)

24. Graham, B., El-Nouby, A., Touvron, H., Stock, P., Joulin, A., Jégou, H., Douze, M.: Levit: a vision transformer in convnet's clothing for faster inference. In: ICCV (2021)
25. Guo, Y., Codella, N.C., Karlinsky, L., Codella, J.V., Smith, J.R., Saenko, K., Rosing, T., Feris, R.: A broader study of cross-domain few-shot learning. In: ECCV (2020)
26. Han, S., Mao, H., Dally, W.J.: Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. arXiv (2015)
27. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn. In: ICCV (2017)
28. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2016)
29. Helber, P., Bischke, B., Dengel, A., Borth, D.: Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing (2019)
30. Hendrycks, D., Gimpel, K.: Gaussian error linear units (gelus). arXiv (2016)
31. Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. arXiv (2015)
32. Hoos, H.H., Stützle, T.: Stochastic local search: Foundations and applications. Elsevier (2004)
33. Howard, A., Sandler, M., Chu, G., Chen, L.C., Chen, B., Tan, M., Wang, W., Zhu, Y., Pang, R., Vasudevan, V., et al.: Searching for mobilenetv3. In: ICCV (2019)
34. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: ICML (2015)
35. Jacob, B., Kligys, S., Chen, B., Zhu, M., Tang, M., Howard, A., Adam, H., Kalenichenko, D.: Quantization and training of neural networks for efficient integer-arithmetic-only inference. In: CVPR (2018)
36. Jia, C., Yang, Y., Xia, Y., Chen, Y.T., Parekh, Z., Pham, H., Le, Q., Sung, Y.H., Li, Z., Duerig, T.: Scaling up visual and vision-language representation learning with noisy text supervision. In: ICML (2021)
37. Jia, D., Han, K., Wang, Y., Tang, Y., Guo, J., Zhang, C., Tao, D.: Efficient vision transformers via fine-grained manifold distillation. arXiv (2021)
38. Kong, Z., Dong, P., Ma, X., Meng, X., Niu, W., Sun, M., Ren, B., Qin, M., Tang, H., Wang, Y.: Spvit: Enabling faster vision transformers via soft token pruning. arXiv (2021)
39. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images (2009)
40. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: ECCV (2014)
41. Liu, Z., Hu, H., Lin, Y., Yao, Z., Xie, Z., Wei, Y., Ning, J., Cao, Y., Zhang, Z., Dong, L., et al.: Swin transformer v2: Scaling up capacity and resolution. In: CVPR (2022)
42. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. In: ICCV (2021)
43. Liu, Z., Wang, Y., Han, K., Zhang, W., Ma, S., Gao, W.: Post-training quantization for vision transformer. NeurIPS (2021)
44. Mehta, S., Rastegari, M.: Mobilevit: Light-weight, general-purpose, and mobile-friendly vision transformer. In: ICLR (2021)
45. Mohanty, S.P., Hughes, D.P., Salathé, M.: Using deep learning for image-based plant disease detection. Frontiers in plant science (2016)
46. Nilsback, M.E., Zisserman, A.: A visual vocabulary for flower classification. In: CVPR (2006)

47. O'Neill, M.E.: Pcg: A family of simple fast space-efficient statistically good algorithms for random number generation. TOMS (2014)
48. Parkhi, O.M., Vedaldi, A., Zisserman, A., Jawahar, C.: Cats and dogs. In: CVPR (2012)
49. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al.: Pytorch: An imperative style, high-performance deep learning library. NeurIPS (2019)
50. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: ICML (2021)
51. Radford, A., Narasimhan, K., Salimans, T., Sutskever, I., et al.: Improving language understanding by generative pre-training (2018)
52. Recht, B., Roelofs, R., Schmidt, L., Shankar, V.: Do imagenet classifiers generalize to imagenet? In: ICML (2019)
53. Ridnik, T., Ben-Baruch, E., Noy, A., Zelnik-Manor, L.: Imagenet-21k pretraining for the masses. In: NeurIPS (2021)
54. Riquelme, C., Puigcerver, J., Mustafa, B., Neumann, M., Jenatton, R., Pinto, A.S., Keysers, D., Houlsby, N.: Scaling vision with sparse mixture of experts. In: NeurIPS (2021)
55. Shen, Z., Liu, Z., Xu, D., Chen, Z., Cheng, K.T., Savvides, M.: Is label smoothing truly incompatible with knowledge distillation: An empirical study. In: ICLR (2020)
56. Shen, Z., Xing, E.: A fast knowledge distillation framework for visual recognition. arXiv (2021)
57. Su, X., You, S., Xie, J., Zheng, M., Wang, F., Qian, C., Zhang, C., Wang, X., Xu, C.: Vitas: Vision transformer architecture search. arXiv (2021)
58. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: CVPR (2016)
59. Tan, M., Le, Q.: Efficientnet: Rethinking model scaling for convolutional neural networks. In: ICML (2019)
60. Tang, J., Shivanna, R., Zhao, Z., Lin, D., Singh, A., Chi, E.H., Jain, S.: Understanding and improving knowledge distillation (2020)
61. Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., Jégou, H.: Training data-efficient image transformers & distillation through attention. In: ICML. PMLR (2021)
62. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: NeurIPS (2017)
63. Wang, W., Xie, E., Li, X., Fan, D.P., Song, K., Liang, D., Lu, T., Luo, P., Shao, L.: Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In: ICCV (2021)
64. Wang, X., Peng, Y., Lu, L., Lu, Z., Bagheri, M., Summers, R.M.: Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. In: CVPR (2017)
65. Wightman, R.: Pytorch image models (2019)
66. Wu, K., Peng, H., Chen, M., Fu, J., Chao, H.: Rethinking and improving relative position encoding for vision transformer. In: ICCV (2021)
67. Xiao, T., Dollar, P., Singh, M., Mintun, E., Darrell, T., Girshick, R.: Early convolutions help transformers see better. NeurIPS (2021)
68. Xu, W., Xu, Y., Chang, T., Tu, Z.: Co-scale conv-attentional image transformers. In: ICCV (2021)

69. Yang, H., Yin, H., Molchanov, P., Li, H., Kautz, J.: Nvit: Vision transformer compression and parameter redistribution. In: arXiv (2021)
70. Yu, S., Chen, T., Shen, J., Yuan, H., Tan, J., Yang, S., Liu, J., Wang, Z.: Unified visual transformer compression. In: ICLR (2022)
71. Yuan, L., Chen, Y., Wang, T., Yu, W., Shi, Y., Tay, F.E., Feng, J., Yan, S.: Tokens-to-token vit: Training vision transformers from scratch on imagenet. ICCV (2021)
72. Yuan, L., Chen, D., Chen, Y.L., Codella, N., Dai, X., Gao, J., Hu, H., Huang, X., Li, B., Li, C., Liu, C., Liu, M., Liu, Z., Lu, Y., Shi, Y., Wang, L., Wang, J., Xiao, B., Xiao, Z., Yang, J., Zeng, M., Zhou, L., Zhang, P.: Florence: A new foundation model for computer vision. In: ArXiv (2021)
73. Yun, S., Han, D., Oh, S.J., Chun, S., Choe, J., Yoo, Y.: Cutmix: Regularization strategy to train strong classifiers with localizable features. In: ICCV (2019)
74. Yun, S., Oh, S.J., Heo, B., Han, D., Choe, J., Chun, S.: Re-labeling imagenet: From single to multi-labels, from global to localized labels. In: CVPR (2021)
75. Zhai, X., Kolesnikov, A., Houlsby, N., Beyer, L.: Scaling vision transformers. In: CVPR (2022)
76. Zhang, H., Cisse, M., Dauphin, Y.N., Lopez-Paz, D.: mixup: Beyond empirical risk minimization. In: ICLR (2018)
77. Zhang, J., Peng, H., Wu, K., Liu, M., Xiao, B., Fu, J., Yuan, L.: Minivit: Compressing vision transformers with weight multiplexing. In: CVPR (2022)
78. Zhang, Q., bin Yang, Y.: Rest: An efficient transformer for visual recognition. In: NeurIPS (2021)
79. Zhang, X., Zhou, X., Lin, M., Sun, J.: Shufflenet: An extremely efficient convolutional neural network for mobile devices. In: CVPR (2018)
80. Zhou, W., Xu, C., McAuley, J.: Meta learning for knowledge distillation (2022)
81. Zhu, M., Tang, Y., Han, K.: Vision transformer pruning. In: KDD Workshop on Model Mining (2021)