Big Self-Supervised Models are Strong Semi-Supervised Learners

Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, Geoffrey Hinton
Google Research, Brain Team

Abstract

One paradigm for learning from few labeled examples while making best use of a large amount of unlabeled data is *unsupervised pretraining* followed by *supervised fine-tuning*. Although this paradigm uses unlabeled data in a *task-agnostic* way, in contrast to common approaches to semi-supervised learning for computer vision, we show that it is surprisingly effective for semi-supervised learning on ImageNet. A key ingredient of our approach is the use of big (deep and wide) networks during pretraining and fine-tuning. We find that, the fewer the labels, the more this approach (task-agnostic use of unlabeled data) benefits from a bigger network. After fine-tuning, the big network can be further improved and distilled into a much smaller one with little loss in classification accuracy by using the unlabeled examples for a second time, but in a *task-specific* way. The proposed semi-supervised learning algorithm can be summarized in three steps: *unsupervised pretraining* of a big ResNet model using SimCLRv2, *supervised fine-tuning* on a few labeled examples, and *distillation with unlabeled examples* for refining and transferring the task-specific knowledge. This procedure achieves 73.9% ImageNet top-1 accuracy with just 1% of the labels ($\leq$13 labeled images per class) using ResNet-50, a $10\times$ improvement in label efficiency over the previous state-of-the-art. With 10% of labels, ResNet-50 trained with our method achieves 77.5% top-1 accuracy, outperforming standard supervised training with all of the labels. [1]

1 Introduction

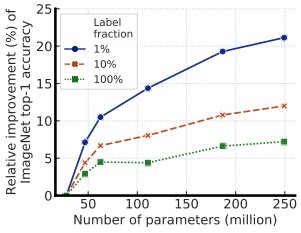


Figure 1: Bigger models yield larger gains when fine-tuning with fewer labeled examples.

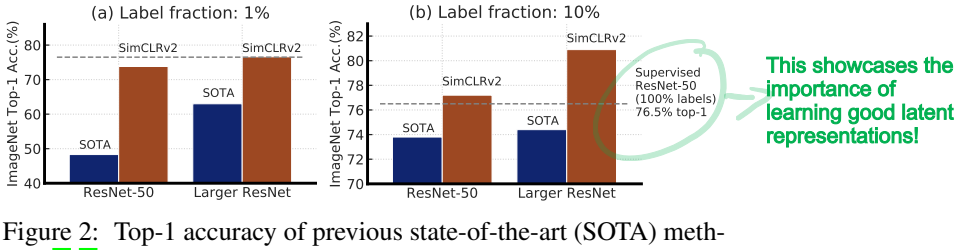


Figure 2: Top-1 accuracy of previous state-of-the-art (SOTA) methods [1, 2] and our method (SimCLRv2) on ImageNet using only 1% or 10% of the labels. Dashed line denotes fully supervised ResNet-50 trained with 100% of labels. Full comparisons in Table 3.

Learning from just a few labeled examples while making best use of a large amount of unlabeled data is a long-standing problem in machine learning. One approach to semi-supervised learning involves unsupervised or self-supervised pretraining, followed by supervised fine-tuning [3, 4]. This

Correspondence to: iamtingchen@google.com

[1] Code and pretrained checkpoints are available at https://github.com/google-research/simclr.

34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada.

The fundamental idea here is to penalize the network for learning different representations for data we can confidently define as similar (e.g. Image A = Image B rotated 90 degrees. The learned representations for image A and image B should be the same).

approach leverages unlabeled data in a *task-agnostic* way during pretraining, as the supervised labels are only used during fine-tuning. Although it has received little attention in computer vision, this approach has become predominant in natural language processing, where one first trains a large language model on unlabeled text (e.g., Wikipedia), and then fine-tunes the model on a few labeled examples [5–10]. An alternative approach, common in computer vision, directly leverages unlabeled data during supervised learning, as a form of regularization. This approach uses unlabeled data in a *task-specific* way to encourage class label prediction consistency on unlabeled data among different models [11, 12, 2] or under different data augmentations [13–15].

Motivated by recent advances in self-supervised learning of visual representations [16–20, 1], this paper first presents a thorough investigation of the "unsupervised pretrain, supervised fine-tune" paradigm for semi-supervised learning on ImageNet [21]. During self-supervised pretraining, images are used without class labels (in a task-agnostic way), hence the representations are not directly tailored to a specific classification task. With this task-agnostic use of unlabeled data, we find that network size is important: Using a big (deep and wide) neural network for self-supervised pretraining and fine-tuning greatly improves accuracy. In addition to the network size, we characterize a few important design choices for contrastive representation learning that benefit supervised fine-tuning and semi-supervised learning.

Once a convolutional network is pretrained and fine-tuned, we find that its task-specific predictions can be further improved and distilled into a smaller network. To this end, we make use of unlabeled data for a second time to encourage the student network to mimic the teacher network's label predictions. Thus, the distillation [22, 23] phase of our method using unlabeled data is reminiscent of the use of pseudo labels [11] in self-training [24, 12], but without much extra complexity.

Confused by "without much extra complexity". I hope this is formalized later in the paper.

In summary, the proposed semi-supervised learning framework comprises three steps as shown in Figure 3: (1) unsupervised or self-supervised pretraining, (2) supervised fine-tuning, and (3) distillation using unlabeled data. We develop an improved variant of a recently proposed contrastive learning framework, SimCLR [1], for unsupervised pretraining of a ResNet architecture [25]. We call this framework SimCLRv2. We assess the effectiveness of our method on ImageNet ILSVRC-2012 [21] with only 1% and 10% of the labeled images available. Our main findings and contributions can be summarized as follows:

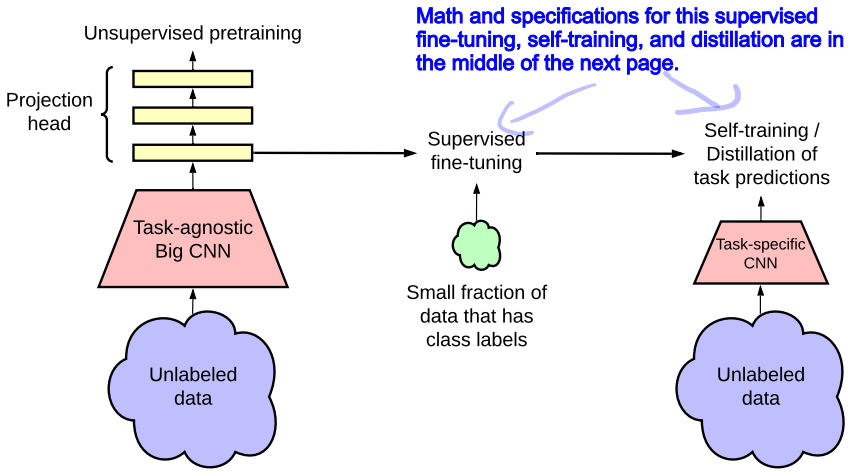
Really nice summary of the contributions of this paper. This particularly excites me about the potential impact given the impressive distillation process for transferring general representations to task specific applications.

- Our empirical results suggest that for semi-supervised learning (via the task-agnostic use of unlabeled data), the fewer the labels, the more it is possible to benefit from a bigger model (Figure 1). Bigger self-supervised models are more label efficient, performing significantly better when fine-tuned on only a few labeled examples, even though they have more capacity to potentially overfit.
- We show that although big models are important for learning general (visual) representations, the extra capacity may not be necessary when a specific target task is concerned. Therefore, with the task-specific use of unlabeled data, the predictive performance of the model can be further improved and transferred into a smaller network.
- We further demonstrate the importance of the nonlinear transformation (a.k.a. projection head) after convolutional layers used in SimCLR for semi-supervised learning. A deeper projection head not only improves the representation quality measured by linear evaluation, but also improves semi-supervised performance when fine-tuning from a *middle layer* of the projection head.

We combine these findings to achieve a new state-of-the-art in semi-supervised learning on ImageNet as summarized in Figure 2. Under the linear evaluation protocol, SimCLRv2 achieves 79.8% top-1 accuracy, a 4.3% relative improvement over the previous state-of-the-art [1]. When fine-tuned on only 1% / 10% of labeled examples and distilled to the same architecture using unlabeled examples, it achieves 76.6% / 80.9% top-1 accuracy, which is a 21.6% / 8.7% relative improvement over previous state-of-the-art. With distillation, these improvements can also be transferred to smaller ResNet-50 networks to achieve 73.9% / 77.5% top-1 accuracy using 1% / 10% of labels. By comparison, a standard supervised ResNet-50 trained on all of labeled images achieves a top-1 accuracy of 76.6%.

2 Method

Inspired by the recent successes of learning from unlabeled data [19, 20, 1, 11, 24, 12], the proposed semi-supervised learning framework leverages unlabeled data in both task-agnostic and task-specific



Math and specifications for this supervised fine-tuning, self-training, and distillation are in the middle of the next page.

Figure 3: The proposed semi-supervised learning framework leverages unlabeled data in two ways: (1) task-agnostic use in unsupervised pretraining, and (2) task-specific use in self-training / distillation.

ways. The first time the unlabeled data is used, it is in a task-agnostic way, for learning general (visual) representations via unsupervised pretraining. The general representations are then adapted for a specific task via supervised fine-tuning. The second time the unlabeled data is used, it is in a task-specific way, for further improving predictive performance and obtaining a compact model. To this end, we train student networks on the unlabeled data with imputed labels from the fine-tuned teacher network. Our method can be summarized in three main steps: *pretrain*, *fine-tune*, and then *distill*. The procedure is illustrated in Figure 3. We introduce each specific component in detail below.

SimCLR cleverly combines multiple augmentation methods into one positive sample transformation. This is more robust than simple rotations for learning invariant representations.

Self-supervised pretraining with SimCLRv2. To learn general visual representations effectively with unlabeled images, we adopt and improve SimCLR [1], a recently proposed approach based on contrastive learning. SimCLR learns representations by maximizing agreement [26] between differently augmented views of the same data example via a contrastive loss in the latent space. More specifically, given a randomly sampled mini-batch of images, each image $\boldsymbol{x}_i$ is augmented twice using random crop, color distortion and Gaussian blur, creating two views of the same example $\boldsymbol{x}_{2k-1}$ and $\boldsymbol{x}_{2k}$. The two images are encoded via an encoder network $f(\cdot)$ (a ResNet [25]) to generate representations $\boldsymbol{h}_{2k-1}$ and $\boldsymbol{h}_{2k}$. The representations are then transformed again with a non-linear transformation network $g(\cdot)$ (a MLP projection head), yielding $\boldsymbol{z}_{2k-1}$ and $\boldsymbol{z}_{2k}$ that are used for the contrastive loss. With a mini-batch of augmented examples, the contrastive loss between a pair of positive example i, j (augmented from the same image) is given as follows:

An additional transformation on the latent space is something I have personally found useful when regularizing representations with contrastive loss. Happy to see it here (and in the original SimCLR).

$$\ell_{i,j}^{\text{NT-Xent}} = -\log \frac{\exp(\text{sim}(\boldsymbol{z}_i, \boldsymbol{z}_j)/\tau)}{\sum_{k=1}^{2N} \mathbb{1}_{[k \neq i]} \exp(\text{sim}(\boldsymbol{z}_i, \boldsymbol{z}_k)/\tau)}, \tag{1}$$

Where $\text{sim}(\cdot, \cdot)$ is cosine similarity between two vectors, and τ is a temperature scalar.

In this work, we propose SimCLRv2, which improves upon SimCLR [1] in three major ways. Below we summarize the changes as well as their improvements of accuracy on Imagenet ILSVRC-2012 [21].

1. To fully leverage the power of general pretraining, we explore larger ResNet models. Unlike SimCLR [1] and other previous work [27, 20], whose largest model is ResNet-50 ($4\times$), we train models that are deeper but less wide. The largest model we train is a 152-layer ResNet [25] with $3\times$ wider channels and selective kernels (SK) [28], a channel-wise attention mechanism that improves the parameter efficiency of the network. By scaling up the model from ResNet-50 to ResNet-152 ($3\times$+SK), we obtain a 29% relative improvement in top-1 accuracy when fine-tuned on 1% of labeled examples.
2. We also increase the capacity of the non-linear network $g(\cdot)$ (a.k.a. projection head), by making it deeper.[2] Furthermore, instead of throwing away $g(\cdot)$ entirely after pretraining as in SimCLR [1],

[2] In our experiments, we set the width of projection head's middle layers to that of its input, so it is also adjusted by the width multiplier. However, a wider projection head improves performance even when the base network remains narrow.

The projection head is further compression from the latent space to another latent space specifically used for contrastive loss. This means that the representations transferred to the task specific application in SimCLR aren't fully regularized via contrastive loss. I find this confusing, so I consulted the original SimCLR paper. The explanation to throwing away the projection head can be found in section 4.2, and boils down to noticeable performance increases by using just the encoder on inference. The authors conjecture that this is because the projection head is designed to enforce invariance to augmentation, and hence can remove information related to important features relevant to downstream tasks (e.g. color). While this argument is logically sound, the whole purpose of contrastive loss with data augmentation is to enforce robust and invariant representations. So I'm a bit uneasy that this was the original approach, and am happy this new version doesn't entirely throw away this projection head.

The addition of a single layer in this projection head is way more significant than I expected. I wonder why? Given the impressive increase here, I am happy to see this paper explore a fourth layer in a later section.

we fine-tune from a middle layer (detailed later). This small change yields a significant improvement for both linear evaluation and fine-tuning with only a few labeled examples. Compared to SimCLR with 2-layer projection head, by using a 3-layer projection head and fine-tuning from the 1st layer of projection head, it results in as much as 14% relative improvement in top-1 accuracy when fine-tuned on 1% of labeled examples (see Figure E.1).

3. Motivated by [29], we also incorporate the memory mechanism from MoCo [20], which designates a memory network (with a moving average of weights for stabilization) whose output will be buffered as negative examples. Since our training is based on large mini-batch which already supplies many contrasting negative examples, this change yields an improvement of $\sim$1% for linear evaluation as well as when fine-tuning on 1% of labeled examples (see Appendix D).

Is this just because any image A from the mini-batch and another random image B from the mini-batch are likely of different classes? Or is there some specific mechanism for negative

Fine-tuning. Fine-tuning is a common way to adapt the task-agnostically pretrained network for a specific task. In SimCLR [1], the MLP projection head $g(\cdot)$ is discarded entirely after pretraining, while only the ResNet encoder $f(\cdot)$ is used during the fine-tuning. Instead of throwing it all away, we propose to incorporate part of the MLP projection head into the base encoder during the fine-tuning. In other words, we fine-tune the model from a *middle layer* of the projection head, instead of the input layer of the projection head as in SimCLR. Note that fine-tuning from the first layer of the MLP head is the same as adding an fully-connected layer to the base network and removing an fully-connected layer from the head, and the impact of this extra layer is contingent on the amount of labeled examples during fine-tuning (as shown in our experiments).

Self-training / knowledge distillation via unlabeled examples. To further improve the network for the target task, here we leverage the unlabeled data directly for the target task. Inspired by [23, 11, 22, 24, 12], we use the fine-tuned network as a teacher to impute labels for training a student network. Specifically, we minimize the following distillation loss where no real labels are used:

$$\mathcal{L}^{\text{distill}} = -\sum_{\boldsymbol{x}_i \in \mathcal{D}} \left[\sum_{y} P^T(y|\boldsymbol{x}_i;\tau) \log P^S(y|\boldsymbol{x}_i;\tau) \right] \tag{2}$$

where $P(y|\boldsymbol{x}_i) = \exp(f^{\text{task}}(\boldsymbol{x}_i)[y]/\tau) / \sum_{y'} \exp(f^{\text{task}}(\boldsymbol{x}_i)[y']/\tau)$, and τ is a scalar temperature parameter. The teacher network, which produces $P^T(y|\boldsymbol{x}_i)$, is fixed during the distillation; only the student network, which produces $P^S(y|\boldsymbol{x}_i)$, is trained.

While we focus on distillation using only unlabeled examples in this work, when the number of labeled examples is significant, one can also combine the distillation loss with ground-truth labeled examples using a weighted combination

$$\mathcal{L} = -(1-\alpha) \sum_{(\boldsymbol{x}_i, y_i) \in \mathcal{D}^L} \left[\log P^S(y_i|\boldsymbol{x}_i) \right] - \alpha \sum_{\boldsymbol{x}_i \in \mathcal{D}} \left[\sum_{y} P^T(y|\boldsymbol{x}_i;\tau) \log P^S(y|\boldsymbol{x}_i;\tau) \right]. \tag{3}$$

This procedure can be performed using students either with the same model architecture (self-distillation), which further improves the task-specific performance, or with a smaller model architecture, which leads to a compact model.

3 Empirical Study

3.1 Settings and Implementation Details

Following the semi-supervised learning setting in [30, 19, 1], we evaluate the proposed method on ImageNet ILSVRC-2012 [21]. While all $\sim$1.28 million images are available, only a randomly sub-sampled 1% (12811) or 10% (128116) of images are associated with labels.[3] As in previous work, we also report performance when training a linear classifier on top of a fixed representation with all labels [31, 16, 17, 1] to directly evaluate SimCLRv2 representations. We use the LARS optimizer [32] (with a momentum of 0.9) throughout for pretraining, fine-tuning and distillation.

Large batch size for CNNs is why LARS is the optimizer of choice here.

For pretraining, similar to [1], we train our model on 128 Cloud TPUs, with a batch size of 4096 and global batch normalization [33], for total of 800 epochs. The learning rate is linearly increased for the first 5% of epochs, reaching maximum of 6.4 ($= 0.1 \times \text{sqrt}(\text{BatchSize})$), and then decayed with a cosine decay schedule. A weight decay of $1e^{-4}$ is used. We use a 3-layer MLP projection head on

[3] See https://www.tensorflow.org/datasets/catalog/imagenet2012_subset for the details of the 1%/10% subsets.

Table 1: Top-1 accuracy of fine-tuning SimCLRv2 models (on varied label fractions) or training a linear classifier on the representations. The supervised baselines are trained from scratch using all labels in 90 epochs. The parameter count only include ResNet up to final average pooling layer. For fine-tuning results with 1% and 10% labeled examples, the models include additional non-linear projection layers, which incurs additional parameter count (4M for $1\times$ models, and 17M for $2\times$ models). See Table H.1 for Top-5 accuracy.

Depth	Width	Use SK [28]	Param (M)	Fine-tuned on 1%	Fine-tuned on 10%	Fine-tuned on 100%	Linear eval	Supervised
50	1×	False	**24**	**57.9**	**68.4**	**76.3**	**71.7**	**76.6**
		True	35	64.5	72.1	78.7	74.6	78.5
	2×	False	94	66.3	73.9	79.1	75.6	77.8
		True	140	70.6	77.0	81.3	77.7	79.3
101	1×	False	43	62.1	71.4	78.2	73.6	78.0
		True	65	68.3	75.1	80.6	76.3	79.6
	2×	False	170	69.1	75.8	80.7	77.0	78.9
		True	257	73.2	78.8	82.4	79.0	80.1
152	1×	False	58	64.0	73.0	79.3	74.5	78.3
		True	89	70.0	76.5	81.3	77.2	79.9
	2×	False	233	70.2	76.6	81.1	77.4	79.1
		True	354	74.2	79.4	82.9	79.4	80.4
152	3×	True	**795**	**74.9**	**80.1**	**83.1**	**79.8**	**80.5**

top of a ResNet encoder. The memory buffer is set to 64K, and exponential moving average (EMA) decay is set to 0.999 according to [20]. We use the same set of simple augmentations as SimCLR [1], namely random crop, color distortion, and Gaussian blur.

For fine-tuning, by default we fine-tune from the first layer of the projection head for 1%/10% of labeled examples, but from the input of the projection head when 100% labels are present. We use global batch normalization, but we remove weight decay, learning rate warmup, and use a much smaller learning rate, i.e. 0.16 ($= 0.005 \times \mathrm{sqrt}(\mathrm{BatchSize})$) for standard ResNets [25], and 0.064 ($= 0.002 \times \mathrm{sqrt}(\mathrm{BatchSize})$) for larger ResNets variants (with width multiplier larger than 1 and/or SK [28]). A batch size of 1024 is used. Similar to [1], we fine-tune for 60 epochs with 1% of labels, and 30 epochs with 10% of labels, as well as full ImageNet labels.

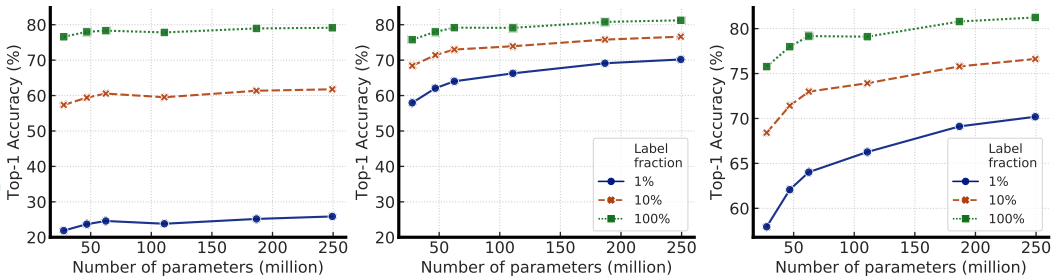
For distillation, we only use unlabeled examples, unless otherwise specified. We consider two types of distillation: self-distillation where the student has the same model architecture as the teacher (excluding projection head), and big-to-small distillation where the student is a much smaller network. We set temperature to 0.1 for self-distillation, and 1.0 for large-to-small distillation (though the effect of temperatures between 0.1 and 1 is very small). We use the same learning rate schedule, weight decay, batch size as pretraining, and the models are trained for 400 epochs. Only random crop and horizontal flips of training images are applied during fine-tuning and distillation.

3.2 Bigger Models Are More Label-Efficient

In order to study the effectiveness of big models, we train ResNet models by varying width and depth as well as whether or not to use selective kernels (SK) [28].[4] Whenever SK is used, we also use the ResNet-D [34] variant of ResNet. The smallest model is the standard ResNet-50, and biggest model is ResNet-152 ($3\times$+SK).

Table 1 compares self-supervised learning and supervised learning under different model sizes and evaluation protocols, including both fine-tuning and linear evaluation. We can see that increasing width and depth, as well as using SK, all improve the performance. These architectural manipulations have relatively limited effects for standard supervised learning (4% differences in smallest and largest models), but for self-supervised models, accuracy can differ by as much as 8% for linear evaluation, and 17% for fine-tuning on 1% of labeled images. We also note that ResNet-152 ($3\times$+SK) is only marginally better than ResNet-152 ($2\times$+SK), though the parameter size is almost doubled, suggesting that the benefits of width may have plateaued.

[4] Although we do not use grouped convolution in this work, we believe it can further improve parameter efficiency.



(a) Supervised (b) Semi-supervised (c) Semi-supervised (y-axis zoomed)

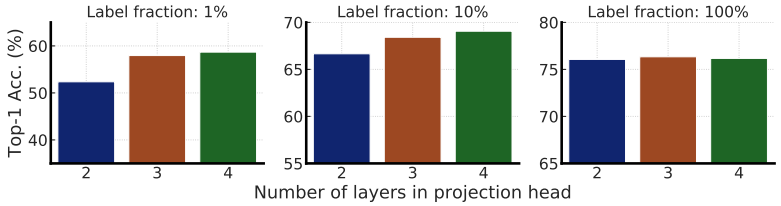
Figure 4: Top-1 accuracy for supervised vs semi-supervised (SimCLRv2 fine-tuned) models of varied sizes on different label fractions. ResNets with depths of 50, 101, 152, width multiplier of 1×, 2× (w/o SK) are presented here. For supervised models on 1%/10% labels, AutoAugment [35] and label smoothing [36] are used. Increasing the size of SimCLRv2 models by 10×, from ResNet-50 to ResNet-152 (2×), improves label efficiency by 10×.

Figure 4 shows the performance as model size and label fraction vary. These results show that bigger models are more label-efficient for *both* supervised and semi-supervised learning, but gains appear to be larger for semi-supervised learning (more discussions in Appendix A). Furthermore, it is worth pointing out that although bigger models are better, some models (e.g. with SK) are more parameter efficient than others (Appendix B), suggesting that searching for better architectures is helpful.

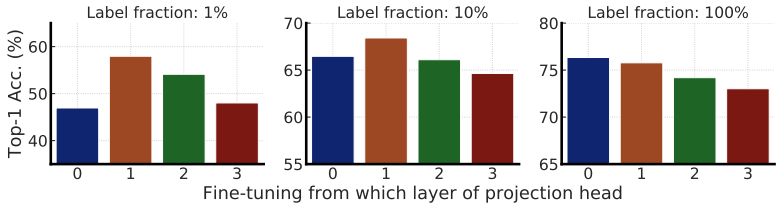
3.3 Bigger/Deeper Projection Heads Improve Representation Learning

This makes sense because the input layer is the latent space from the CNN: the learned general representation of the image. I don't see why it would be useful to fine-tune that part of the network.

To study the effects of projection head for fine-tuning, we pretrain ResNet-50 using SimCLRv2 with different numbers of projection head layers (from 2 to 4 fully connected layers), and examine performance when fine-tuning from different layers of the projection head. We find that using a deeper projection head during pretraining is better when fine-tuning from the optimal layer of projection head (Figure 5a), and this optimal layer is typically the first layer of projection head rather than the input (0^{th} layer), especially when fine-tuning on fewer labeled examples (Figure 5b).



(a) Effect of projection head's depth when fine-tuning from optimal middle layer.



(b) Effect of fine-tuning from middle of a 3-layer projection head (0 is SimCLR).

Figure 5: Top-1 accuracy via fine-tuning under different projection head settings and label fractions (using ResNet-50).

It is also worth noting that when using bigger ResNets, the improvements from having a deeper projection head are smaller (see Appendix E). In our experiments, wider ResNets also have wider projection heads, since the width multiplier is applied to both. Thus, it is possible that increasing the depth of the projection head has limited effect when the projection head is already relatively wide.

Table 2: Top-1 accuracy of a ResNet-50 trained on different types of targets. For distillation, the temperature is set to 1.0, and the teacher is ResNet-50 (2×+SK), which gets 70.6% with 1% of the labels and 77.0% with 10%, as shown in in Table 1. The distillation loss (Eq. 2) does not use label information. Neither strong augmentation nor extra regularization are used.

Method	Label fraction 1%	10%
Label only	12.3	52.0
Label + distillation loss (on labeled set)	23.6	66.2
Label + distillation loss (on labeled+unlabeled sets)	69.0	75.1
Distillation loss (on labeled+unlabeled sets; our default)	68.9	74.3

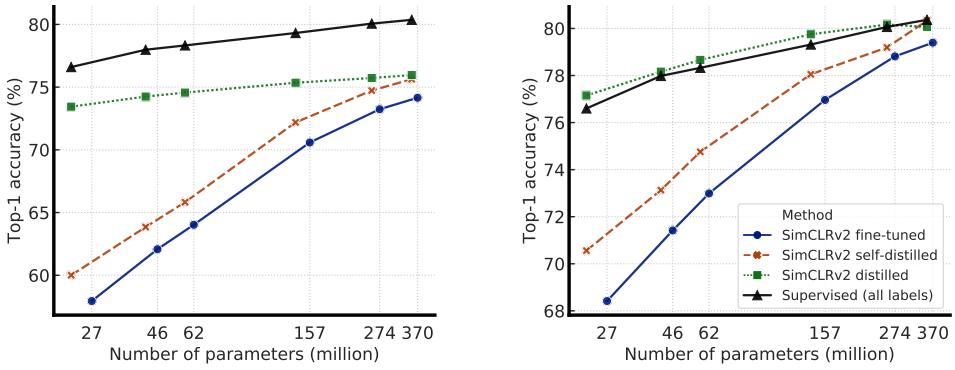
When varying architecture, the accuracy of fine-tuned models is correlated with the accuracy of linear evaluation (see Appendix C). Although we use the input of the projection head for linear classification, we find that the correlation is higher when fine-tuning from the optimal middle layer of the projection head than when fine-tuning from the projection head input.

3.4 Distillation Using Unlabeled Data Improves Semi-Supervised Learning

Distillation typically involves both a distillation loss that encourages the student to match a teacher and an ordinary supervised cross-entropy loss on the labels (Eq. 3). In Table 2, we demonstrate the importance of using unlabeled examples when training with the distillation loss. Furthermore, using the distillation loss alone (Eq. 2) works almost as well as balancing distillation and label losses (Eq. 3) when the labeled fraction is small. For simplicity, Eq. 2 is our default for all other experiments.

Reducing the size of the architecture will inherently improve model efficiency, but why does figure 6 convince me that it does so by "transferring task-specific knowledge".

Distillation with unlabeled examples improves fine-tuned models in two ways, as shown in Figure 6: (1) when the student model has a smaller architecture than the teacher model, it improves the model efficiency by transferring task-specific knowledge to a student model, (2) even when the student model has the same architecture as the teacher model (excluding the projection head after ResNet encoder), self-distillation can still meaningfully improve the semi-supervised learning performance. To obtain the best performance for smaller ResNets, the big model is self-distilled before distilling it to smaller models.



(a) Label fraction 1% (b) Label fraction 10%

Figure 6: Top-1 accuracy of distilled SimCLRv2 models compared to the fine-tuned models as well as supervised learning with all labels. The self-distilled student has the same ResNet as the teacher (without MLP projection head). The distilled student is trained using the self-distilled ResNet-152 (2×+SK) model, which is the largest model included in this figure.

We compare our best models with previous state-of-the-art semi-supervised learning methods (and a concurrent work [43]) on ImageNet in Table 3. Our approach greatly improves upon previous results, for both small and big ResNet variants.

Table 3: ImageNet accuracy of models trained under semi-supervised settings. For our methods, we report results with distillation after fine-tuning. For our smaller models, we use self-distilled ResNet-152 (3×+SK) as the teacher.

Method	Architecture	Top-1 Label fraction 1%	Top-1 Label fraction 10%	Top-5 Label fraction 1%	Top-5 Label fraction 10%
Supervised baseline [30]	ResNet-50	25.4	56.4	48.4	80.4
Methods using unlabeled data in a task-specific way:					
Pseudo-label [11, 30]	ResNet-50	-	-	51.6	82.4
VAT+Entropy Min. [37, 38, 30]	ResNet-50	-	-	47.0	83.4
Mean teacher [39]	ResNeXt-152	-	-	-	90.9
UDA (w. RandAug) [14]	ResNet-50	-	68.8	-	88.5
FixMatch (w. RandAug) [15]	ResNet-50	-	71.5	-	89.1
S4L (Rot+VAT+Entropy Min.) [30]	ResNet-50 (4×)	-	73.2	-	91.2
MPL (w. RandAug) [2]	ResNet-50	-	73.8	-	-
CowMix [40]	ResNet-152	-	73.9	-	91.2
Methods using unlabeled data in a task-agnostic way:					
InstDisc [17]	ResNet-50	-	-	39.2	77.4
BigBiGAN [41]	RevNet-50 (4×)	-	-	55.2	78.8
PIRL [42]	ResNet-50	-	-	57.2	83.8
CPC v2 [19]	ResNet-161(∗)	52.7	73.1	77.9	91.2
SimCLR [1]	ResNet-50	48.3	65.6	75.5	87.8
SimCLR [1]	ResNet-50 (2×)	58.5	71.7	83.0	91.2
SimCLR [1]	ResNet-50 (4×)	63.0	74.4	85.8	92.6
BYOL [43] (concurrent work)	ResNet-50	53.2	68.8	78.4	89.0
BYOL [43] (concurrent work)	ResNet-200 (2×)	71.2	77.7	89.5	93.7
Methods using unlabeled data in both ways:					
SimCLRv2 distilled (ours)	ResNet-50	73.9	77.5	91.5	93.4
SimCLRv2 distilled (ours)	ResNet-50 (2×+SK)	75.9	80.2	93.0	95.0
SimCLRv2 self-distilled (ours)	ResNet-152 (3×+SK)	**76.6**	**80.9**	**93.4**	**95.5**

4 Related work

Task-agnostic use of unlabeled data. Unsupervised or self-supervised pretraining followed by supervised fine-tuning on a few labeled examples has been extensively used in natural language processing [6, 5, 7–9], but has only shown promising results in computer vision very recently [19, 20, 1]. Our work builds upon recent success on contrastive learning of visual representations [44, 16, 17, 45, 18, 19, 46, 42, 20, 47, 48, 1], a sub-area within self-supervised learning. These contrastive learning based approaches learn representations in a discriminative fashion instead of a generative one as in [3, 49, 50, 41, 51]. There are other approaches to self-supervised learning that are based on handcrafted pretext tasks [52, 31, 53, 54, 27, 55]. We also note a concurrent work on advancing self-supervised pretraining without using negative examples [43], which we also compare against in Table 3. Our work also extends the "unsupervised pretrain, supervised fine-tune" paradigm by combining it with (self-)distillation [23, 22, 11] using unlabeled data.

Task-specific use of unlabeled data. Aside from the representation learning paradigm, there is a large and diverse set of approaches for semi-supervised learning, we refer readers to [56–58] for surveys of classical approaches. Here we only review methods closely related to ours (especially within computer vision). One family of highly relevant methods are based on pseudo-labeling [11, 15] or self-training [12, 24, 59]. The main differences between these methods and ours are that our initial / teacher model is trained using SimCLRv2 (with unsupervised pretraining and supervised fine-tuning), and the student models can also be smaller than the initial / teacher model. Furthermore, we use temperature scaling instead of confidence-based thresholding, and we do not use strong augmentation for training the student. Another family of methods are based on label consistency regularization [60–62, 39, 14, 13, 63, 15], where unlabeled examples are directly used as a regularizer to encourage task prediction consistency. Although in SimCLRv2 pretraining, we maximize the agreement/consistency of representations of the same image under different augmented views, there is *no* supervised label utilized to compute the loss, a crucial difference from label consistency losses.

I strongly recommend these survey papers to learn about semi-supervised learning

What does "strong" mean here? What is the intuition behind using "weak augmentation" for training the student?

5 Discussion

In this work, we present a simple framework for semi-supervised ImageNet classification in three steps: unsupervised pretraining, supervised fine-tuning, and distillation with unlabeled data. Although similar approaches are common in NLP, we demonstrate that this approach can also be a surprisingly strong baseline for semi-supervised learning in computer vision, outperforming the state-of-the-art by a large margin.

We observe that bigger models can produce larger improvements with fewer labeled examples. We primarily study this phenomenon on ImageNet, but we observe similar results on CIFAR-10, a much smaller dataset (see appendix G). The effectiveness of big models have been demonstrated on supervised learning [64–67], fine-tuning supervised models on a few examples [68], and unsupervised learning on language [9, 69, 10, 70]. However, it is still somewhat surprising that bigger models, which could easily overfit with few labeled examples, can generalize much better. With task-agnostic use of unlabeled data, we conjecture bigger models can learn more general features, which increases the chances of learning task-relevant features. However, further work is needed to gain a better understanding of this phenomenon. Beyond model size, we also see the importance of increasing parameter efficiency as the other important dimension of improvement.

I had this thought as well, and this conjecture at least makes sense. I also strongly agree further work is necessary to understand what's going on here.

Although big models are important for pretraining and fine-tuning, given a specific task, such as classifying images into 1000 ImageNet classes, we demonstrate that task-agnostically learned general representations can be distilled into a more specialized and compact network using unlabeled examples. We simply use the teacher to impute labels for the unlabeled examples for this purpose, without using noise, augmentation, confidence thresholding, or consistency regularization. When the student network has the same or similar architecture as the teacher, this process can consistently improve the classification performance. We believe our framework can benefit from better approaches to leverage the unlabeled data for improving and transferring task-specific knowledge. We also recognize that ImageNet is a well-curated dataset, and may not reflect all real-world applications of semi-supervised learning. Thus, a potential future direction is to explore wider range of real datasets.

6 Broader Impact

The findings described in this paper can potentially be harnessed to improve accuracy in any application of computer vision where it is more expensive or difficult to label additional data than to train larger models. Some such applications are clearly beneficial to society. For example, in medical applications where acquiring high-quality labels requires careful annotation by clinicians, better semi-supervised learning approaches can potentially help save lives. Applications of computer vision to agriculture can increase crop yields, which may help to improve the availability of food. However, we also recognize that our approach could become a component of harmful surveillance systems. Moreover, there is an entire industry built around human labeling services, and technology that reduces the need for these services could lead to a short-term loss of income for some of those currently employed or contracted to provide labels.

Acknowledgements

We would like to thank David Berthelot, Han Zhang, Lala Li, Xiaohua Zhai, Lucas Beyer, Alexander Kolesnikov for their helpful feedback on the draft. We are also grateful for general support from Google Research teams in Toronto and elsewhere.

References

[1] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*, 2020.

[2] Hieu Pham, Qizhe Xie, Zihang Dai, and Quoc V Le. Meta pseudo labels. *arXiv preprint arXiv:2003.10580*, 2020.

[3] Geoffrey E Hinton, Simon Osindero, and Yee-Whye Teh. A fast learning algorithm for deep belief nets. *Neural computation*, 18(7):1527–1554, 2006.

[4] Yoshua Bengio, Pascal Lamblin, Dan Popovici, and Hugo Larochelle. Greedy layer-wise training of deep networks. In *Advances in neural information processing systems*, pages 153–160, 2007.

[5] Andrew M Dai and Quoc V Le. Semi-supervised sequence learning. In *Advances in neural information processing systems*, pages 3079–3087, 2015.

[6] Ryan Kiros, Yukun Zhu, Russ R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Skip-thought vectors. In *Advances in neural information processing systems*, pages 3294–3302, 2015.

[7] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. 2018.

[8] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Proc. of NAACL*, 2018.

[9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[10] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners.

[11] Dong-Hyun Lee. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks.

[12] Qizhe Xie, Eduard Hovy, Minh-Thang Luong, and Quoc V Le. Self-training with noisy student improves imagenet classification. *arXiv preprint arXiv:1911.04252*, 2019.

[13] David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin A Raffel. Mixmatch: A holistic approach to semi-supervised learning. In *Advances in Neural Information Processing Systems*, pages 5050–5060, 2019.

[14] Qizhe Xie, Zihang Dai, Eduard Hovy, Minh-Thang Luong, and Quoc V Le. Unsupervised data augmentation. *arXiv preprint arXiv:1904.12848*, 2019.

[15] Kihyuk Sohn, David Berthelot, Chun-Liang Li, Zizhao Zhang, Nicholas Carlini, Ekin D Cubuk, Alex Kurakin, Han Zhang, and Colin Raffel. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *arXiv preprint arXiv:2001.07685*, 2020.

[16] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[17] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3733–3742, 2018.

[18] Philip Bachman, R Devon Hjelm, and William Buchwalter. Learning representations by maximizing mutual information across views. In *Advances in Neural Information Processing Systems*, pages 15509–15519, 2019.

[19] Olivier J Hénaff, Ali Razavi, Carl Doersch, SM Eslami, and Aaron van den Oord. Data-efficient image recognition with contrastive predictive coding. *arXiv preprint arXiv:1905.09272*, 2019.

[20] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. *arXiv preprint arXiv:1911.05722*, 2019.

[21] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

[22] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[23] Cristian Buciluǎ, Rich Caruana, and Alexandru Niculescu-Mizil. Model compression. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 535–541, 2006.

[24] I Zeki Yalniz, Hervé Jégou, Kan Chen, Manohar Paluri, and Dhruv Mahajan. Billion-scale semi-supervised learning for image classification. *arXiv preprint arXiv:1905.00546*, 2019.

[25] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[26] Suzanna Becker and Geoffrey E Hinton. Self-organizing neural network that discovers surfaces in random-dot stereograms. *Nature*, 355(6356):161–163, 1992.

[27] Alexander Kolesnikov, Xiaohua Zhai, and Lucas Beyer. Revisiting self-supervised visual representation learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1920–1929, 2019.

[28] Xiang Li, Wenhai Wang, Xiaolin Hu, and Jian Yang. Selective kernel networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 510–519, 2019.

[29] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020.

[30] Xiaohua Zhai, Avital Oliver, Alexander Kolesnikov, and Lucas Beyer. S4l: Self-supervised semi-supervised learning. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019.

[31] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *European conference on computer vision*, pages 649–666. Springer, 2016.

[32] Yang You, Igor Gitman, and Boris Ginsburg. Large batch training of convolutional networks. *arXiv preprint arXiv:1708.03888*, 2017.

[33] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[34] Tong He, Zhi Zhang, Hang Zhang, Zhongyue Zhang, Junyuan Xie, and Mu Li. Bag of tricks for image classification with convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 558–567, 2019.

[35] Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. Autoaugment: Learning augmentation strategies from data. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 113–123, 2019.

[36] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.

[37] Yves Grandvalet and Yoshua Bengio. Semi-supervised learning by entropy minimization. In *Advances in neural information processing systems*, pages 529–536, 2005.

[38] Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, and Shin Ishii. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE transactions on pattern analysis and machine intelligence*, 41(8):1979–1993, 2018.

[39] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *Advances in neural information processing systems*, pages 1195–1204, 2017.

[40] Geoff French, Avital Oliver, and Tim Salimans. Milking cowmask for semi-supervised image classification. *arXiv preprint arXiv:2003.12022*, 2020.

[41] Jeff Donahue and Karen Simonyan. Large scale adversarial representation learning. In *Advances in Neural Information Processing Systems*, pages 10541–10551, 2019.

[42] Ishan Misra and Laurens van der Maaten. Self-supervised learning of pretext-invariant representations. *arXiv preprint arXiv:1912.01991*, 2019.

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

[44] Alexey Dosovitskiy, Jost Tobias Springenberg, Martin Riedmiller, and Thomas Brox. Discriminative unsupervised feature learning with convolutional neural networks. In *Advances in neural information processing systems*, pages 766–774, 2014.

[45] R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. *arXiv preprint arXiv:1808.06670*, 2018.

[46] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. *arXiv preprint arXiv:1906.05849*, 2019.

[47] Junnan Li, Pan Zhou, Caiming Xiong, Richard Socher, and Steven CH Hoi. Prototypical contrastive learning of unsupervised representations. *arXiv preprint arXiv:2005.04966*, 2020.

[48] Yonglong Tian, Chen Sun, Ben Poole, Dilip Krishnan, Cordelia Schmid, and Phillip Isola. What makes for good views for contrastive learning. *arXiv preprint arXiv:2005.10243*, 2020.

[49] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[50] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[51] Ting Chen, Xiaohua Zhai, Marvin Ritter, Mario Lucic, and Neil Houlsby. Self-supervised gans via auxiliary rotation loss. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 12154–12163, 2019.

[52] Carl Doersch, Abhinav Gupta, and Alexei A Efros. Unsupervised visual representation learning by context prediction. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1422–1430, 2015.

[53] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *European Conference on Computer Vision*, pages 69–84. Springer, 2016.

[54] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. *arXiv preprint arXiv:1803.07728*, 2018.

[55] Priya Goyal, Dhruv Mahajan, Abhinav Gupta, and Ishan Misra. Scaling and benchmarking self-supervised visual representation learning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6391–6400, 2019.

[56] Olivier Chapelle, Bernhard Scholkopf, and Alexander Zien. Semi-supervised learning. *MIT Press*, 2006.

[57] Xiaojin Zhu and Andrew B Goldberg. Introduction to semi-supervised learning. *Synthesis lectures on artificial intelligence and machine learning*, 3(1):1–130, 2009.

[58] Avital Oliver, Augustus Odena, Colin A Raffel, Ekin Dogus Cubuk, and Ian Goodfellow. Realistic evaluation of deep semi-supervised learning algorithms. In *Advances in Neural Information Processing Systems*, pages 3235–3246, 2018.

[59] Yang Zou, Zhiding Yu, Xiaofeng Liu, BVK Kumar, and Jinsong Wang. Confidence regularized self-training. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5982–5991, 2019.

[60] Philip Bachman, Ouais Alsharif, and Doina Precup. Learning with pseudo-ensembles. In *Advances in neural information processing systems*, pages 3365–3373, 2014.

[61] Samuli Laine and Timo Aila. Temporal ensembling for semi-supervised learning. *arXiv preprint arXiv:1610.02242*, 2016.

[62] Mehdi Sajjadi, Mehran Javanmardi, and Tolga Tasdizen. Regularization with stochastic transformations and perturbations for deep semi-supervised learning. In *Advances in neural information processing systems*, pages 1163–1171, 2016.

[63] Vikas Verma, Alex Lamb, Juho Kannala, Yoshua Bengio, and David Lopez-Paz. Interpolation consistency training for semi-supervised learning. *arXiv preprint arXiv:1903.03825*, 2019.

[64] Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. Revisiting unreasonable effectiveness of data in deep learning era. In *Proceedings of the IEEE international conference on computer vision*, pages 843–852, 2017.

[65] Joel Hestness, Sharan Narang, Newsha Ardalani, Gregory Diamos, Heewoo Jun, Hassan Kianinejad, Md Patwary, Mostofa Ali, Yang Yang, and Yanqi Zhou. Deep learning scaling is predictable, empirically. *arXiv preprint arXiv:1712.00409*, 2017.

[66] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

[67] Dhruv Mahajan, Ross Girshick, Vignesh Ramanathan, Kaiming He, Manohar Paluri, Yixuan Li, Ashwin Bharambe, and Laurens van der Maaten. Exploring the limits of weakly supervised pretraining. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 181–196, 2018.

[68] Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Joan Puigcerver, Jessica Yung, Sylvain Gelly, and Neil Houlsby. Large scale learning of general visual representations for transfer. *arXiv preprint arXiv:1912.11370*, 2019.

[69] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*, 2019.

[70] Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.

[71] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

A When Do Bigger Models Help More?

Figure A.1 shows relative improvement by increasing the model size under different amount of labeled examples. Both supervised learning and semi-supervised learning (i.e. SimCLRv2) seem to benefit from having bigger models. The benefits are larger when (1) regularization techniques (such as augmentation, label smoothing) are used, or (2) the model is pretrained using unlabeled examples. It is also worth noting that these results may reflect a "ceiling effect": as the performance gets closer to the ceiling, the improvement becomes smaller.

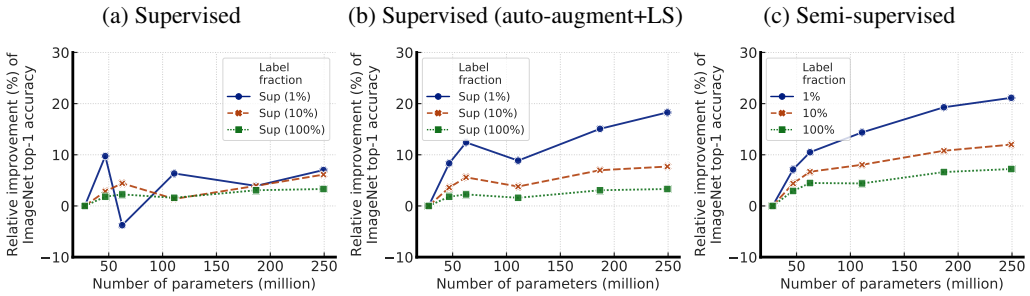
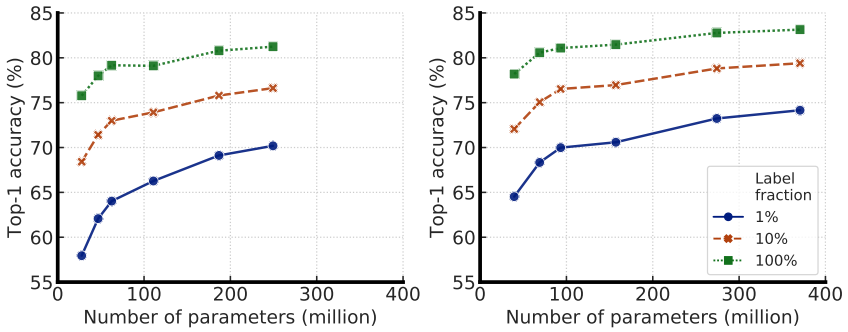


Figure A.1: Relative improvement (top-1) when model size is increased. (a) supervised learning without extra regularization, (b), Supervised learning with auto-augmentation [35] and label smoothing [71] are applied for 1%/10% label fractions, (c) semi-supervised learning by fine-tuning SimCLRv2.

B Parameter Efficiency Also Matters

Figure B.1 shows the top-1 accuracy of fine-tuned SimCLRv2 models of different sizes. It shows that (1) bigger models are better, but (2) with SK [28], better performance can be achieved with the same parameter count. It is worth to note that, in this work, we do not leverage group convolution for SK [28] and we use only 3×3 kernels. We expect further improvement in terms of parameter efficiency if group convolution is utilized.



(a) Models without SK [28] (b) Models with SK [28]

Figure B.1: Top-1 accuracy of fined-tuned SimCLRv2 models of different sizes on three label fractions. ResNets with depth in $\{50, 101, 152\}$, width in $\{1\times, 2\times\}$ are included here. Parameter efficiency also plays an important role. For fine-tuning on 1% of labels, SK is much more efficient.

C The Correlation Between Linear Evaluation and Fine-tuning

Most existing work [17, 19, 18, 42, 20, 1] on self-supervised learning leverages linear evaluation as a main metric for evaluating representation quality, and it is not clear how it correlates with semi-supervised learning through fine-tuning. Here we further study the correlation of fine-tuning and linear evaluation (the linear classifier is trained on the ResNet output instead of some middle layer of projection head). Figure C.1 shows the correlation under two different fine-tuning strategies:

fine-tuning from the input of projection head, or fine-tuning from a middle layer of projection head. We observe that overall there is a linear correlation. When fine-tuned from a middle layer of the projection head, we observe a even stronger linear correlation. Additionally, we notice the slope of correlation becomes smaller as number of labeled images for fine-tuning increases .

Observe that including projection head g in the fine-tuning process yields more stable predictions. Happy to see results like this given my expressed concerns of excluding g on page 4.

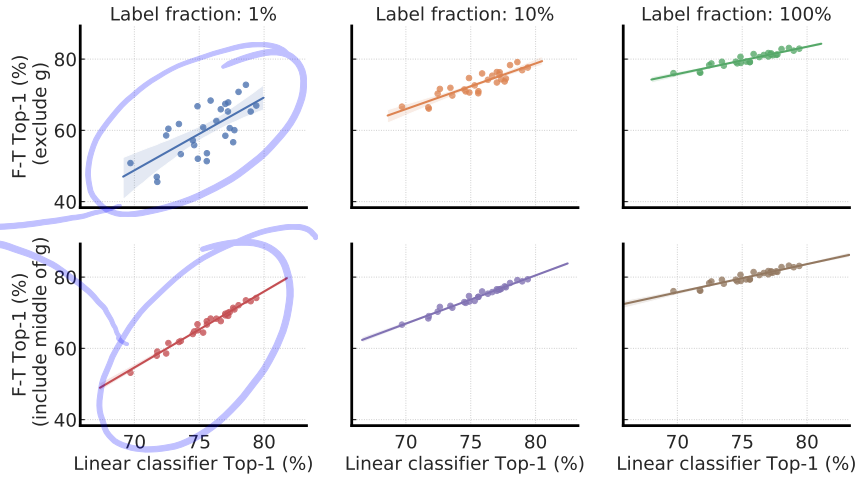


Figure C.1: The effects of projection head for the correlation between fine-tuning and linear evaluation. When allowing fine-tuning from the middle of the projection head, the linear correlation becomes stronger. Furthermore, as label fraction increases, the slope is decreasing. The points here are from the variants of ResNets with depth in $\{50, 101, 152\}$, width in $\{1\times, 2\times\}$, and with/without SK.

D The Impact of Memory

Figure D.1 shows the top-1 comparisons for SimCLRv2 models trained with or without memory (MoCo) [20]. Memory provides modest advantages in terms of linear evaluation and fine-tuning with 1% of the labels; the improvement is around 1%. We believe the reason that memory only provides marginal improvement is that we already use a big batch size (i.e. 4096).

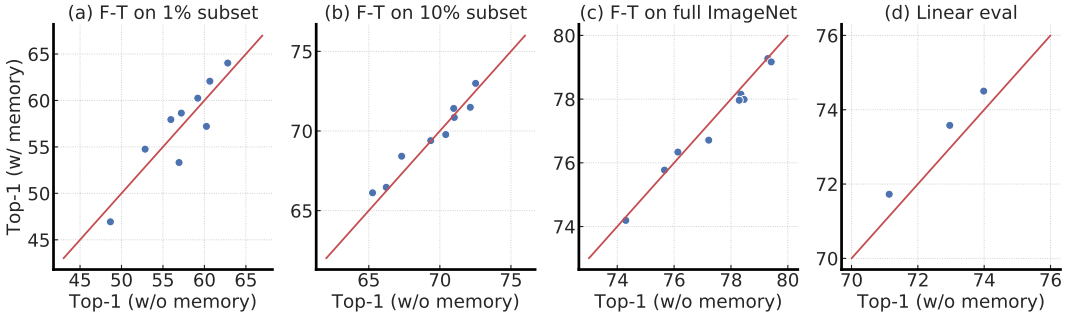
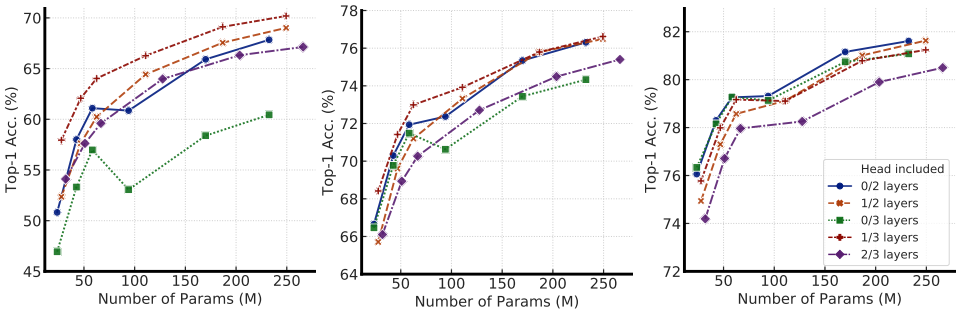


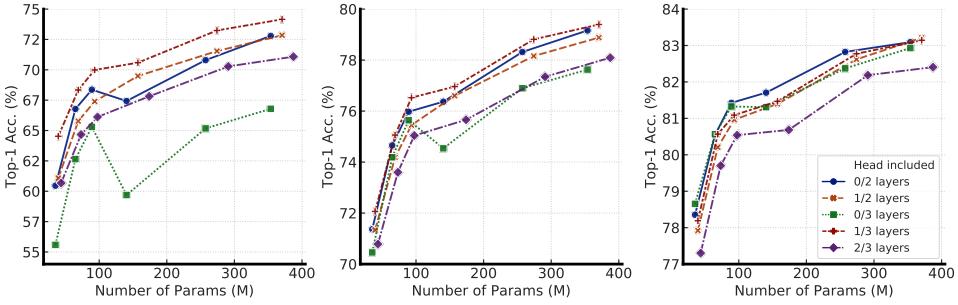
Figure D.1: Top-1 results of ResNet-50, ResNet-101, and ResNet-152 trained with or without memory.

E The Impact of Projection Head Under Different Model Sizes

To understand the effects of projection head settings across model sizes, Figure E.1 shows effects of fine-tuning from different layers of 2- and 3-layer projection heads. These results confirm that with only a few labeled examples, pretraining with a deeper projection head and fine-tuning from a middle layer can improve the semi-supervised learning performance. The improvement is larger with a smaller model size.



(a) w/o SK (label fraction: 1%) (b) w/o SK (label fraction: 10%) (c) w/o SK (label fraction: 100%)



(d) w/ SK (label fraction: 1%) (e) w/ SK (label fraction: 10%) (f) w/ SK (label fraction: 100%)

Figure E.1: Top-1 fine-tuning performance under different projection head settings (number of layers included for fine-tuning) and model sizes. With fewer labeled examples, fine-tuning from the first layer of a 3-layer projection head is better, especially when the model is small. Points reflect ResNets with depths of $\{50, 101, 152\}$ and width multipliers of $\{1\times, 2\times\}$. Networks in the first row are without SK, and in the second row are with SK.

Figure E.2 shows fine-tuning performance for different projection head settings of a ResNet-50 pretrained using SimCLRv2. Figure 5 in the main text is an aggregation of results from this Figure.

I would have liked to see plots like this for performance of original SimCLR with larger projection heads. The reason for this is that even though the projection head is thrown out for the task-specific applications in the original SimCLR paper, it still informs the encoder, and hence the latent space. I am curious if larger projection heads increases performance because more layers are fine-tuned. This would not have been the case with the original SimCLR paper, because the projection head is excluded from the fine-tuning process.

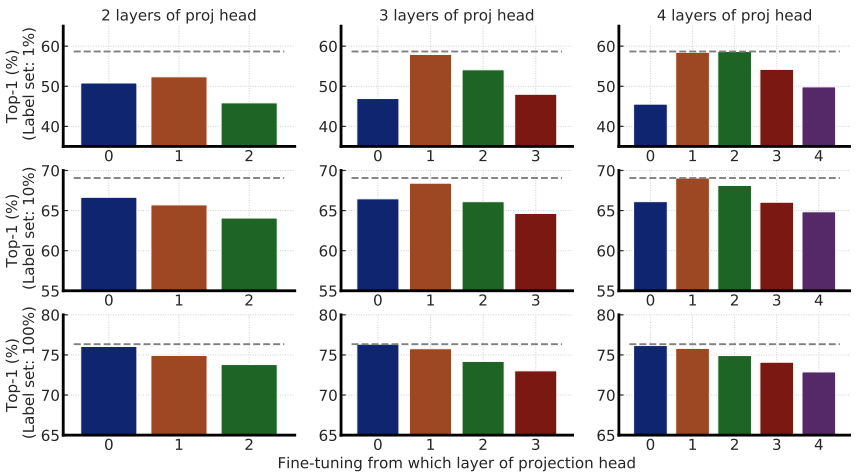
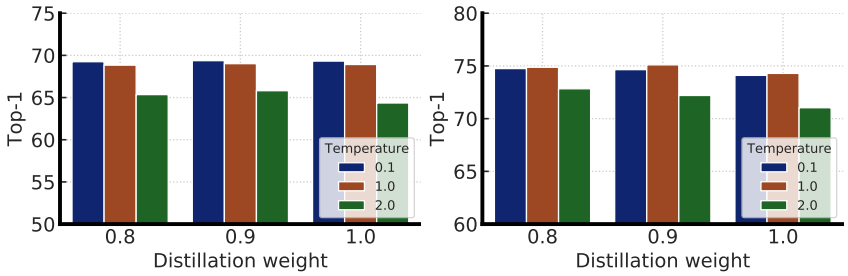


Figure E.2: Top-1 accuracy of ResNet-50 with different projection head settings. Deeper projection head help more, when allowing to fine-tune from a middle layer of the projection head.

F Further Distillation Ablations

Figure F.1 shows the impact of distillation weight (α) in Eq. 3, and temperature used for distillation. We see distillation without actual labels (i.e. distillation weight is 1.0) works on par with distillation

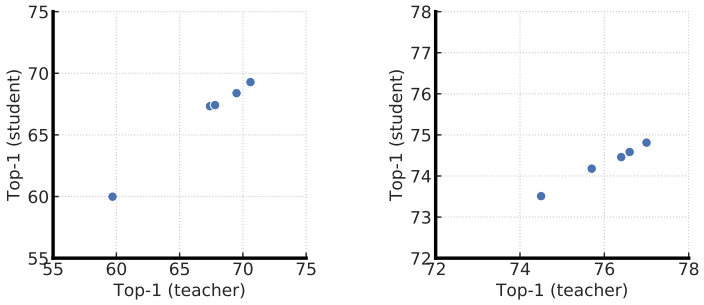
with actual labels. Furthermore, temperature of 0.1 and 1.0 work similarly, but 2.0 is significantly worse. For our distillation experiments in this work, we by default use a temperature of 0.1 when the teacher is a fine-tuned model, otherwise 1.0.



(a) label fraction: 1% (b) Label fraction: 10%

Figure F.1: Top-1 accuracy with different distillation weight (α), and temperature (τ).

We further study the distillation performance with teachers that are fine-tuned using different projection head settings. More specifically, we pretrain two ResNet-50 (2×+SK) models, with two or three layers of projection head, and fine-tune from a middle layer. This gives us five different teachers, corresponding to different projection head settings. Unsurprisingly, as shown in Figure F.2, distillation performance is strongly correlated with the top-1 accuracy of fine-tuned teacher. This suggests that a better fine-tuned model (measured by its top-1 accuracy), regardless their projection head settings, is a better teacher for transferring task specific knowledge to the student using unlabeled data.



(a) Label fraction: 1% (b) Label fraction: 10%

Figure F.2: The strong correlation between teacher's task performance and student's task performance.

G CIFAR-10

We perform main experiments on ImageNet since it is a large-scale and well studied dataset. Here we conduct similar experiments on small-scaled CIFAR-10 dataset to test our findings in ImageNet. More specifically, we pretrain ResNets on CIFAR-10 without labels following [1] [5]. The ResNet variants we trained are of 6 depths, namely 18, 34, 50, 101, 152 and 200. To keep experiments tractable, by default we use Selective Kernel, and a width multiplier of 1×. After the models are pretrained, we then fine-tune them (using simple augmentations of random crop and horizontal flipping) on different numbers of labeled examples (250, 4000, and total of 5000 labeled examples), following MixMatch's protocol and running on 5 seeds.

The fine-tuning performances are shown in the Figure G.1, and suggest similar trends to our results on ImageNet: big pretrained models can perform well, often better, with a few labeled examples. These

[5] For CIFAR-10, we do not find it beneficial to have a 3-layer projection head, nor using a EMA network. However, we expand augmentations in [1] with a broader set of color operators and Sobel filtering for pretraining. We pretrain it for 800 epochs, with a batch size of 512 and temperature of 0.2.

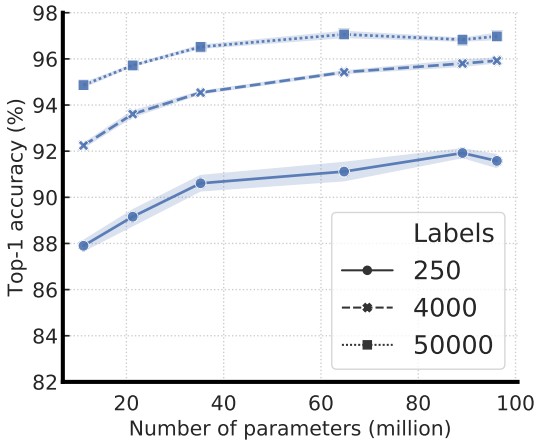


Figure G.1: Fine-tuning pre-trained ResNets on CIFAR-10.

results can be further improved with better augmentations during fine-tuning and an extra distillation step. The linear evaluation also improves with a bigger network. Our best result is 96.4% obtained from ResNet-101 (+SK) and ResNet-152 (+SK), but it is slightly worse (96.2%) with ResNet-200 (+SK).

H Extra Results

Table H.1 shows top-5 accuracy of the fine-tuned SimCLRv2 (under different model sizes) on ImageNet.

Table H.1: Top-5 accuracy of fine-tuning SimCLRv2 (on varied label fractions) or training a linear classifier on the ResNet output. The supervised baselines are trained from scratch using all labels in 90 epochs. The parameter count only include ResNet up to final average pooling layer. For fine-tuning results with 1% and 10% labeled examples, the models include additional non-linear projection layers, which incurs additional parameter count (4M for 1× models, and 17M for 2× models).

Depth	Width	Use SK [28]	Param (M)	Fine-tuned on 1%	Fine-tuned on 10%	Fine-tuned on 100%	Linear eval	Supervised
50	1×	False	**24**	**82.5**	**89.2**	**93.3**	**90.4**	**93.3**
		True	35	86.7	91.4	94.6	92.3	94.2
	2×	False	94	87.4	91.9	94.8	92.7	93.9
		True	140	90.2	93.7	95.9	93.9	94.5
101	1×	False	43	85.2	90.9	94.3	91.7	93.9
		True	65	89.2	93.0	95.4	93.1	94.8
	2×	False	170	88.9	93.2	95.6	93.4	94.4
		True	257	91.6	94.5	96.4	94.5	95.0
152	1×	False	58	86.6	91.8	94.9	92.4	94.2
		True	89	90.0	93.7	95.9	93.6	95.0
	2×	False	233	89.4	93.5	95.8	93.6	94.5
		True	354	92.1	94.7	96.5	94.7	95.0
152	3×	True	**795**	**92.3**	**95.0**	**96.6**	**94.9**	**95.1**

The fact that fine-tune on 10% performs comparably to supervised (on 100%) is less impressive to me because fine-tuning on 100% yields a 1.5% improvement from supervision. Don't get me wrong, this is still a great paper with great contributions, but this suggests the real improvement is in the representations from the original SimCLR and less from this new fine-tuning + distillation mechanism.