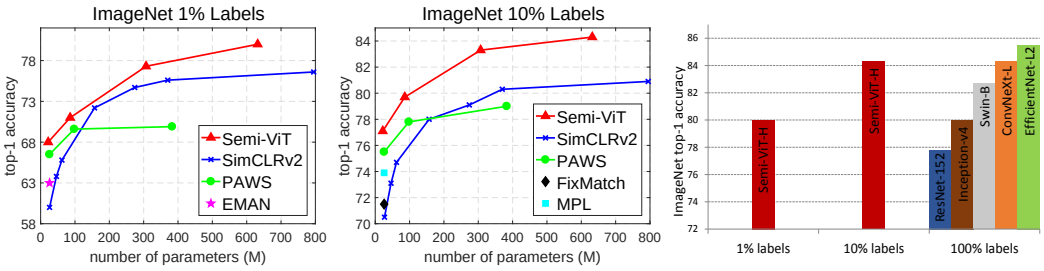
Semi-supervised Vision Transformers at Scale

Zhaowei Cai, Avinash Ravichandran, Paolo Favaro, Manchen Wang,
Davide Modolo, Rahul Bhotika, Zhuowen Tu, Stefano Soatto
AWS AI Labs
{zhaoweic,ravinash,pffavaro,manchenw,dmodolo,ztu,soattos}@amazon.com

Abstract

We study semi-supervised learning (SSL) for vision transformers (ViT), an under-explored topic despite the wide adoption of the ViT architectures to different tasks. To tackle this problem, we use a SSL pipeline, consisting of first *un/self-supervised pre-training*, followed by *supervised fine-tuning*, and finally *semi-supervised fine-tuning*. At the semi-supervised fine-tuning stage, we adopt an exponential moving average (EMA)-Teacher framework instead of the popular FixMatch, since the former is more stable and delivers higher accuracy for semi-supervised vision transformers. In addition, we propose a *probabilistic pseudo mixup* mechanism to interpolate unlabeled samples and their pseudo labels for improved regularization, which is important for training ViTs with weak inductive bias. Our proposed method, dubbed *Semi-ViT*, achieves comparable or better performance than the CNN counterparts in the semi-supervised classification setting. Semi-ViT also enjoys the scalability benefits of ViTs that can be readily scaled up to large-size models with increasing accuracies. For example, Semi-ViT-Huge achieves an impressive 80% top-1 accuracy on ImageNet using only 1% labels, which is comparable with Inception-v4 using 100% ImageNet labels. The code is available at https://github.com/amazon-research/semi-vit.

1 Introduction



(a) (b) (c)

Figure 1: (a) and (b) are the comparisons of our Semi-ViT with the state-of-the-art SSL algorithms at different model scales, and (c) is the comparison with the state-of-the-art supervised models.

In the past few years, Vision Transformers (ViT) [18], which adapt the transformer architectures [64] to the visual domain, have achieved remarkable progresses in supervised learning [63, 44, 73], un/self-supervised learning [16, 12, 25], and many other computer vision tasks [11, 19, 1, 58] (with architecture modifications). However, ViTs have yet to show the same advantage in semi-supervised learning (SSL), where only a small subset of the training data is labeled, a problem in the middle between supervised and un/self-supervised learning. Although several recent methods in SSL have

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

significantly advanced the field [39, 62, 7, 55, 70, 10, 53], the transfer of these methods from Convolutional Neural Networks (CNN) to ViT architectures has yet to show much promise. For example, as discussed in [68], the direct application of FixMatch [55], one of the most popular SSL methods, to ViT leads to an inferior performance (about 10 points worse) than when used with a CNN architecture. The challenge could be potentially caused by the fact that ViTs are known to require more data for training and to have a weaker inductive bias than CNNs [18]. However, in this paper we show that semi-supervised ViTs can outperform the CNN counterparts when trained properly, suggesting promising potential to advance SSL beyond CNN architectures.

To achieve such success, we use the following SSL pipeline: 1) *un/self-supervised pre-training* on all data (both labeled and unlabeled), followed by 2) *supervised fine-tuning* only on labeled data, and finally 3) *semi-supervised fine-tuning* on all data. This pipeline is stable and helps reduce the sensitivity of hyperparameter tuning when training ViTs for SSL in our experiments. At the final stage of *semi-supervised fine-tuning*, we adopt the EMA-Teacher framework [62, 10], an improved version over the popular FixMatch [55]. Unlike FixMatch that often fails to converge when training semi-supervised ViT, EMA-Teacher shows more stable training behaviors and better performance. In addition, we propose *probabilistic pseudo mixup* for the pseudo-labeling based SSL methods, which interpolates the unlabeled samples coupled with pseudo labels for enhanced regularization. In the standard mixup [75] the mixup ratio is randomly sampled from a Beta distribution. In contrast, in the *probabilistic pseudo mixup* the ratio depends on the respective confidences of two mixed-up samples, such that the sample with higher confidence will weigh more in the final interpolated sample. This new data augmentation technique brings non-negligible gains since ViT has weak inductive bias, especially for scenarios where the training is more difficult, *e.g.*, without un/self-supervised pre-training or on data regimes with very few labeled samples (*e.g.*, 1% labels). We call our method *Semi-ViT*. Notice that Semi-ViT is built on exactly the same design of ViTs (*i.e.*, there are neither additional parameters nor architectural changes).

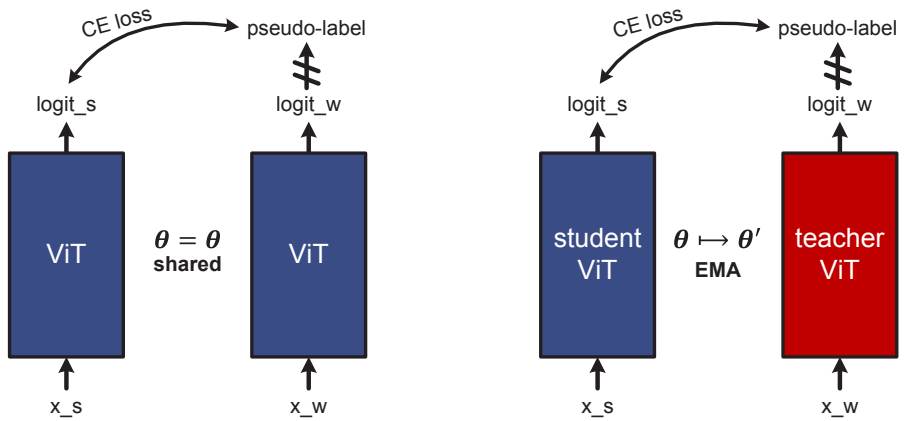
Semi-ViT achieves promising results from different aspects (Figure 1). 1) For the first time, we show that *pure* ViTs can reach comparable or better accuracy than CNNs on SSL[1]. 2) Semi-ViT can be readily scaled up under the SSL setting. This is illustrated in Figure 1 (a) and (b) on ViT architectures at different scales, ranging from ViT-Small to ViT-Huge, and Semi-ViT outperforms the prior art such as SimCLRv2 [15]. 3) Semi-ViT has shown the potential for a substantial reduction of labeling cost. For example, as seen in Figure 1 (c), Semi-ViT-Huge with 1% (10%) ImageNet labels achieves a comparable performance of a fully-supervised Inception-v4 [59] (ConvNeXt-L [45]). This implies a $100\times$ ($10\times$) reduction in human annotation cost. 4) Semi-ViT achieves the state-of-the-art SSL results on ImageNet, *e.g.*, 80.0% (84.3%) top-1 accuracy with only 1% (10%) labels. In addition, the substantial boost in performance by Semi-ViT is not isolated on ImageNet: we find an increase of 13%-21% (7%-10%) top-1 accuracy with 1% (10%) labels over the supervised fine-tuning baselines, for other datasets including Food-101 [9], iNaturalist [30] and GoogleLandmark [52].

2 Semi-supervised Vision Transformers

2.1 Pipeline

Some pipelines for semi-supervised learning exist in the literature. For example: 1) The model is directly trained from scratch using SSL techniques, *e.g.*, FixMatch [55]; 2) The model is un/self-supervised pretrained first and finetuned on the labeled data later [26, 14, 22]; 3) The model is self-supervised pretrained first and then semi-supervised finetuned on both labeled and unlabeled data [10]. In this paper, we instead adopt the following pipeline: at first, optional *self-supervised pre-training* on all the data without using any labels; next, standard *supervised fine-tuning* on the available labeled data; and finally, *semi-supervised fine-tuning* on both labeled and unlabeled data. This procedure is similar to [15], with the difference that they use knowledge distillation [29] in their final stage. We find that this training pipeline is stable to train semi-supervised vision transformers and achieves promising results, with possibly less hyperparameter tuning.

[1] Although [68] was the first to use transformer for SSL, it is a mixture architecture of CNN and ViT and requires to use CNN as the teacher to produce pseudo labels.



(a) FixMatch (b) EMA-Teacher

Figure 2: The framework comparison between FixMatch (a) and EMA-Teacher (b). x_s/x_w is the strongly/weakly augmented view of a sample x, and θ is the model parameters.

2.2 EMA-Teacher Framework

FixMatch [55] emerged as a popular SSL method in the past few years. As discussed in [10], it can be interpreted as a student-teacher framework, where the student and teacher models are identical, as seen in Figure 2 (a). However, FixMatch has unexpected behaviors, especially when the model consists of batch normalization (BN) [35]. Although ViT uses Layer Normalization (LN) [4] instead of BN as normalization, we still found that FixMatch with ViT underperforms the CNN counterparts and often does not converge. This phenomenon was also observed in [68]. A potential reason to this is that the student and the teacher models are identical in FixMatch, which could easily lead to model collapse [26, 22]. This instability of the identical student-teacher framework has also been observed in other areas, e.g. semi-supervised speech recognition [42, 48, 28]. As suggested in [10], the EMA-Teacher (shown in Figure 2 (b)) is an improved version of FixMatch, thus we adopt it for our Semi-ViT. In the EMA-Teacher framework, the teacher parameters θ' are updated by exponential moving average (EMA) from the student parameters θ,

$$\theta' := m\theta' + (1-m)\theta, \tag{1}$$

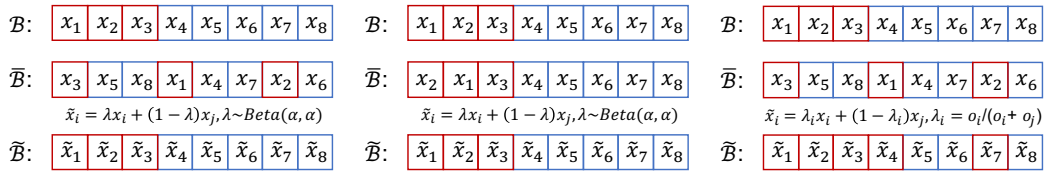
where the momentum decay m is a number close to 1, *e.g.*, 0.9999. The student parameters are updated by standard learning optimization, *e.g.*, SGD or AdamW [46]. The other components are exactly the same as FixMatch, as seen in Figure 2. This temporal weight averaging can stabilize the training trajectories [3, 36] and avoids the model collapse issue [26, 22]. Our experiments also show this EMA-Teacher framework has better results and more stable training behaviors than FixMatch when training Semi-ViT.

2.3 Semi-supervised Learning Formulation

In the EMA-Teacher framework, there are both labeled and unlabeled samples in a minibatch during training. The loss on the labeled samples $\{(x_i^l, y_i^l)\}_{i=1}^{N_l}$ is the standard cross-entropy loss, $\mathcal{L}_l = \frac{1}{N_l}\sum_{i=1}^{N_l} CE(x_i^l, y_i^l)$. For an unlabeled sample $x^u \in \{x_i^u\}_{i=1}^{N_u}$, a weak and a strong augmentation are applied to it, generating $x^{u,w}$ and $x^{u,s}$, respectively. The weak augmented $x^{u,w}$ is forwarded through the teacher network, and outputs the probabilities over classes, $p = f(x^{u,w}; \theta')$. Then the pseudo label is produced by $\hat{y} = \arg\max_c p_c$ with its associated confidence $o = \max p_c$. The pseudo label with confidence higher than a confidence threshold τ is then used to supervise the learning of the student on the strong augmented sample $x^{u,s}$,

$$\mathcal{L}_u = \frac{1}{N_u}\sum_{i=1}^{N_u} [o_i \geq \tau] CE(x_i^{u,s}, \hat{y}_i), \tag{2}$$

where $[\cdot]$ is the indicator function. The overall loss is $\mathcal{L} = \mathcal{L}_l + \mu\mathcal{L}_u$, where μ is the trade-off weight. Note that only the pseudo labels with confidence higher than a threshold contribute to the final loss;



(a) Pseudo Mixup (b) Pseudo Mixup+ (c) Probabilistic Pseudo Mixup

Figure 3: Different variations of mixup on unlabeled data. The red samples are the ones passing the confidence threshold, but not the blue samples.

the others are instead not used. The philosophy behind this filtering is that the pseudo labels with low confidences are noisier and could hijack the SSL training.

3 Probabilistic Pseudo Mixup

3.1 Mixup

Mixup [75] performs convex combinations of pairs of samples and their labels,

$$\begin{aligned} \tilde{x} &= \lambda x_i + (1-\lambda)x_j, \\ \tilde{y} &= \lambda y_i + (1-\lambda)y_j, \end{aligned} \tag{3}$$

where the mixup ratio $\lambda \sim Beta(\alpha, \alpha) \in [0, 1]$, for $\alpha \in (0, \infty)$. The samples are mixed-up usually in a single minibatch during training. Given a minibatch $\mathcal{B}$ and its shuffled version $\bar{\mathcal{B}}$, the mixed-up minibatch is $\tilde{\mathcal{B}} = \lambda\mathcal{B} + (1-\lambda)\bar{\mathcal{B}}$, where λ could be either batch-wise or element-wise. Due to the nature of weak inductive bias, ViT is more data hungry than CNN, thus effective data augmentation, *e.g.*, mixup, is critical for training fully-supervised ViT [18, 63, 44, 73]. This also applies to Semi-ViT since it inherits the nature of weak inductive bias from ViT. Although it is standard to use mixup in supervised learning, how to employ it under pseudo-labeling based SSL framework, *e.g.*, EMA-Teacher, is still unclear yet, and we are going to discuss it next.

3.2 Pseudo Mixup

Under the pseudo-labeling based SSL framework [40, 55, 53, 10], given a unlabeled sample and its pseudo label $(x^u, \hat{y})$, only when its confidence o is not smaller than the confidence threshold τ, it will contribute to loss $\mathcal{L}_u$, as seen in (2). According to their confidence scores, the unlabeled minibatch $\mathcal{B}^u$ can be grouped into a clean subset $\hat{\mathcal{B}}^u = \{(x_i^u, \hat{y}_i) | o_i \geq \tau\}$ and a noisy subset $\dot{\mathcal{B}}^u = \mathcal{B}^u - \hat{\mathcal{B}}^u$. One straightforward solution is to apply mixup on the full unlabeled minibatch $\mathcal{B}^u$, with no differentiation between clean and noisy samples, denoted as *pseudo mixup*, as show in Figure 3 (a). After the pseudo mixup, still only the samples in $\hat{\mathcal{B}}^u$ contribute to the loss, and samples in $\dot{\mathcal{B}}^u$ are abandoned. In this way, the mixup operation is more than just a data augmentation. In fact, a sample in $\dot{\mathcal{B}}^u$ will also contribute to the final loss if it is mixed-up with a sample in $\hat{\mathcal{B}}^u$. As a result, it could involve a substantial number of noisy samples into the loss calculation due to the randomness, which, however, is against the philosophy of pseudo-labeling. Since only the clean subset $\hat{\mathcal{B}}^u$ contributes to the final loss, another choice is to use mixup only on $\hat{\mathcal{B}}^u$, denoted as *pseudo mixup+*, as shown in Figure 3 (b). In this way, no sample in the noisy subset $\dot{\mathcal{B}}^u$ will affect the training.

3.3 Probabilistic Pseudo Mixup

Although the samples in $\dot{\mathcal{B}}^u$ are noisy, they still carry some useful information for the model to learn. The *pseudo mixup* above can somehow leverage those information by blending the noisy and clean pseudo samples together. However, the problem is the mixup ratio is randomly generated from a Beta distribution, which does not depend on the confidence of each sample. This is not ideal. For example, when two samples are mixed-up, the sample with higher confidence should have higher mixup ratio, such that it can weigh more in the final loss. Motivated by this intuition, we propose *probabilistic pseudo mixup* (Figure 3 (c)), where the mixup ratio λ relfects the sample confidences,

$$\lambda_i = o_i / (o_i + o_j). \tag{4}$$

Model	Param	Method	1%	10%	100%
ViT-Base	86M	finetune	57.4	73.7	83.7
		Semi-ViT	71.0	79.7	-
ViT-Large	307M	finetune	67.1	79.2	86.0
		Semi-ViT	77.3	83.3	-
ViT-Huge	632M	finetune	71.5	81.4	86.9
		Semi-ViT	80.0	84.3	-

Table 1: Semi-ViT results comparing with fine-tuning. The models are self-pretrained by MAE[25].

Model	Pretrained	Method	1%	10%
ViT-Small	None	FixMatch	-	✘
		EMA-Teacher	-	65.6
ViT-Base	None	FixMatch	-	✘
		EMA-Teacher	-	68.9
ViT-Base	MAE	FixMatch	✘	74.8
		EMA-Teacher	65.3	78.1

Table 2: The comparison between FixMatch and EMA-Teacher. ✘means the training is failed with accuracy close to 0.

Also, the confidence score of x_i^u is updated after mixup operation as

$$o_i^* = \max(o_i, o_j), \tag{5}$$

because the confidence score should align with the majority of the image content. And the final clean subset $\tilde{\mathcal{B}}^u = \{(\tilde{x}_i^u, \tilde{y}_i^u) | o_i^* \geq \tau\}$ will contribute to the final loss. Using this *probabilistic pseudo mixup* can enhance regularization, leverage information from all samples, even the noisy ones, and not violate the philosophy of pseudo labeling at the same time. It can effectively alleviate the issue of weak inductive bias of Semi-ViT and bring substantial gains, as will be shown in our experiments.

4 Experiments

We evaluate Semi-ViT mainly on ImageNet, which consists of $\sim$1.28M training and 50K validation images. We sample 10%/1% labels from ImageNet training set for semi-supervised evaluation. We study both scenarios: with and without self-supervised pre-training. Without self-pretraining, we only evaluate on 10% labels, since learning from scratch on 1% labels is very difficult. When self-pretrained, MAE [25] is mainly used, and we directly use their pretrained models. All learning is optimized with AdamW [46], using cosine learning rate schedule, with a weight decay of 0.05. The default momentum decay m of (1) is 0.9999. In a minibatch, $N_u = 5N_l$, and the loss trade-off $\mu = 5$. The mixup is the combination of mixup [75] and Cutmix [74] as the implementation of [69]. More details can be found in the appendix.

4.1 Semi-ViT Results

When the model is self-pretrained by MAE [25], we first evaluate the fine-tuning performances of MAE on the labeled data only, as the common practice in self/un-supervised learning literature [26, 14, 22], with results shown Table 1. This already leads to strong semi-supervised baselines, *e.g.*, 81.4 top-1 accuracy for ViT-Huge on 10% labels, indicating MAE is a strong self-supervised learning technique. However, Semi-ViT has additional significant improvements over the strong baselines for all models, *e.g.*, 8.5-13.6 points for 1% labels and 2.9-6.0 points for 10% labels. The fine-tuning results on 100% data are provided as upper-bounds for our Semi-ViT, and their gaps to Semi-ViT are small, *e.g.*, 4.0/2.7/2.6 points for ViT-Base/Large/Huge on 10% labels. An interesting observation is that the larger model is more effective for smaller number of labels, which is consistent with the observations in [15]. For example, the fine-tuning gaps between 1% and 100% labels are 26.3/18.9/15.4 points for ViT-Base/Large/Huge, which are decreasing. The observation on Semi-ViT results is similar, *e.g.*, 12.7/8.7/6.9 points to their upper-bounds on 1% labels. These results have shown that vision transformers can also perform very well in semi-supervised learning, as well as supervised learning and un/self-supervised learning.

Model	Pretrained	Mixup	1%	10%
ViT-Small	None	EMA-Teacher	-	65.6
		Pseudo Mixup	-	68.3
		Pseudo Mixup+	-	68.8
		ProbPseudo Mixup	-	70.9
ViT-Base	None	EMA-Teacher	-	68.9
		Pseudo Mixup	-	71.6
		Pseudo Mixup+	-	72.1
		ProbPseudo Mixup	-	73.5
ViT-Base	MAE	EMA-Teacher	65.3	78.1
		Pseudo Mixup	69.5	78.3
		Pseudo Mixup+	70.1	78.7
		ProbPseudo Mixup	71.0	79.7

Table 3: The comparison among different mixup variations.

Method (ViT-Base)	label	$\tau = 0$	$\tau = 0.3$	$\tau = 0.4$	$\tau = 0.5$	$\tau = 0.6$	$\tau = 0.7$	$\tau = 0.8$	$\tau = 0.9$
EMA-Teacher	1%	63.1	64.4	64.6	65.1	65.3	65.1	64.4	-
EMA-Teacher	10%	75.4	76.7	77.2	77.7	77.9	78.1	78.2	77.9
Semi-ViT	1%	70.8	71.4	71.3	71.3	71.0	70.4	68.6	-
Semi-ViT	10%	79.4	79.5	79.7	79.7	79.6	79.4	-	-

Table 4: The ablation on the confidence threshold.

4.2 Ablation Studies

We ablate different factors of Semi-ViT in this section.

FixMatch v.s. EMA-Teacher is compared in Table 2. These experiments do not use the pseudo mixup techniques of Section 3 yet. When the model is not self-pretrained, the training of FixMatch is unstable and often failed. When the model is self-pretrained, FixMatch training becomes stable, and start to achieve reasonable results, *e.g.*, 74.8 for ViT-Base on 10% labels, which is already better than the prior art on ResNet-50, *e.g.*, 73.9 of MPL [53] and 74.0 of EMAN [10]. But it is only 1.1 points higher than the fine-tuning baseline of Table 1, indicating FixMatch is not an effective SSL framework for ViT. But EMA-Teacher achieves much better results, 3.3 points improvement over FixMatch when self-pretrained. Even without self-pretraining, EMA-Teacher can still achieve decent numbers, but FixMatch is failed.

Probabilistic Pseudo Mixup Different mixup variations on unlabeled data are compared in Table 3. Note that the standard mixup with implementation of [69] is used to the labeled data as usual. The EMA-Teacher does not use any mixup mechanism on the unlabeled data, serving as baselines here. When *pseudo mixup* of Figure 3 (a) is applied on the unlabeled data, the performances usually have some substantial gains over the EMA-Teacher baselines, especially for the scenarios that the training is more difficult, *e.g.*, without self-pretraining or on 1% labels. This shows the importance to use mixup on the unlabeled data for improved regularization. However, as discussed Section 3.2, *pseudo mixup* could involve many noisy samples into training. On the other hand, *pseudo mixup+* of Figure 3 (b) can increase over *pseudo mixup* constantly, by about 0.5 points, showing that removing those noisy samples does help. In addition, *probabilistic pseudo mixup* of Figure 3 (c) can further improve over *pseudo mixup+* by 1-2 points in all cases. These results have implied that those noisy samples do carry some useful information for SSL training, but their weights should be suppressed especially when their confidences are low. This data augmentation technique also effectively alleviate the training difficulty of semi-supervised vision transformers with weak inductive bias.

Effect of Confidence Threshold We ablate the effect of confidence threshold τ of (2) in Table 4. We can find Semi-ViT is quite robust to the low confidence thresholds. One possible reason is that Semi-ViT uses *probabilistic pseudo mixup*. When τ is low, the low-confidence samples will not hijack the training since their contributions depend on their confidence scores. These imply that the hyperparameter of τ can possibly be removed ($\tau = 0$) in Semi-ViT. But EMA-Teacher has a drop of 2-3 points when the confidence threshold is removed. The final choices of τ for different Semi-ViT models are shown in Table 13 in the appendix.

Method (ViT-Base)	label	$m = 0$	$m = 0.9$	$m = 0.99$	$m = 0.999$	$m = 0.9999$	$m = 0.99999$
EMA-Teacher	1%	✘	23.5	49.3	63.1	65.3	59.7
EMA-Teacher	10%	74.8	75.3	76.4	77.2	78.1	77.9
Semi-ViT	1%	69.3	69.7	71.1	71.6	71.0	63.1
Semi-ViT	10%	79.5	79.5	79.6	79.8	79.7	79.0

Table 5: The ablation on the momentum decay of exponential moving average.

Method (ViT-Base)	label	epochs=0	epochs=10	epochs=50	epochs=100	epochs=200
Supervised-ViT	1%	-	24.7	53.6	57.4	56.9
Supervised-ViT	10%	-	66.3	72.9	73.7	73.2
EMA-Teacher	1%	62.7	62.5	60.9	65.3	66.9
EMA-Teacher	10%	76.5	74.2	77.7	78.1	78.2
Semi-ViT	1%	69.7	69.8	70.4	71.0	70.9
Semi-ViT	10%	79.3	79.4	79.6	79.7	79.6

Table 6: The ablation on supervised fine-tuning.

Effect of Momentum Decay Table 5 shows the effect of momentum decay m in the EMA teacher. Note that when $m = 0$, the frameworks is reduced to FixMatch. We can find Semi-ViT is robust to m. For 10% labels, Semi-ViT has very minor changes when m decreases to 0, but EMA-Teacher has a drop of 3.3 points. For 1% labels, the training is more challenging, and the choice of m becomes more important. In this case, Semi-ViT has a drop of 2.3 points from $m = 0.999$ to $m = 0$, but EMA-Teacher could fail when m decreases to 0. The robustness of Semi-ViT to momentum decay m is also attributed to the use of *probabilistic pseudo mixup*.

Effect of Self-pretraining The self-pretraining of MAE [25] has a substantial boost in performances, as seen in Table 3. For ViT-Base, MAE helps to improve by 6.2 and 9.2 points for EMA-Teacher with and without *probabilistic pseudo mixup*, respectively. In addition, it helps to train the models in more challenging scenarios, *e.g.*, 1% labels. Without self-pretraining, the training fails to deliver good results on 1% labels. Notice that, even without pre-training, our Semi-ViT ("ProbPseudo Mixup" in Table 3) also achieves slightly better performance than the CNN counterparts: 70.9 of Semi-ViT-Small v.s. 67.1 of FixMatch-ResNet50 or 69.2 [55] of EMAN-ResNet50 [10] when trained from scratch for 100 epochs.

Effect of Supervised Fine-tuning is ablated in Table 6 by varying the number of *supervised fine-tuning* epochs. The rows of "Supervised-ViT" are the numbers of *supervised fine-tuning*, where the latter *semi-supervised fine-tuning* begins. Semi-ViT is still robust to the length of *supervised fine-tuning*, where the accuracy decrease is 0.4 (1.3) for 10% (1%) labels when *supervised fine-tuning* is removed. However, the performance of EMA-Teacher decreases 1.7 (4.2) points. These show sufficient *supervised fine-tuning* does stabilize the training procedure, especially for less robust framework, *e.g.*, EMA-Teacher. However, notice that *supervised fine-tuning* could sometimes hurt the performance if it is not sufficient (*e.g.*, epochs=10) for EMA-Teacher.

Other Self-pretraining Techniques Beyond MAE, we also experiment on other self-pretraining technique, including MoCo-v3 [16] and DINO [12], in Table 7. By comparing the fine-tuning results, DINO is close to MoCo-v3 for ViT-Base but much better for ViT-Small, and both of them are better than MAE for ViT-Base, suggesting that DINO could be a better self-pretraining technique for smaller scales of ViT models. On top of the strong fine-tuning baselines, the *semi-supervised fine-tuning*, using EMA-Teacher, still has nontrivial improvements for both DINO and MoCo-v3, *e.g.*, 5.8 (2.1) points on 1% (10%) labels for DINO-ViT-Base. In addition, the *probabilistic pseudo mixup* can further improve over the EMA-Teacher, independent of the self-pretraining algorithms. And the final Semi-ViT-Base of DINO is 2.1 (0.5) points better than that of MAE on 1% (10%) labels.

Other Network Architectures Although in this paper we mainly focus on ViT architectures, the proposed *probabilistic pseudo mixup* is not limited to them. We also try it for CNN architectures, *e.g.*, ResNet [27]. However, we find the direct use of the standard mixup does not improve fully-supervised ResNet performances, so will the *probabilistic pseudo mixup* for its SSL setting. Instead, we evaluate it on the recently proposed ConvNeXt [45], which uses mixup for improved results. Since the goal is

Model	Pretrained	Method	1%	10%
ViT-Small	MoCo-v3 [16]	finetune	51.2	69.1
		EMA-Teacher	61.9	72.3
		+ProbPseudo Mixup	64.7	72.9
ViT-Small	DINO [12]	finetune	58.7	73.9
		EMA-Teacher	66.3	76.3
		+ProbPseudo Mixup	68.0	77.1
ViT-Base	MoCo-v3 [16]	finetune	66.3	74.5
		EMA-Teacher	68.9	77.7
		+ProbPseudo Mixup	72.3	79.2
ViT-Base	DINO [12]	finetune	65.0	76.0
		EMA-Teacher	70.8	78.1
		+ProbPseudo Mixup	73.1	80.2

Table 7: Semi-ViT results with other self-pretraining techniques.

Model	Upper-bound	Method	10%
ConvNeXt-T	80.7	supervised	61.2
		EMA-Teacher	70.4
		+ProbPseudo Mixup	74.1
ConvNeXt-S	81.4	supervised	64.1
		EMA-Teacher	71.7
		+ProbPseudo Mixup	75.1

Table 8: The results on ConvNeXt [45].

not to fully reproduce the results of [45], all models are trained only for 100 epochs, including the supervised upper-bounds. The results in Table 8 demonstrate that *probabilistic pseudo mixup* is not limited to ViT, but also to CNN architectures, *e.g.*, with improvements of 3-4 points, suggesting it can be well generalized.

4.3 Comparison with the State-of-the-Art

Semi-ViTs are compared with the state-of-the-art semi-supervised learning algorithms in Table 9. When the model capacity is close, our Semi-ViT has shown much better results than the prior art, *e.g.*, MPL-RN-50 [53] v.s. Semi-ViT-Small, CowMix-RN152 [20] v.s. Semi-ViT-Base, S4L-RN50-4$\times$ [8] v.s. Semi-ViT-Large and SimCLRv2+KD-RN152-3$\times$-SK [15] v.s. Semi-ViT-Huge. The only transformer based SSL method is SemiFormer [68], but it requires to use CNN as the teacher model and blend convolution and transformer modules together for good performances. However, our Semi-ViT is *pure* ViT based, without any additional parameters and architecture changes, and the Semi-ViT-Small model is already better than SemiFormer (77.1 v.s. 75.5). These comparisons support that Semi-ViT does advance the state-of-the-art of semi-supervised learning.

Scalability is an advantage of ViT, and we compare the scalability of Semi-ViT with previous works in Figure 1 (a) and (b). The comparison has shown that Semi-ViT can achieve better trade-off between model capacity and accuracy and can be scaled up more effectively than the prior art, SimCLRv2 [15]. For example, SimCLRv2 and PAWS [2] scale up the model usually in terms of network depth and width, and they seem to saturate when the model is of medium size, *e.g.*, around 300M parameters, but our Semi-ViT continues to improve steadily beyond that point.

Semi-ViT is also compared with the supervised state-of-the-art in Table 10. Our Semi-ViT-Huge is comparable with Inception-v4 [59], but with 100$\times$ annotation cost reduction, and comparable with ConvNeXt-L [45] (better than Swin-B [44]), but with 10$\times$ annotation cost reduction. These comparisons imply that Semi-ViT has great potential for labeling cost reduction.

4.4 Other Datasets

The generalization of Semi-ViT is evaluated on datasets including Food-101 [9], iNaturalist [30] and GoogleLandmark [52]. Since these datasets are beyond ImageNet, we assume that the ImageNet dataset is available and the model is already supervised pretrained on ImageNet, and then the model is finetuned to different target datasets with a few labels. The results are shown in Table 11. On

	Method	Architecture	Param	1%	10%
CNN	UDA [70]	ResNet-50	26M	-	68.8
	FixMatch [55]	ResNet-50	26M	-	71.5
	S4L [8]	ResNet-50 (4×)	375M	-	73.2
	MPL [53]	ResNet-50	26M	-	73.9
	CowMix [20]	ResNet-152	60M	-	73.9
	EMAN [10]	ResNet-50	26M	63.0	74.0
	PAWS [2]	ResNet-50	26M	66.5	75.5
	SimCLRv2+KD [15]	RN152 (3×+SK)	794M	76.6	80.9
Transformer	DINO [12]	ViT-Small	22M	64.5	72.2
	SemiFormer [68]	ViT-S+Conv	42M	-	75.5
	Semi-ViT (ours)	ViT-Small	22M	68.0	77.1
	Semi-ViT (ours)	ViT-Base	86M	71.0	79.7
	Semi-ViT (ours)	ViT-Large	307M	77.3	83.3
	Semi-ViT (ours)	ViT-Huge	632M	80.0	84.3

Table 9: The comparison with the state-of-the-art SSL models.

	Model	Param	Data	top-1	top-5
CNN	ResNet-50 [27]	26M	ImageNet	76.0	93.0
	ResNet-152 [27]	60M	ImageNet	77.8	93.8
	DenseNet-264 [32]	34M	ImageNet	77.9	93.9
	Inception-v3 [60]	24M	ImageNet	78.8	94.4
	Inception-v4 [59]	48M	ImageNet	80.0	95.0
	ResNeXt-101 [72]	84M	ImageNet	80.9	95.6
	SENet-154 [31]	146M	ImageNet	81.3	95.5
	ConvNeXt-L [45]	198M	ImageNet	84.3	-
	EfficientNet-L2 [61]	480M	ImageNet	85.5	97.5
Transformer	ViT-Huge [18]	632M	JFT+ImageNet	88.6	-
	DeiT-B [63]	86M	ImageNet	81.8	-
	Swin-B [44]	88M	ImageNet	83.3	-
	MAE-ViT-Huge [25]	632M	ImageNet	86.9	-
	Semi-ViT-Huge (ours)	632M	1%ImageNet	80.0	93.1
	Semi-ViT-Huge (ours)	632M	10%ImageNet	84.3	96.6

Table 10: The comparison with the state-of-the-art fully supervised models.

these dataset, our Semi-ViT can improve over the fine-tuning baselines by 13-21 (7-10) points on 1% (10%) labels. Note that on Food-101, Semi-ViT on 1% (10%) labels is close to fine-tuning baseline on 10% (100%) labels, *i.e.*, 82.1 v.s. 84.5 (91.3 v.s. 93.1), indicating that using Semi-ViT can help to save annotation costs by about 10 times on this dataset.

5 Related Work

Semi-supervised learning has a long history of research [78, 13]. The recent works can be roughly clustered into two groups, consistency-based [39, 62, 50, 70, 66] and pseudo-labeling based [40, 55, 53, 10]. Consistency-based methods usually add some noise to the input or model, and then enforce their feature or probability outputs to be consistent. For example, to construct two outputs for later consistency regularization, Π-model [39] adds noise to the model weights using dropout [57], Mean-teacher [62] builds a teacher model by EMA updated from the student model, and UDA [70] applies a weak and a strong data augmentation to the input. On the other hand, the idea of pseudo-labeling or self-training can be traced back to [34, 49], which uses model predictions as hard pseudo labels to guide the learning on unlabeled data. This idea becomes popular in SSL recently [40, 55, 53, 10, 71], and some theoretical explanations are available [76, 24]. In the offline pseudo labeling [40, 71], the model used to generate pseudo labels is usually frozen or updated once in a while during training, *e.g.*, at the end of every training epoch, but for online pseudo-labeling [55, 10] the teacher model is updated continuously along with the student. Beyond classification, pseudo-labeling has also achieved promising progresses in more challenging tasks, *e.g.*, object detection [56, 43, 67]. Our Semi-ViT also falls into the category of online pseudo-labeling.

Dataset	# train/test	# class	Method	1%	10%	100%
Food-101 [9]	75.7K/25.2K	101	Finetune	60.9	84.5	93.1
			Semi-ViT	82.1	91.3	-
iNaturalist [30]	265K/3K	1010	Finetune	19.6	57.3	81.2
			Semi-ViT	32.3	67.7	-
GoogleLandmark [52]	200K/15.6K	256	Finetune	45.3	74.0	91.5
			Semi-ViT	61.0	81.0	-

Table 11: The Semi-ViT-Base results on other datasets.

Mixup [75] is an effective data augmentation technique, which interpolates the input samples and their labels linearly and performs vicinal risk minimization. It has been successfully used in image classification and some other domains, *e.g.*, generative adversarial networks [47], sentence classification [23], etc. Other variants have also been developed, *e.g.*, Manifold Mixup [65] that mixes up in the feature space or CutMix [74] which cuts a patch from one image and pastes it into another one. Mixup has also been successfully adopted in self-supervised learning [37, 41] and semi-supervised learning [7, 66, 6]. Although [7, 66, 6] also used mixup for SSL, they have differences with our *probabilistic pseudo mixup*: 1) they are consistency-based SSL framework, but ours is pseudo-labeling based; 2) their mixup ratio is random sampled, but ours depends on the pseudo label confidence; 3) they have only shown successes on small CNN architectures and small datasets, *e.g.*, CIFAR [38] and SVHN [51], but our successes are built on various scales of transformer architectures and large-scale datasets, *e.g.*, ImageNet [54], INaturalist [30], GoogleLandmark [52], etc.

6 Conclusion

In this paper, we propose Semi-ViT for vision transformers based semi-supervised learning. This is the first time that *pure* vision transformers can achieve promising results on semi-supervised learning and even surpass the previous best CNN based counterparts by a large margin. In addition, Semi-ViT inherits the scalable benefits from ViT, and the larger model leads to smaller gap to the fully supervised upper-bounds. This has shown to be a promising direction for semi-supervised learning. And the advantages of Semi-ViT can be well generalized to other datasets, suggesting potentially broader impacts. We hope these promising results could encourage more efforts in semi-supervised vision transformers.

Limitations Our paper only considers the standard semi-supervised classification settings where the full dataset, *e.g.*, ImageNet, is downsampled to smaller scales, and not the advanced settings where full ImageNet is used as labeled and additional data, *e.g.*, ImageNet-21K, is used as unlabeled. And we have only evaluated on classification task. It is unclear whether the same conclusion can hold to that advanced classification setting and more challenging tasks, *e.g.*, detection or segmentation.

Potential Negative Social Impacts Semi-ViT has shown that strong models can be obtained with only a few labels, *e.g.*, 1%. This increases the good AI models accessibility to anyone, which could potentially lead to inappropriate use of them.

References

[1] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lucic, and Cordelia Schmid. Vivit: A video vision transformer. In *ICCV*, pages 6816–6826. IEEE, 2021.

[2] Mahmoud Assran, Mathilde Caron, Ishan Misra, Piotr Bojanowski, Armand Joulin, Nicolas Ballas, and Michael G. Rabbat. Semi-supervised learning of visual features by non-parametrically predicting view assignments with support samples. In *ICCV*, pages 8423–8432. IEEE, 2021.

[3] Ben Athiwaratkun, Marc Finzi, Pavel Izmailov, and Andrew Gordon Wilson. There are many consistent explanations of unlabeled data: Why you should average. In *ICLR*. OpenReview.net, 2019.

[4] Lei Jimmy Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization. *CoRR*, abs/1607.06450, 2016.

[5] Hangbo Bao, Li Dong, and Furu Wei. Beit: Bert pre-training of image transformers. *arXiv preprint arXiv:2106.08254*, 2021.

[6] David Berthelot, Nicholas Carlini, Ekin D. Cubuk, Alex Kurakin, Kihyuk Sohn, Han Zhang, and Colin Raffel. Remixmatch: Semi-supervised learning with distribution matching and augmentation anchoring. In *ICLR*. OpenReview.net, 2020.

[7] David Berthelot, Nicholas Carlini, Ian J. Goodfellow, Nicolas Papernot, Avital Oliver, and Colin Raffel. Mixmatch: A holistic approach to semi-supervised learning. In *NeurIPS*, pages 5050–5060, 2019.

[8] Lucas Beyer, Xiaohua Zhai, Avital Oliver, and Alexander Kolesnikov. S4L: self-supervised semi-supervised learning. In *ICCV*, pages 1476–1485. IEEE, 2019.

[9] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101 - mining discriminative components with random forests. In *ECCV*, volume 8694 of *Lecture Notes in Computer Science*, pages 446–461. Springer, 2014.

[10] Zhaowei Cai, Avinash Ravichandran, Subhransu Maji, Charless C. Fowlkes, Zhuowen Tu, and Stefano Soatto. Exponential moving average normalization for self-supervised and semi-supervised learning. In *CVPR*, pages 194–203, 2021.

[11] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, volume 12346 of *Lecture Notes in Computer Science*, pages 213–229. Springer, 2020.

[12] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *ICCV*, pages 9630–9640. IEEE, 2021.

[13] Olivier Chapelle, Bernhard Scholkopf, and Alexander Zien. Semi-supervised learning (chapelle, o. et al., eds.; 2006). *IEEE Transactions on Neural Networks*, 20(3):542–542, 2009.

[14] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, volume 119 of *Proceedings of Machine Learning Research*, pages 1597–1607. PMLR, 2020.

[15] Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey E. Hinton. Big self-supervised models are strong semi-supervised learners. In *NeurIPS*, 2020.

[16] Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised vision transformers. In *ICCV*, pages 9620–9629. IEEE, 2021.

[17] Ekin Dogus Cubuk, Barret Zoph, Jonathon Shlens, and Quoc Le. Randaugment: Practical automated data augmentation with a reduced search space. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *NeurIPS*, 2020.

[18] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*. OpenReview.net, 2021.

[19] Haoqi Fan, Bo Xiong, Karttikeya Mangalam, Yanghao Li, Zhicheng Yan, Jitendra Malik, and Christoph Feichtenhofer. Multiscale vision transformers. In *ICCV*, pages 6804–6815. IEEE, 2021.

[20] Geoff French, Avital Oliver, and Tim Salimans. Milking cowmask for semi-supervised image classification. In *VISIGRAPP*, pages 75–84. SCITEPRESS, 2022.

[21] Priya Goyal, Piotr Dollár, Ross B. Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch SGD: training imagenet in 1 hour. *CoRR*, abs/1706.02677, 2017.

[22] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Ávila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap your own latent - A new approach to self-supervised learning. In *NeurIPS*, 2020.

[23] Hongyu Guo. Nonlinear mixup: Out-of-manifold data augmentation for text classification. In *AAAI*, pages 4044–4051. AAAI Press, 2020.

[24] Haiyun He, Hanshu Yan, and Vincent YF Tan. Information-theoretic characterization of the generalization error for iterative semi-supervised learning. *Journal of Machine Learning Research*, 23:1–52, 2022.

[25] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. *arXiv preprint arXiv:2111.06377*, 2021.

[26] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross B. Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, pages 9726–9735. Computer Vision Foundation / IEEE, 2020.

[27] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778. IEEE Computer Society, 2016.

[28] Yosuke Higuchi, Niko Moritz, Jonathan Le Roux, and Takaaki Hori. Momentum pseudo-labeling for semi-supervised speech recognition. In *Interspeech*, pages 726–730. ISCA, 2021.

[29] Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. *CoRR*, abs/1503.02531, 2015.

[30] Grant Van Horn, Oisin Mac Aodha, Yang Song, Alexander Shepard, Hartwig Adam, Pietro Perona, and Serge J. Belongie. The inaturalist challenge 2017 dataset. *CoRR*, abs/1707.06642, 2017.

[31] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *CVPR*, pages 7132–7141. Computer Vision Foundation / IEEE Computer Society, 2018.

[32] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks. In *CVPR*, pages 2261–2269. IEEE Computer Society, 2017.

[33] Gao Huang, Yu Sun, Zhuang Liu, Daniel Sedra, and Kilian Q. Weinberger. Deep networks with stochastic depth. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *ECCV*, volume 9908 of *Lecture Notes in Computer Science*, pages 646–661. Springer, 2016.

[34] H. J. Scudder III. Probability of error of some adaptive pattern-recognition machines. *IEEE Trans. Inf. Theory*, 11(3):363–371, 1965.

[35] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, volume 37 of *JMLR Workshop and Conference Proceedings*, pages 448–456. JMLR.org, 2015.

[36] Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry P. Vetrov, and Andrew Gordon Wilson. Averaging weights leads to wider optima and better generalization. In *UAI*, pages 876–885. AUAI Press, 2018.

[37] Yannis Kalantidis, Mert Bülent Sariyildiz, Noé Pion, Philippe Weinzaepfel, and Diane Larlus. Hard negative mixing for contrastive learning. In *NeurIPS*, 2020.

[38] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[39] Samuli Laine and Timo Aila. Temporal ensembling for semi-supervised learning. In *ICLR*, 2017.

[40] Dong-Hyun Lee. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, volume 3, 2013.

[41] Kibok Lee, Yian Zhu, Kihyuk Sohn, Chun-Liang Li, Jinwoo Shin, and Honglak Lee. i-mix: A domain-agnostic strategy for contrastive representation learning. In *ICLR*. OpenReview.net, 2021.

[42] Tatiana Likhomanenko, Qiantong Xu, Jacob Kahn, Gabriel Synnaeve, and Ronan Collobert. slimipl: Language-model-free iterative pseudo-labeling. In *Interspeech*, pages 741–745. ISCA, 2021.

[43] Yen-Cheng Liu, Chih-Yao Ma, Zijian He, Chia-Wen Kuo, Kan Chen, Peizhao Zhang, Bichen Wu, Zsolt Kira, and Peter Vajda. Unbiased teacher for semi-supervised object detection. In *ICLR*. OpenReview.net, 2021.

[44] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, pages 9992–10002. IEEE, 2021.

[45] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. *arXiv preprint arXiv:2201.03545*, 2022.

[46] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*. OpenReview.net, 2019.

[47] Thomas Lucas, Corentin Tallec, Yann Ollivier, and Jakob Verbeek. Mixed batches and symmetric discriminators for GAN training. In *ICML*, volume 80 of *Proceedings of Machine Learning Research*, pages 2850–2859. PMLR, 2018.

[48] Vimal Manohar, Tatiana Likhomanenko, Qiantong Xu, Wei-Ning Hsu, Ronan Collobert, Yatharth Saraf, Geoffrey Zweig, and Abdelrahman Mohamed. Kaizen: Continuously improving teacher using exponential moving average for semi-supervised speech recognition. In *ASRU*, pages 518–525. IEEE, 2021.

[49] Geoffrey J McLachlan. Iterative reclassification procedure for constructing an asymptotically optimal rule of allocation in discriminant analysis. *Journal of the American Statistical Association*, 70(350):365–369, 1975.

[50] Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, and Shin Ishii. Virtual adversarial training: A regularization method for supervised and semi-supervised learning. *IEEE Trans. Pattern Anal. Mach. Intell.*, 41(8):1979–1993, 2019.

[51] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. 2011.

[52] Hyeonwoo Noh, Andre Araujo, Jack Sim, Tobias Weyand, and Bohyung Han. Large-scale image retrieval with attentive deep local features. In *ICCV*, pages 3476–3485. IEEE Computer Society, 2017.

[53] Hieu Pham, Zihang Dai, Qizhe Xie, and Quoc V. Le. Meta pseudo labels. In *CVPR*, pages 11557–11568. Computer Vision Foundation / IEEE, 2021.

[54] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael S. Bernstein, Alexander C. Berg, and Li Fei-Fei. Imagenet large scale visual recognition challenge. *Int. J. Comput. Vis.*, 115(3):211–252, 2015.

[55] Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. In *NeurIPS*, 2020.

[56] Kihyuk Sohn, Zizhao Zhang, Chun-Liang Li, Han Zhang, Chen-Yu Lee, and Tomas Pfister. A simple semi-supervised learning framework for object detection. *arXiv preprint arXiv:2005.04757*, 2020.

[57] Nitish Srivastava, Geoffrey E. Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, 15(1):1929–1958, 2014.

[58] Robin Strudel, Ricardo Garcia Pinel, Ivan Laptev, and Cordelia Schmid. Segmenter: Transformer for semantic segmentation. In *ICCV*, pages 7242–7252. IEEE, 2021.

[59] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A. Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *AAAI*, pages 4278–4284. AAAI Press, 2017.

[60] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *CVPR*, pages 2818–2826. IEEE Computer Society, 2016.

[61] Mingxing Tan and Quoc V. Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *ICML*, volume 97 of *Proceedings of Machine Learning Research*, pages 6105–6114. PMLR, 2019.

[62] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *NeurIPS*, pages 1195–1204, 2017.

[63] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *ICML*, volume 139 of *Proceedings of Machine Learning Research*, pages 10347–10357. PMLR, 2021.

[64] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, pages 5998–6008, 2017.

[65] Vikas Verma, Alex Lamb, Christopher Beckham, Amir Najafi, Ioannis Mitliagkas, David Lopez-Paz, and Yoshua Bengio. Manifold mixup: Better representations by interpolating hidden states. In *ICML*, volume 97 of *Proceedings of Machine Learning Research*, pages 6438–6447. PMLR, 2019.

[66] Vikas Verma, Alex Lamb, Juho Kannala, Yoshua Bengio, and David Lopez-Paz. Interpolation consistency training for semi-supervised learning. In *IJCAI*, pages 3635–3641. ijcai.org, 2019.

[67] Pei Wang, Zhaowei Cai, Hao Yang, Gurumurthy Swaminathan, Nuno Vasconcelos, Bernt Schiele, and Stefano Soatto. Omni-DETR: Omni-supervised object detection with transformers. In *CVPR*, 2022.

[68] Zejia Weng, Xitong Yang, Ang Li, Zuxuan Wu, and Yu-Gang Jiang. Semi-supervised vision transformers. *arXiv preprint arXiv:2111.11067*, 2021.

[69] Ross Wightman. Pytorch image models. `https://github.com/rwightman/pytorch-image-models`, 2019.

[70] Qizhe Xie, Zihang Dai, Eduard H. Hovy, Thang Luong, and Quoc Le. Unsupervised data augmentation for consistency training. In *NeurIPS*, 2020.

[71] Qizhe Xie, Minh-Thang Luong, Eduard H. Hovy, and Quoc V. Le. Self-training with noisy student improves imagenet classification. In *CVPR*, pages 10684–10695. Computer Vision Foundation / IEEE, 2020.

[72] Saining Xie, Ross B. Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *CVPR*, pages 5987–5995. IEEE Computer Society, 2017.

[73] Weijian Xu, Yifan Xu, Tyler A. Chang, and Zhuowen Tu. Co-scale conv-attentional image transformers. In *ICCV*, pages 9961–9970. IEEE, 2021.

[74] Sangdoo Yun, Dongyoon Han, Sanghyuk Chun, Seong Joon Oh, Youngjoon Yoo, and Junsuk Choe. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *ICCV*, pages 6022–6031. IEEE, 2019.

[75] Hongyi Zhang, Moustapha Cissé, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *ICLR*. OpenReview.net, 2018.

[76] Shuai Zhang, Meng Wang, Sijia Liu, Pin-Yu Chen, and Jinjun Xiong. How does unlabeled data improve generalization in self-training? a one-hidden-layer theoretical analysis. *arXiv preprint arXiv:2201.08514*, 2022.

[77] Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. In *AAAI*, pages 13001–13008. AAAI Press, 2020.

[78] Xiaojin Jerry Zhu. Semi-supervised learning literature survey. Technical report, University of Wisconsin-Madison Department of Computer Sciences, 2005.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] See the abstract and introduction sections.
   (b) Did you describe the limitations of your work? [Yes] See the conclusion section.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] See the conclusion section.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] See the supplemental material. The code will be released upon acceptance.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See the supplemental material. We have tried our best to provide all experimental details. And the code will be released upon acceptance for reproduction purpose.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] See the supplemental material.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See the supplemental material.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] See the references.
   (b) Did you mention the license of the assets? [No] Those are common assets.
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [No] Those are common assets.
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

A Implementation Details

Dataset Sampling We sample 1% (10%) images per class from the datasets we use for the semi-supervised learning experiments of 1% (10%) labels. For example, on ImageNet, the number of sampled images is 12,820 (128,118) in total, for 1% (10%) labels.

ViT Architectures We use exactly same architecture as the standard ViT [18]. For the position tokens, we use the learnable positional embedding when the model is self-pretrained, but the sine-cosine version of positional embedding (non-learnable) when not self-pretrained since we find it leads to better results than the learnable one. For all architectures, the classifier is built on top of the average pooling of the encoder output, except for ViT-Huge where we find the classifier on top of the output of the class token leads to higher performances.

Data Augmentation We use the common data augmentations of `RandomResizedCrop`, `RandomHorizontalFlip`, `RandAugment('m9-mstd0.5-inc1')` [17] and `RandomErasing` [77] on the labeled data. The same augmentations are also used on the unlabeled data as the strong augmentation in EMA-Teacher, whereas the weak augmentations are `RandomResizedCrop`, `RandomHorizontalFlip` and `ColorJitter(0.4)`. We do not extensively explore the space of data augmentation for weak and strong augmentations. The center 224×224 crop is used at inference.

Self-supervised Pretraining We directly use the pretrained models from MAE [25], DINO [16] and MoCo-v3 [12]. The final Semi-ViT-Small is with DINO self-pretraining.

Supervised Fine-tuning Settings The settings for the stage of *supervised fine-tuning*, with and without self-pretraining, are shown in Table 12. These settings are mainly following the settings of finetuning and learning from scratch in [25], with some minor modifications. We also use the linear learning rate scaling rule [21]: $lr = base_lr \times batchsize/256$. When supervised fine-tuning on 1% data, we find the performances are bad when the regularization is strong, hence we do not use mixup/cutmix, drop path and random erasing, in that case, except ViT-Small.

Semi-supervised Fine-tuning Settings The settings for the stage of *semi-supervised fine-tuning*, with and without self-pretraining, are shown in Table 13. When semi-supervised fine-tuning small models (ViT-Small/Base) on 1% data, we find the performances are bad when the regularization is strong, hence we do not use mixup/cutmix on labeled data, and drop path and random erasing on both labeled and unlabeled data, in that case. The linear learning rate scaling rule is: $lr = base_lr \times batchsize_l/256$, where $batchsize_l$ is the batch size of the labeled data.

Confidence Threshold The confidence thresholds of Semi-ViT are shown in Table 13, even though sometimes those are not optimal according to the ablation study in Table 4.

Computing Resources We run all experiments on V100 GPUs of 32G memory. For Semi-ViT-Small/Base/Large/Huge, we use 8/8/16/32 GPUs for training 100/100/100/50 epochs, which takes about 19/31/61/115 hours.

Random Seeds and Error Bar Since some of the experiments are expensive to run, e.g., Semi-ViT-Huge. We only test the randomness on Semi-ViT-Base models. We sample three different 10%/1% subsets of ImageNet and repeat the experiments for three times with different random seeds. When self-pretrained by MAE, the accuracy is 79.71 ± 0.037 (70.95 ± 0.029) for 10% (1%) labels. When not self-pretrained, the accuracy is 73.44 ± 0.065 for 10% labels. The results have shown that Semi-ViT is quite robust to different random seeds and different subsets of the samples. To keep consistency, all experiments in the paper are run with the same ImageNet subset and the same random seed.

Settings on Other Datasets The model is ViT-Base on the datasets of Food-101 [9], iNaturalist [30] and GoogleLandmark [52]. The *supervised fine-tuning* settings are almost the same as those with self-pretraining in Table 12, with the differences: 1) we search the optimal base learning rate from $\{1e^{-4}, 2.5e^{-4}, 5e^{-4}, 1e^{-3}\}$ and the optimal layer-wise learning rate decay from $\{0.65, 0.75\}$;

config	10% labels	1% labels	10% labels (scratch)
optimizer	AdamW	AdamW	AdamW
base learning rate	1e-4 (S), 2.5e-4 (B) 1e-3 (L/H)	1e-4 (S), 5e-5 (B) 1e-3 (L), 0.01 (H)	1e-4
weight decay	0.05	0.05	0.3
optimizer momentum	β_1, β_2=0.9, 0.999	β_1, β_2=0.9, 0.999	β_1, β_2=0.9, 0.95
layer-wise lr decay [5]	0.65 (S/B), 0.75 (L/H)	0.65 (S/B), 0.75 (L/H)	1.0
batch size	512 (S/B/L), 256 (H)	512 (S/B/L), 128 (H)	1024
learning rate schedule	cosine decay	cosine decay	cosine decay
warmup epochs	5	5	50
training epochs	100 (S/B), 50 (L/H)	100 (S/B), 50 (L/H)	500
label smoothing [60]	0.1	0.1	0.1
mixup [75]	0.8	0.8 (S), 0 (B/L/H)	0.8
cutmix [74]	1.0	1.0 (S), 0 (B/L/H)	1.0
drop path [33]	0.1 (S/B/L) 0.2 (H)	0.1 (S), 0 (B/L/H)	0.1
random erasing [77]	0.25	0.25 (S), 0 (B/L/H)	0.25

Table 12: Supervised fine-tuning settings with and without self-pretraining.

config	10% labels	1% labels	10% labels (scratch)
optimizer	AdamW	AdamW	AdamW
base learning rate	2e-4 (S), 1e-3 (B) 2e-3 (L), 2.5e-3 (H)	5e-4 (S), 1e-3 (B/L) 5e-3 (H)	1e-3
weight decay	0.05	0.05	0.05
optimizer momentum	β_1, β_2=0.9, 0.999	β_1, β_2=0.9, 0.999	β_1, β_2=0.9, 0.999
layer-wise lr decay [5]	0.75	0.75	0.85
batch size (labeled)	128	128 (S/B/L), 64 (H)	128
learning rate schedule	cosine decay	cosine decay	cosine decay
confidence threshold	0.5 (S/B), 0.6 (L/H)	0.6	0.5
warmup epochs	5	5	5
training epochs	100 (S/B/L), 50 (H)	100 (S/B/L), 50 (H)	100
label smoothing [60]	0.1	0.1	0.1
mixup [75]	0.8	0.8	0.8
cutmix [74]	1.0	1.0	1.0
drop path [33]	0.1 (S/B/H), 0.2 (L)	0 (S/B), 0.1 (L), 0.05 (H)	0.1
random erasing [77]	0.25	0 (S/B), 0.25 (L/H)	0.25

Table 13: Semi-supervised fine-tuning settings with and without self-pretraining.

2) we enable mixup/cutmix, drop path and random erasing for 1% experiments (except iNaturalist), with the same values of those for 10% experiments. The *semi-supervised fine-tuning* settings are almost the same as those with self-pretraining in Table 13, with the difference that we set the EMA momentum decay $m = 0.999$ instead of $m = 0.9999$ as in the ImageNet experiments, since these datasets are smaller and need faster EMA update rate.