S^4L: Self-Supervised Semi-Supervised Learning

Xiaohua Zhai*, Avital Oliver*, Alexander Kolesnikov*, Lucas Beyer*
Google Research, Brain Team
{xzhai, avitalo, akolesnikov, lbeyer}@google.com

Abstract

This work tackles the problem of semi-supervised learning of image classifiers. Our main insight is that the field of semi-supervised learning can benefit from the quickly advancing field of self-supervised visual representation learning. Unifying these two approaches, we propose the framework of self-supervised semi-supervised learning (S^4L) and use it to derive two novel semi-supervised image classification methods. We demonstrate the effectiveness of these methods in comparison to both carefully tuned baselines, and existing semi-supervised learning methods. We then show that S^4L and existing semi-supervised methods can be jointly trained, yielding a new state-of-the-art result on semi-supervised ILSVRC-2012 with 10% of labels.

1. Introduction

Modern computer vision systems demonstrate outstanding performance on a variety of challenging computer vision benchmarks, such as image recognition [34], object detection [22], semantic image segmentation [8], *etc*. Their success relies on the availability of a large amount of annotated data that is time-consuming and expensive to acquire. Moreover, applicability of such systems is typically limited in scope defined by the dataset they were trained on.

Many real-world computer vision applications are concerned with visual categories that are not present in standard benchmark datasets, or with applications of dynamic nature where visual categories or their appearance may change over time. Unfortunately, building large labeled datasets for all these scenarios is not practically feasible. Therefore, it is an important research challenge to design a learning approach that can successfully learn to recognize new concepts by leveraging only a small amount of labeled examples. The fact that humans quickly understand new concepts after seeing only a few (labeled) examples suggests that this goal is achievable in principle.

Notably, a large research effort is dedicated towards learning from unlabeled data that, in many realistic applications, is much less onerous to acquire than labeled data. Within this effort, the field of self-supervised visual representation learning has recently demonstrated the most promising results [17]. Self-supervised learning techniques define *pretext* tasks which can be formulated using only unlabeled data, but do require higher-level semantic understanding in order to be solved. As a result, models trained for solving these pretext tasks learn representations that can be used for solving other downstream tasks of interest, such as image recognition.

Despite demonstrating encouraging results [17], purely self-supervised techniques learn visual representations that are significantly inferior to those delivered by fully-supervised techniques. Thus, their practical applicability is

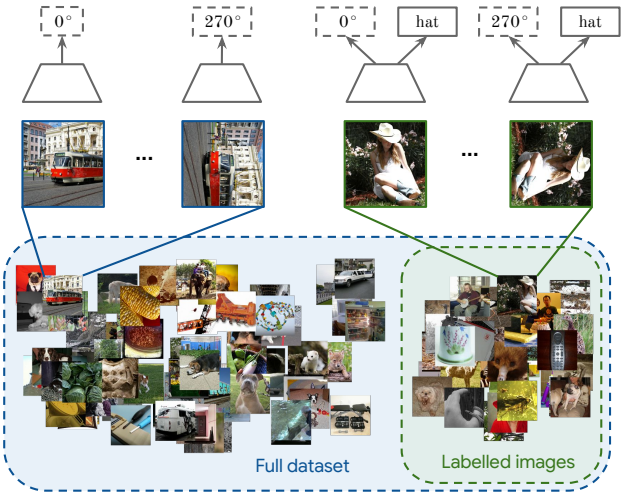

Figure 1. A schematic illustration of one of the proposed self-supervised semi-supervised techniques: S^4L-Rotation. Our model makes use of both labeled and unlabled images. The first step is to create four input images for any image by rotating it by 0°, 90°, 180° and 270° (inspired by [10]). Then, we train a single network that predicts which rotation was applied to all these images and, additionally, predicts semantic labels of annotated images. This conceptually simple technique is competitive with existing semi-supervised learning methods.

*equal contribution

limited and as of yet, self-supervision alone is insufficient.

We hypothesize that self-supervised learning techniques could dramatically benefit from a small amount of labeled examples. By investigating various ways of doing so, we bridge self-supervised and semi-supervised learning, and propose a framework of semi-supervised losses arising from self-supervised learning targets. We call this framework self-supervised semi-supervised learning or, in short, S^4L. The techniques derived in that way can be seen as new semi-supervised learning techniques for natural images. Figure 1 illustrates the idea of the proposed S^4L techniques. We thus evaluate our models both in the semi-supervised setup, as well as in the transfer setup commonly used to evaluate self-supervised representations. Moreover, we design strong baselines for benchmarking methods which learn using only $10\,\%$ or $1\,\%$ of the labels in ILSVRC-2012.

We further experimentally investigate whether our S^4L methods could further benefit from regularizations proposed by the semi-supervised literature, and discover that they are complementary, *i.e*. combining them leads to improved results.

Our main contributions can be summarized as follows:

- We propose a new family of techniques for semi-supervised learning with natural images that leverage recent advances in self-supervised representation learning.
- We demonstrate that the proposed self-supervised semi-supervised (S^4L) techniques outperform carefully tuned baselines that are trained with no unlabeled data, and achieve performance competitive with previously proposed semi-supervised learning techniques.
- We further demonstrate that by combining our best S^4L methods with existing semi-supervised techniques, we achieve new state-of-the-art performance on the semi-supervised ILSVRC-2012 benchmark.

2. Related Work

In this work we build on top of the current state-of-the-art in both fields of semi-supervised and self-supervised learning. Therefore, in this section we review the most relevant developments in these fields.

2.1. Semi-supervised Learning

Semi-supervised learning describes a class of algorithms that seek to learn from both unlabeled and labeled samples, typically assumed to be sampled from the same or similar distributions. Approaches differ on what information to gain from the structure of the unlabeled data.

Given the wide variety of semi-supervised learning techniques proposed in the literature, we refer to [4] for an extensive survey. For more context, we focus on recent developments based on deep neural networks.

The standard protocol for evaluating semi-supervised learning algorithms works as such: (1) Start with a standard labeled dataset; (2) Keep only a portion of the labels (say, $10\,\%$) on that dataset; (3) Treat the rest as unlabeled data. While this approach may not reflect realistic settings for semi-supervised learning [30], it remains the standard evaluation protocol, which we follow it in this work.

Many of the initial results on semi-supervised learning with deep neural networks were based on generative models such as denoising autoencoders [33], variational autoencoders [16] and generative adversarial networks [29, 35]. More recently, a line of research showed improved results on standard baselines by adding *consistency regularization losses* computed on unlabeled data. These consistency regularization losses measure discrepancy between predictions made on perturbed unlabeled data points. Additional improvements have been shown by smoothing predictions before measuring these perturbations. Approaches of these kind include Π-Model [19], Temporal Ensembling [19], Mean Teacher [41] and Virtual Adversarial Training [24]. Recently, fast-SWA[1] showed improved results by training with cyclic learning rates and measuring discrepancy with an ensemble of predictions from multiple checkpoints. By minimizing consistency losses, these models implicitly push the decision boundary away from high-density parts of the unlabeled data. This may explain their success on typical image classification datasets, where points in each clusters typically share the same class.

Two additional important approaches for semi-supervised learning, which have shown success both in the context of deep neural networks and other types of models are Pseudo-Labeling [20], where one imputes approximate classes on unlabeled data by making predictions from a model trained only on labeled data, and conditional entropy minimization [11], where all unlabeled examples are encouraged to make confident predictions on *some* class.

Semi-supervised learning algorithms are typically [30, 24, 2, 42, 1, 23] evaluated on small-scale datasets such as CIFAR-10 [18] and SVHN [25]. We are aware of very few examples in the literature where semi-supervised learning algorithms are evaluated on larger, more challenging datasets such as ILSVRC-2012 [34]. To our knowledge, Mean Teacher [41] currently holds the state-of-the-art result on ILSVRC-2012 when using only $10\,\%$ of the labels. Recent concurrent work [43, 13] presents competitive results on ILSVRC-2012.

2.2. Self-supervised Learning

Self-supervised learning is a general learning framework that relies on surrogate (pretext) tasks that can be formulated using only unsupervised data. A pretext task is de-

signed in a way that solving it requires learning of a useful image representation. Self-supervised techniques have a variety of applications in a broad range of computer vision topics [15, 37, 7, 31, 36].

In this paper we employ self-supervised learning techniques that are designed to learn useful visual representations from image databases. These techniques achieve state-of-the-art performance among approaches that learn visual representations from unsupervised images only. Below we provided a non-comprehensive summary of the most important developments in this direction.

Doersch et al. propose to train a CNN model that predicts relative location of two randomly sampled non-overlapping image patches [5]. Follow-up papers [26, 28] generalize this idea for predicting a permutation of multiple randomly sampled and permuted patches.

Beside the above patch-based methods, there are self-supervised techniques that employ image-level losses. Among those, in [44] the authors propose to use grayscale image colorization as a pretext task. Another example is a pretext task [10] that predicts an angle of the rotation transformation that was applied to an input image.

Some techniques go beyond solving surrogate classification tasks and enforce constraints on the representation space. A prominent example is the *exemplar* loss from [6] that encourages the model to learn a representation that is invariant to heavy image augmentations. Another example is [27], that enforces additivity constraint on visual representation: the sum of representations of all image patches should be close to representation of the whole image. Finally, [3] proposes a learning procedure that alternates between clustering images in the representation space and learning a model that assigns images to their clusters.

3. Methods

In this section we present our self-supervised semi-supervised learning (S^4L) techniques. We first provide a general description of our approach. Afterwards, we introduce specific instantiations of our approach.

We focus on the semi-supervised image classification problem. Formally, we assume an (unknown) data generating joint distribution $p(X, Y)$ over images and labels. The learning algorithm has access to a labeled training set D_l, which is sampled i.i.d. from $p(X, Y)$ and an unlabeled training set D_u, which is sampled i.i.d. from the marginal distribution $p(X)$.

The semi-supervised methods we consider in this paper have a learning objective of the following form:

$$\min_{\theta} \; \mathcal{L}_l(D_l, \theta) + w\mathcal{L}_u(D_u, \theta), \tag{1}$$

where $\mathcal{L}_l$ is a standard cross-entropy classification loss of all labeled images in the dataset, $\mathcal{L}_\text{u}$ is a loss defined on unsupervised images (we discuss its particular instances later in this section), w is a non-negative scalar weight and θ is the parameters for model $f_\theta(\cdot)$. Note that the learning objective can be extended to multiple unsupervised losses.

3.1. Self-supervised Semi-supervised Learning

We now describe our self-supervised semi-supervised learning techniques. For simplicity, we present our approach in the context of multiclass image recognition, even though it can be easily generalized to other scenarios, such as dense image segmentation.

It is important to note that in practice the objective 1 is optimized using a stochastic gradient descent (or a variant) that uses mini-batches of data to update the parameters θ. In this case the size of a supervised mini-batch $x_l, y_l \subset D_l$ and an unsupervised mini-batch $x_u \subset D_u$ can be arbitrary chosen. In our experiments we always default to simplest possible option of using minibatches of equal sizes.

We also note that we can choose whether to include the minibatch x_l into the self-supervised loss, i.e. apply $\mathcal{L}_\text{self}$ to the union of x_u and x_l. We experimentally study the effect of this choice in our experiments Section 4.4.

We demonstrate our framework on two prominent self-supervised techniques: predicting image rotation [10] and exemplar [6]. Note, that with our framework, more self-supervised losses can be explored in the future.

S^4L-Rotation. The key idea of rotation self-supervision is to rotate an input image then predict which rotation degree was applied to these rotated images. The loss is defined as:

$$\mathcal{L}_{rot} = \frac{1}{|\mathcal{R}|} \sum_{r \in \mathcal{R}} \sum_{x \in D_u} \mathcal{L}(f_\theta(x^r), r) \tag{2}$$

where $\mathcal{R}$ is the set of the 4 rotation degrees $\{0°, 90°, 180°, 270°\}$, x^r is the image x rotated by r, $f_\theta(\cdot)$ is the model with parameters θ,$\mathcal{L}$ is the cross-entropy loss. This results in a 4-class classification problem. We follow a recommendation from [10] and in a single optimization step we always apply and predict all four rotations for every image in a minibatch.

We also apply the self-supervised loss to the labeled images in each minibatch. Since we process rotated supervised images in this case, we suggest to also apply a classification loss to these images. This can be seen as an additional way to regularize a model in a regime when a small amount of labeled image are available. We measure the effect of this choice later in Section 4.4.

S^4L-Exemplar. The idea of exemplar self-supervision [6] is to learn a visual representation that is invariant to a wide range of image transformations. Specifically, we use "Inception" cropping [40], random horizontal mirroring, and HSV-space color randomization as in [6] to produce 8 different instances of each image in a minibatch. Following [17], we implement $\mathcal{L}_u$ as the batch hard triplet loss [14]

with a soft margin. This encourages transformation of the same image to have similar representations and, conversely, encourages transformations of different images to have diverse representations.

Similarly to the rotation self-supervision case, $\mathcal{L}_u$ is applied to all eight instances of each image.

3.2. Semi-supervised Baselines

In the following section, we compare S^4L to several leading semi-supervised learning algorithms that are not based on self-supervised objectives. We now describe the approaches that we compare to.

Our proposed objective 1 is applicable for semi supervised learning methods as well, where the loss $\mathcal{L}_u$ is the standard semi supervised loss as described below.

Virtual Adversarial Training (VAT) [24]: The idea is making the predicted labels robust around input data point against local perturbation. It approximates the maximal change in predictions within an ϵ_{vat} vicinity of unlabeled data points, where ϵ_{vat} is a hyperparameter. Concretely, the VAT loss for a model f_θ is:

$$\mathcal{L}_{\text{vat}} = \frac{1}{|\mathcal{D}_u|} \sum_{x \in \mathcal{D}_u} \text{KL}(f_\theta(x) \parallel f_\theta(x + \Delta x)), \quad (3)$$

where

$$\Delta x = \underset{\delta \text{ s.t. } |\delta|_2 = \epsilon}{\arg\max} \text{KL}(f_\theta(x) \parallel f_\theta(x + \delta)). \quad (4)$$

While computing Δx directly is not tractable, it can be efficiently approximated at the cost of an extra forward and backwards pass for every optimization step. [24].

Conditional Entropy Minimization (EntMin) [11]: This works under the assumption that unlabeled data indeed has one of the classes that we are training on, even when the particular class is not known during training. It adds a loss for unlabeled data that, when minimized, encourages the model to make confident predictions on unlabeled data. Specifically, the conditional entropy minimization loss for a model f_θ (treating f_θ as a conditional distribution of labels over images) is:

$$\mathcal{L}_{\text{entmin}} = \frac{1}{|\mathcal{D}_u|} \sum_{x \in \mathcal{D}_u} \sum_{y \in Y} -f_\theta(y|x) \log f_\theta(y|x) \quad (5)$$

Alone, the EntMin loss is not useful in the context of deep neural networks because the model can easily become extremely confident by increasing the weights of the last layer. One way to resolve this is to encourage the model predictions to be locally-Lipschitz, which VAT does[38]. Therefore, we only consider VAT and EntMin combined, not just EntMin alone, *i.e.* $\mathcal{L}_u = w_{vat}\mathcal{L}_{\text{vat}} + w_{\text{entmin}}\mathcal{L}_{\text{entmin}}$.

Pseudo-Label [20] is a simple approach: Train a model only on labeled data, then make predictions on unlabeled data. Then enlarge your training set with the predicted classes of the unlabeled data points whose predictions are confident past some threshold of confidence. Re-train your model with this enlarged labeled dataset. While [30] shows that in a simple "two moons" dataset, psuedo-label fails to learn a good model, in many real datasets this approach does show meaningful gains.

4. ILSVRC-2012 Experiments and Results

In this section, we present the results of our main experiments. We used the ILSVRC-2012 dataset due to its widespread use in self-supervised learning methods, and to see how well semi-supervised methods scale.

Since the test set of ILSVRC-2012 is not available, and numbers from the validation set are usually reported in the literature, we performed all hyperparameter selection for all models that we trained on a custom train/validation split of the public training set. This custom split contains 1 231 121 training and 50 046 validation images. We then retrain the model using the best hyperparameters on the full training set (1 281 167 images), possibly with fewer labels, and report final results obtained on the public validation set (50 000 images).

We follow standard practice [41, 32] and perform experiments where class-balanced labels are available for only $10\,\%$ of the dataset. Note that $10\,\%$ of ILSVRC-2012 still corresponds to roughly 128 000 labeled images, and that previous work uses the full (public) validation set for model selection. While we use a custom validation set extracted from the training set, using such a large validation set does not correspond to a realistic scenario, as already discussed by [33, 41, 30]. We also want to cover more realistic cases in our evaluation. We thus perform experiments on $1\,\%$ of labeled examples (roughly 13 000 labeled images), while also using a validation set of only 5000 images. We analyze the impact of validation set size in Section 7.

We always define epochs in terms of the available labeled data, *i.e.* one epoch corresponds to one full pass through the labeled data, regardless of how many unlabeled examples have been seen. We optimize our models using stochastic gradient descent with momentum on minibatches of size 256 unless specified otherwise. While we do tune the learning rate, we keep the momentum fixed at 0.9 across all experiments. Table 1 summarizes our main results.

4.1. Plain Supervised Learning

Whenever new methods are introduced, it is crucial to compare them against a solid baseline of existing methods. The simplest baseline to which any semi-supervised learning method should be compared to, is training a plain supervised model on the available labeled data.

Oliver *et al*. [30] discovered that reported baselines trained on labeled examples alone are unfairly weak, perhaps given that there is not a strong community behind tuning those baselines. They provide strong supervised-only baselines for SVHN and CIFAR-10, and show that the gap shown by the use of unlabeled data is smaller than reported.

We observed the same in the case of ILSVRC-2012. Thus, we aim to provide a strong baseline for future research by performing a relatively large search over training hyperparameters for training a model on only $10\,\%$ of ILSVRC-2012. Specifically, we try weight-decay values in $\{1, 3\} \cdot 10^{\{-2,-3,-4\}}$, learning rates in $\{0.3, 0.1, 0.03\}$, four different learning rate schedules spanning 30 to 500 epochs, and finally we explore various model architectures: ResNet50, ResNet34, ResNet18, in both "regular" (v1) and "pre-activation" (v2) variants, as well as half-, double-, triple-, and quadruple-width variants of these, testing the assumption that smaller or shallower models overfit less.

In total, we trained several thousand models on our custom training/validation split of the public training set of ILSVRC-2012. In summary, it is crucial to tune both weight decay and training duration while, perhaps surprisingly, model architecture, depth, and width only have a small influence on the final results. We thus use a standard, unmodified ResNet50v2 as model, trained with weight decay of 10^{-3} for 200 epochs, using a standard learning rate of 0.1, ramped up from 0 for the first five epochs, and decayed by a factor of 10 at epochs 140, 160, and 180. We train in total for 200 epochs. The standard augmentation procedure of random cropping and horizontal flipping is used during training, and predictions are made using a single central crop keeping aspect ratio.

We perform a similar search when training our baseline on $1\,\%$ of ILSVRC-2012, but additionally include two choices of data augmentation (whether or not to apply random color augmentation) and two minibatch sizes (256 and 1024) in the hyperparameter search. Perhaps somewhat surprisingly, the results here are similar, in that tuning the weight decay and training duration is crucial, but model architecture does not matter much. Additionally, performing color augmentation becomes important. Here too, we use a standard, unmodified ResNet50v2 as model, trained with weight decay of 10^{-2} for 1000 epochs, using a learning rate of 0.01[1], ramped up from 0.0 for the first ten epochs[2], and decayed by a factor of 10 at epochs 700, 800, and 900. We train in total for 1000 epochs. A more detailed presentation of the results is provided in the supplementary material.

Using this only slightly altered training procedure, our baseline models achieve $80.43\,\%$ top5 accuracy ($56.35\,\%$ top1) on the public ILSVRC-2012 validation set when trained on only $10\,\%$ of the full training set. Our $1\,\%$ baseline achieves $48.43\,\%$ top5 accuracy ($25.39\,\%$ top1). These results form a solid baseline to compare to, considering that the same standard ResNet50v2 model achieves $92.82\,\%$ top5 accuracy ($75.89\,\%$ top1) on $100\,\%$ of the labels.

[1]While the standard learning rate of 0.1 worked equally well, learning curves seemed significantly less stable.

[2]This was likely not necessary, but kept for consistency.

Table 1. Top-5 accuracy [%] obtained by individual methods when training them on ILSVRC-2012 with a subset of labels. All methods use the same standard width ResNet50v2 architecture.

ILSVRC-2012 labels: (i.e. images per class)	$10\,\%$ (128)	$1\,\%$ (13)
Supervised Baseline (Section 4.1)	80.43	48.43
Pseudolabels [20]	82.41	51.56
VAT [24]	82.78	44.05
VAT + Entropy Minimization [11]	83.39	46.96
Self-sup. Rotation [17] + Linear	39.75	25.98
Self-sup. Exemplar [17] + Linear	32.32	21.33
Self-sup. Rotation [17] + Fine-tune	78.53	45.11
Self-sup. Exemplar [17] + Fine-tune	81.01	44.90
S^4L-Rotation	83.82	53.37
S^4L-Exemplar	83.72	47.02

For all further experiments, we reuse the best hyperparameters discovered here, except that we try two additional learning rates: $\{0.3, 0.1, 0.03\}$ for $10\,\%$ and $\{0.03, 0.01, 0.003\}$ for $1\,\%$, and two additional weight decays: $\{10^{-4}, 3{\cdot}10^{-4}, 10^{-3}\}$ for $10\,\%$ and $\{3{\cdot}10^{-3}, 10^{-2}, 3{\cdot}10^{-2}\}$ for $1\,\%$. We also try two different weights w_u for the additionally introduced loss $\mathcal{L}_u$: $w_u \in \{0.5, 1.0\}$.

4.2. Semi-supervised Baselines

We train semi-supervised baseline models using (1) Pseudo-Label, (2) VAT, and (3) VAT+EntMin. To the best of our knowledge, we present the first evaluation of these techniques on ILSVRC-2012.

Pseudo-Label Using the plain supervised learning models from Section 4.1, we assign pseudo-labels to the full dataset. Then, in a second step, we train a ResNet50v2 from scratch following standard practice, *i.e*. with learning rate 0.1, weight decay 10^{-4}, and 100 epochs on the full (pseudo-labeled) dataset.

We try both using all predictions as pseudo-labels, as well as using only those predictions with a confidence above 0.5. Both perform closely on our validation set, and we choose no filtering for the final model for simplicity.

Table 1 shows that a second step training with pseudo-labels consistently improves the results on both $10\,\%$ and the $1\,\%$ labels case. This motivates us to apply the idea to our best semi supervised model, which is discussed in Section 5.

VAT We first verify our VAT implementation on CIFAR-10. With 4000 labels, we are able to achieve $86.41\,\%$ top-1 accuracy, which is in line with the $86.13\,\%$ reported in [30].

Besides the previously mentioned hyperparameters common to all methods, VAT needs tuning ϵ_{vat}. Since it corresponds to a distance in pixel space, we use a simple heuristic for defining a range of values to try for ϵ_{vat}: values should be lower than half the distance between neighbouring images in the dataset. Based on this heuristic, we try values of $\epsilon_{\text{vat}} \in \{50, 50 \cdot 2^{-1/3}, 50 \cdot 2^{-2/3}, 25\}$ and found $\epsilon_{\text{vat}} \approx 40$ to work best.

VAT+EntMin VAT is intended to be used together with an additional entropy minimization (EntMin) loss. EntMin adds a single hyperparameter to our best VAT model: the weight of the EntMin loss, for which we try $w_{\text{entmin}} \in \{0, 0.03, 0.1, 0.3, 1\}$, without re-tuning ϵ_{vat}.

The results of our best VAT and VAT+EntMin model are shown in Table 1. As can be seen, VAT performs well in the $10\,\%$ case, and adding adding entropy minimization consistently improves its performance. In Section 5, we further extend the co-training idea to include the self-supervised rotation loss.

4.3. Self-supervised Baselines

Previous work has evaluated features learned via self-supervision on the unlabeled data in a "semi-supervised" way by either freezing the features and learning a linear classifier on top, or by using the self-supervised model as an initialization and fine-tuning, using a subset of the labels in both cases. In order to compare our proposed way to do self-supervised semi-supervised learning to these common evaluations, we train a rotation and an exemplar model following the best practice from [17] but with standard width ("$4\times$" in [17]).

Following our established protocol, we tune the weight decay and learning rate for the logistic regression, although interestingly the standard values from [12] of 10^{-4} weight decay and 0.1 learning rate worked best.

The results of evaluating these models with both $10\,\%$ and $1\,\%$ are presented in Table 1 as "Self-sup. + Linear" and "Self-sup. + Fine-tune". Note that while our results for the linear experiment are similar to those reported in [17], they are not directly. This is due to 1) ours being evaluated on the public validation set, while they evaluated on a custom validation set, and 2) they used L-BFGS while we use SGD with standard augmentations. Furthermore, fine-tuning approaches or slightly surpasses the supervised baseline.

4.4. Self-supervised Semi-supervised Learning (S^4L)

For training our full self-supervised semi-supervised models (S^4L), we follow the same protocol as for our semi-supervised baselines, *i.e.* we use the best settings of the plain supervised baseline and only tune the learning rate, weight decay, and weight of the newly introduced loss. We found that for both S^4L-Rotation and S^4L-Exemplar, the self-supervised loss weight $w = 1$ worked best (though not by much) and the optimal weight decay and learning rate were the same as for the supervised baseline.

As described in Section 3.1, we apply the self-supervised loss on both labeled and unlabeled images. Furthermore, both Rotation and Exemplar self-supervision generate augmented copies of each image, and we do apply the supervision loss on all copies of the labeled images. We performed one case study on S^4L-Rotation in order to investigate this choice, and found that whether or not the self-supervision loss $\mathcal{L}_{\text{self}}$ is also applied on the labeled images does not have significant impact. On the other hand, applying the supervision loss $\mathcal{L}_{\text{sup}}$ on the augmented images generated by self-supervision does indeed improve performance by almost $1\,\%$. Furthermore, this allows to use multiple transformed copies of an image at inference-time (*e.g.* four rotations) and take the average of their predictions. While this 4-rot prediction is $1\,\%$ to $2\,\%$ more accurate, the results we report do *not* make use of this in order to keep comparison fair.

The results shown in Table 1 show that our proposed way of doing self-supervised semi-supervised learning is indeed effective for the two self-supervision methods we tried. We hypothesize that such approaches can be designed for other self-supervision objectives.

We additionally verified that our proposed method is not sensitive to the random seed, nor the split of the dataset, see Appendix B for details.

Finally, in order to explore the limits of our proposed models and match capacity of the architectures used in concurrent papers (e.g. [13]), we train the S^4L-Rotation model with a more powerful architecture, such as ResNet152v2 $2\times$wider, and also use large computational budget to tune hyperparameters. In this case our model achieves even better results: $86.41\,\%$ top-5 accuracy with $10\,\%$ labels and $57.50\,\%$ with $1\,\%$ labels.

5. Semi-supervised Learning is Complementary to S^4L

Since we found that different types of models perform similarly well, the natural next question is whether they are complementary, in which case a combination would lead to an even better model, or whether they all reach a common "intrinsic" performance plateau.

In this section, we thus describe our *Mix Of All Models* (MOAM). In short: in a first step, we combine S^4L-Rotation and VAT+EntMin to learn a $4\times$ wider [17] model. We then use this model in order to generate pseudo labels for a second training step, followed by a final fine-tuning step. Results of the final model, as well as the models ob-

Table 2. Comparing our MOAM to previous methods in the literature on ILSVRC-2012 with 10 % of the labels. Note that *different models use different architectures*, larger than those in Table 1.

	labels	Top-5	Top-1
MOAM full (*proposed*)	10%	**91.23**	**73.21**
MOAM + pseudo label (*proposed*)	10%	89.96	71.56
MOAM (*proposed*)	10%	88.80	69.73
ResNet50v2 (4×wider)	10%	81.29	58.15
VAE + Bayesian SVM [32]	10%	64.76	48.41
Mean Teacher [41]	10%	90.89	-
†UDA [43]	10%	88.52†	68.66†
†CPCv2 [13]	10%	84.88†	64.03†
Training with all labels:			
ResNet50v2 (4×wider)	100%	94.10	78.57
MOAM (*proposed*)	100%	**94.97**	**80.17**
†UDA [43]	100%	94.45†	79.04†
†CPCv2 [13]	100%	93.35†	-

† *marks concurrent work.*

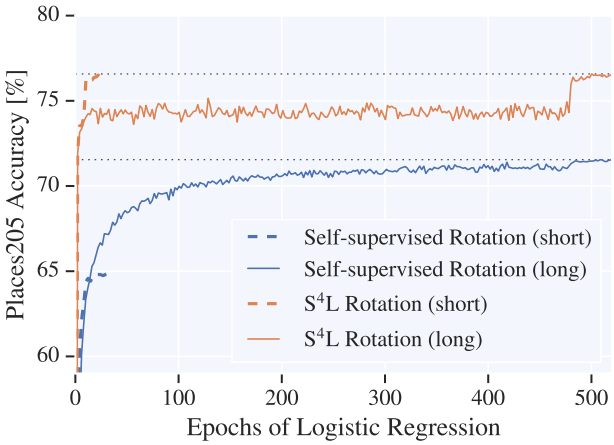

Figure 2. Places205 learning curves of logistic regression on top of the features learned by pre-training a self-supervised versus S^4L-Rotation model on ILSVRC-2012. The significantly faster convergence ("long" training schedule vs. "short" one) suggests that more easily separable features are learned.

tained in the two intermediate steps, are reported in Table 2 along with previous results reported in the literature.

Step 1: Rotation+VAT+EntMin In the first step, our model jointly optimizes the S^4L-Rotation loss and the VAT and EntMin losses. We iterated on hyperparameters for this setup in a less structured way than in our controlled experiments above (always on our custom validation set) and only mention the final values here. Our model was trained with batch size 128, learning rate 0.1, weight decay $2 \cdot 10^{-4}$, training for 200 epochs with linear learning rate rampup up to epoch 10, then 10-fold decays at 100, 150, and 190 epochs. We use inception crop augmentation as well as horizontal mirroring. We used the following relative loss weights: $w_{\text{sup}} = 0.3$, $w_{\text{rot}} = 0.7$, $w_{\text{vat}} = 0.3$, $w_{\text{entmin}} = 0.3$. We tried a few heuristics for setting those weights automatically, but found that manually tuning them led to better performance. We also applied Polyak averaging to the model parameters, choosing the decay factor such that parameters decay by 50 % over each epoch. Joint training of these losses consistently improve over the models with a single objective. The model obtained after this first step achieves 88.80% top-5 accuracy on the ILSVRC-2012 dataset.

Step 2: Retraining on Pseudo Labels Using the above model, we assign pseudo labels to the full dataset by averaging predictions across five crops and four rotations of each image[3]. We then train the same network again in the exact same way (*i.e.* with all the losses) except for the following three differences: (1) the network is initialized using the weights obtained in the first step (2) every example has a label: the pseudo label (3) due to this, an epoch now corresponds to the full dataset; we thus train for 18 epochs, decaying the learning rate after 6 and 12 epochs.

Step 3: Fine-tuning the model Finally, we fine-tune the model obtained in the second step on the original 10 % labels only. This step is trained with weight decay $3 \cdot 10^{-3}$ and learning rate $5 \cdot 10^{-4}$ for 20 epochs, decaying the learning rate 10-fold every 5 epochs.

Remember that all hyper-parameters described here were selected on our custom validation set which is taken from the training set. The final model "MOAM (full)" achieves 91.23 % top-5 accuracy, which sets the new state-of-the-art.

We conduct additional experiments and report performance of MOAM (*i.e.* only Step 1) with 100 % labels in Table 2. Interestingly, MOAM achieves promising results even in the high-data regime with 100 % labels, outperforming the fully supervised baseline: +0.87 % for top-5 accuracy and +1.6 % for top-1 accuracy.

6. Transfer of Learned Representations

Self-supervision methods are typically evaluated in terms of how generally useful their learned representation is. This is done by treating the learned model as a fixed feature extractor, and training a linear logistic regression model on top the features it extracts on a different dataset, usually Places205 [45]. We perform such an evaluation on our S^4L models in order to gain some insight into the generality of the learned features, and how they compare to those obtained by pure self-supervision.

[3]Generating pseudo-labels using 20 crops only slightly improved performance by 0.25 %, but is cheap and simple.

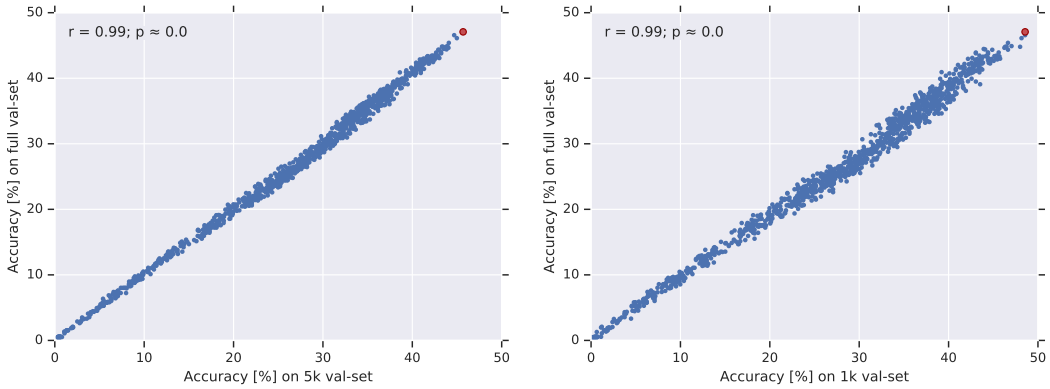

Figure 3. Correlation between validation score on a (custom) validation set of 1000, 5000, and 50 046 images on ILSVRC-2012. Each point corresponds to a *trained model* during a sweep for plain supervised baseline for the 1 % labeled case. The best model according to the validation set of 1 000 is marked in red. As can be seen, evaluating our models even with only a single validation image per class is robust, and in particular selecting an optimal model with this validation set works as well as with the full validation set.

We closely follow the protocol defined by [17]. The representation is extracted from the pre-logits layer. We use stochastic gradient descent (SGD) with momentum for training these linear evaluation models with a minibatch size of 2048 and an initial learning rate of 0.1, warmed up in the first epoch.

While Kolesnikov *et al*. [17] show that a very long training schedule (520 epochs) is required for the linear model to converge using self-supervised representations, we observe dramatically different behaviour when evaluating our self-supervised semi-supervised representations. Figure 2 shows the accuracy curve of the plain self-supervised rotation method [17] and our proposed S^4L-Rotation method trained on 10 % of ILSVRC-2012. As can be seen, the logistic regression is able to find a good separating hyperplane in very few epochs and then plateaus, whereas in the self-supervised case it struggles for a very long number of epochs. This indicates that the addition of labeled data leads to much more separable representations, even across datasets.

We further investigate the gap between the representation learned by a good S^4L model (MOAM) and a corresponding baseline trained on 100 % of the labels (the baseline from Table 2). Surprisingly, we found that the representation learned by "MOAM (full)" transfers slightly better than the baseline, which used ten times more labelled data: 83.3 % accuracy vs. 83.1 % accuracy, respectively. We provide full details of this experiment in the Supplementary Material.

7. Is a Tiny Validation Set Enough?

Current standard practice in semi-supervised learning is to use a subset of the labels for training on a large dataset, but still perform model selection using scores obtained on the full validation set.[4] But having a large labeled validation set at hand is at odds with the promised practicality of semi-supervised learning, which is all about having only few labeled examples. This fact has been acknowledged by [33], but has been mostly ignored in the semi-supervised literature. Oliver *et al*. [30] questions the viability of tuning with small validation sets by comparing the estimated model accuracy on small validation sets. They find that the variance of the estimated accuracy gap between two models can be larger than the actual gap between those models, hinting that model selection with small validation sets may not be viable. That said, they did not empirically evaluate whether it's possible to find the *best* model with a small validation set, especially when choosing hyperparameters for a particular semi-supervised method.

We now describe our analysis of this important question. We look at the many models we trained for the plain supervised baseline on 1 % of ILSVRC-2012. For each model, we compute a validation score on a validation set of 1000 labeled images (*i.e*. one labeled image per class), 5000 labeled images (*i.e*. five labeled images per class), and compare these scores to those obtained on a "full-size" validation set of 50 046 labeled images. The result is shown in Figure 3 and it is striking: there is a very strong correlation between performance on the tiny and the full validation set. Especially, while in parts there is high variability, those hyperparameters which work best do so in either case. Most notably, the *best model* tuned on a small validation set is also the best model tuned on a large validation set. We thus conclude that for selecting hyperparameters of a model, a tiny validation set is enough.

[4] To make matters worse, in the case of ILSVRC-2012, this validation set is used both to select hyperparameters as well as to report final performance. Remember that we avoid this by creating a custom validation set from part of the training set for all hyperparameter selections.

8. Discussion and Future Work

In this paper, we have bridged the gap between self-supervision methods and semi-supervised learning by suggesting a framework (S^4L) which can be used to turn any self-supervision method into a semi-supervised learning algorithm.

We instantiated two such methods: S^4L-Rotation and S^4L-Exemplar and have shown that they perform competitively to methods from the semi-supervised literature on the challenging ILSVRC-2012 dataset. We further showed that S^4L methods are complementary to existing semi-supervision techniques, and MOAM, our proposed combination of those, leads to state-of-the-art performance.

While all of the methods we investigated show promising results for learning with $10\,\%$ of the labels on ILSVRC-2012, the picture is much less clear when using only $1\,\%$. It is possible that in this low data regime, when only 13 labeled examples per class are available, the setting fades into the few-shot scenario, and a very different set of methods would be required for reaching much better performance.

Nevertheless, we hope that this work inspires other researchers in the field of self-supervision to consider extending their methods into semi-supervised methods using our S^4L framework, as well as researchers in the field of semi-supervised learning to take inspiration from the vast amount of recently proposed self-supervision methods.

Acknowledgements. We thank the Google Brain Team in Zürich, and especially Sylvain Gelly for discussions.

References

[1] B. Athiwaratkun, M. Finzi, P. Izmailov, and A. G. Wilson. There are many consistent explanations of unlabeled data: Why you should average. *ICLR*, 2019. 2

[2] D. Berthelot, N. Carlini, I. Goodfellow, N. Papernot, A. Oliver, and C. Raffel. Mixmatch: A holistic approach to semi-supervised learning. *arXiv preprint arXiv:1905.02249*, 2019. 2

[3] M. Caron, P. Bojanowski, A. Joulin, and M. Douze. Deep clustering for unsupervised learning of visual features. *European Conference on Computer Vision (ECCV)*, 2018. 3

[4] O. Chapelle, B. Schlkopf, and A. Zien. *Semi-Supervised Learning*. The MIT Press, 1st edition, 2010. 2

[5] C. Doersch, A. Gupta, and A. A. Efros. Unsupervised visual representation learning by context prediction. In *International Conference on Computer Vision (ICCV)*, 2015. 3

[6] A. Dosovitskiy, J. T. Springenberg, M. Riedmiller, and T. Brox. Discriminative unsupervised feature learning with convolutional neural networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2014. 3

[7] F. Ebert, S. Dasari, A. X. Lee, S. Levine, and C. Finn. Robustness via retrying: Closed-loop robotic manipulation with self-supervised learning. *Conference on Robot Learning (CoRL)*, 2018. 3

[8] M. Everingham, S. A. Eslami, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes challenge: A retrospective. *IJCV*, 111(1), 2015. 1

[9] C. D. Freeman and J. Bruna. Topology and geometry of half-rectified network optimization. *arXiv preprint arXiv:1611.01540*, 2016. 12

[10] S. Gidaris, P. Singh, and N. Komodakis. Unsupervised representation learning by predicting image rotations. In *International Conference on Learning Representations (ICLR)*, 2018. 1, 3

[11] Y. Grandvalet and Y. Bengio. Semi-supervised learning by entropy minimization. In L. K. Saul, Y. Weiss, and L. Bottou, editors, *Advances in Neural Information Processing Systems 17*, pages 529–536. MIT Press, 2005. 2, 4, 5, 12

[12] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 6

[13] O. J. Hénaff, A. Razavi, C. Doersch, S. Eslami, and A. v. d. Oord. Data-efficient image recognition with contrastive predictive coding. *arXiv preprint arXiv:1905.09272*, 2019. 2, 6, 7

[14] A. Hermans, L. Beyer, and B. Leibe. In Defense of the Triplet Loss for Person Re-Identification. *arXiv preprint arXiv:1703.07737*, 2017. 3

[15] E. Jang, C. Devin, V. Vanhoucke, and S. Levine. Grasp2Vec: Learning object representations from self-supervised grasping. In *Conference on Robot Learning*, 2018. 3

[16] D. P. Kingma, D. J. Rezende, S. Mohamed, and M. Welling. Semi-supervised learning with deep generative models. *CoRR*, abs/1406.5298, 2014. 2

[17] A. Kolesnikov, X. Zhai, and L. Beyer. Revisiting self-supervised visual representation learning. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 1, 3, 5, 6, 8, 12

[18] A. Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009. 2

[19] S. Laine and T. Aila. Temporal ensembling for semi-supervised learning. *CoRR*, abs/1610.02242, 2016. 2

[20] D.-H. Lee. Pseudo-label : The simple and efficient semi-supervised learning method for deep neural networks. *ICML 2013 Workshop : Challenges in Representation Learning (WREPL)*, 07 2013. 2, 4, 5, 12

[21] H. Li, Z. Xu, G. Taylor, C. Studer, and T. Goldstein. Visualizing the loss landscape of neural nets. In *Advances in Neural Information Processing Systems*, pages 6391–6401, 2018. 12

[22] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft COCO: Common objects in context. In *ECCV*. Springer, 2014. 1

[23] B. Liu, Z. Wu, H. Hu, and S. Lin. Deep metric transfer for label propagation with limited annotated data. *arXiv preprint arXiv:1812.08781*, 2018. 2

[24] T. Miyato, S.-i. Maeda, M. Koyama, and S. Ishii. Virtual adversarial training: A regularization method for supervised and semi-supervised learning. *arXiv preprint arXiv:1704.03976*, 2017. 2, 4, 5, 12

[25] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011*, 2011. 2

[26] M. Noroozi and P. Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *European Conference on Computer Vision (ECCV)*, 2016. 3

[27] M. Noroozi, H. Pirsiavash, and P. Favaro. Representation learning by learning to count. In *International Conference on Computer Vision (ICCV)*, 2017. 3

[28] M. Noroozi, A. Vinjimoor, P. Favaro, and H. Pirsiavash. Boosting self-supervised learning via knowledge transfer. *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 3

[29] A. Odena. Semi-supervised learning with generative adversarial networks. *arXiv preprint arXiv:1606.01583*, 2016. 2

[30] A. Oliver, A. Odena, C. A. Raffel, E. D. Cubuk, and I. Goodfellow. Realistic evaluation of deep semi-supervised learning algorithms. In *Advances in Neural Information Processing Systems*, pages 3239–3250, 2018. 2, 4, 5, 6, 8

[31] A. Owens and A. A. Efros. Audio-visual scene analysis with self-supervised multisensory features. *European Conference on Computer Vision (ECCV)*, 2018. 3

[32] Y. Pu, Z. Gan, R. Henao, X. Yuan, C. Li, A. Stevens, and L. Carin. Variational autoencoder for deep learning of images, labels and captions. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 2352–2360. 2016. 4, 7

[33] A. Rasmus, M. Berglund, M. Honkala, H. Valpola, and T. Raiko. Semi-supervised learning with ladder networks. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 3546–3554. Curran Associates, Inc., 2015. 2, 4, 8

[34] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. 1, 2

[35] T. Salimans, I. J. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen. Improved techniques for training gans. *CoRR*, abs/1606.03498, 2016. 2

[36] N. Sayed, B. Brattoli, and B. Ommer. Cross and learn: Cross-modal self-supervision. *arXiv preprint arXiv:1811.03879*, 2018. 3

[37] P. Sermanet, C. Lynch, Y. Chebotar, J. Hsu, E. Jang, S. Schaal, and S. Levine. Time-contrastive networks: Self-supervised learning from video. *arXiv preprint arXiv:1704.06888*, 2017. 3

[38] R. Shu, H. Bui, H. Narui, and S. Ermon. A DIRT-t approach to unsupervised domain adaptation. In *International Conference on Learning Representations*, 2018. 4

[39] M. Soltanolkotabi, A. Javanmard, and J. D. Lee. Theoretical insights into the optimization landscape of over-parameterized shallow neural networks. *IEEE Transactions on Information Theory*, 65(2):742–769, 2019. 12

[40] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 3

[41] A. Tarvainen and H. 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. 2, 4, 7

[42] V. Verma, A. Lamb, J. Kannala, Y. Bengio, and D. Lopez-Paz. Interpolation consistency training for semi-supervised learning. *arXiv preprint arXiv:1903.03825*, 2019. 2

[43] Q. Xie, Z. Dai, E. Hovy, M.-T. Luong, and Q. V. Le. Unsupervised data augmentation. *arXiv preprint arXiv:1904.12848*, 2019. 2, 7

[44] R. Zhang, P. Isola, and A. A. Efros. Colorful image colorization. In *European Conference on Computer Vision (ECCV)*, 2016. 3

[45] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva. Learning deep features for scene recognition using places database. In *Advances in neural information processing systems*, pages 487–495, 2014. 7, 12

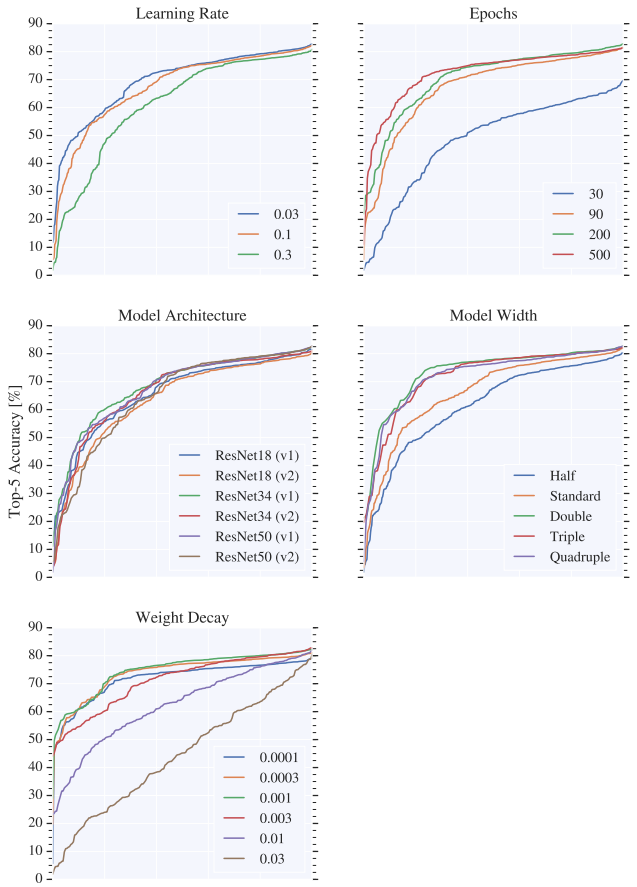

Figure 4. The "hypersweep curves" for the supervised baseline trained on 10 % of ILSVRC-2012. See text for details.

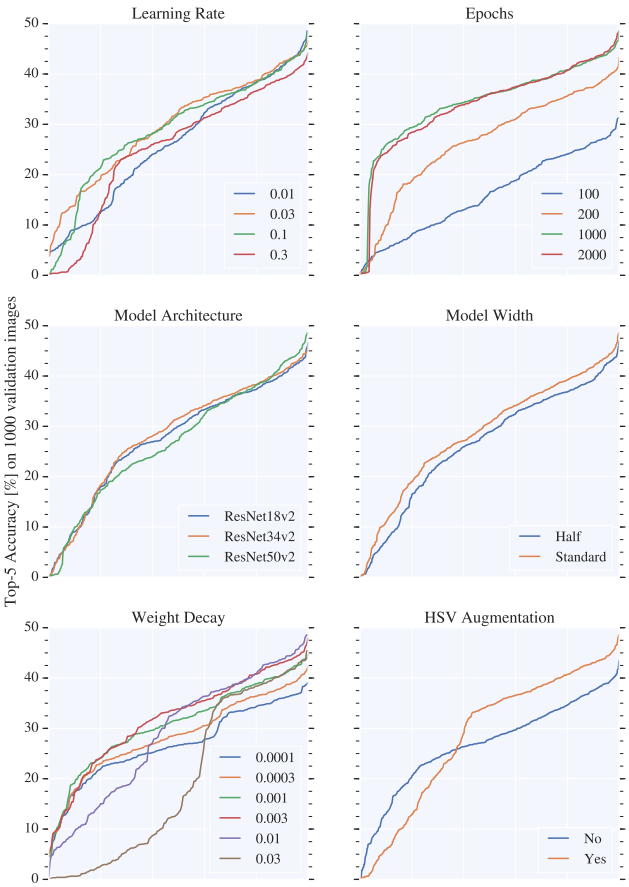

Figure 5. The "hypersweep curves" for the supervised baseline trained on 1 % of ILSVRC-2012. See text for details.

A. Detailed Results of the Supervised Baselines

Since we performed quite extensive hyperparameter search and trained many models in order to find a solid fully-supervised baseline on 10 % and 1 % of ILSVRC-2012, we believe that it is valuable to report the full results to the community, instead of just providing the final best model.

We present the results in the form of what we call "hypersweep curves" in Figures 4 and 5.

Each plot shows a large collection of models – *each point on each plot is a fully trained model*. The curves are sorted by accuracy, allowing testing sensitivity to different hyperparameters, not only comparing the best model.

For each curve, we plot the accuracy of models where one of the hyperparameters is fixed.

Thus, by comparing curves, one can see:

1. Which value of a hyperparameter performs best by looking at which curve's rightmost point is highest.

2. How sensitive the model is to a hyperparameter *in the best case* by looking at how far apart the curves are from eachother at their rightmost point.

3. How robust a hyperparameter is *on average* by looking at how similar the curves are overall.

4. How independent a specific hyperparameter value is from all others by looking at the curve's shape, and whether curves cross-over (strong interplay) or not (strong independence).

While the results shown in Figure 4 use the full (custom) validation set, those in Figure 5 were computed using the validation set of size 1000, *i.e.* with only one image per class. As we have shown in Section 7, this is sufficient to determine the best hyperparameters, and we encourage the community to follow this more realistic protocol.

As can be seen, weight decay and number of training epochs are the two things which matter most when training using only a fraction of ILSVRC-2012.

Perhaps the most surprising finding is that, contrary to current folklore, *reducing model capacity is detrimental*

to performance on the smaller dataset. Neither reducing depth, nor reducing width improve performance. In fact, the deeper and wider models still outperform their shallower and thinner counterparts, even when using only $1\,\%$ of the training data. Even more so, the wider models are more robust to other hyperparameter's values as evidenced by their curves being significantly higher on the left end. This is in line with recent findings suggesting wider models ease optimization [21, 9, 39].

Furthermore, when reducing the dataset size to $1\,\%$, we found that adding the same color augmentation as introduced by Exemplar is helpful. We thereafter tried adding it to our best few models on $10\,\%$, but it did not help there.

Finally, while in the $1\,\%$ case, learning-rate of 0.1 and 0.01 seem to perform equally well in the good cases (right hand side of curves), we manually inspected training curves and found that 0.1 is significantly less robust, typically not learning anything before the first decay, and only catching up later on.

While we trained thousands of models in order to rigorously test multiple hypotheses (such as that of reducing model capacity), almost all boost in performance could have been achieved in just a few dozen trials with intuitively important hyperparameters (weight decay and epochs), which would take about a week on a modern four-GPU machine.

Overall, we hope that this thorough baseline investigation inspires the semi-supervised learning community to be more careful with baselines, as those that we found perform almost $20\,\%$ absolute better than those previously reported in the literature.

B. Randomness of S^4L

Table 3. S^4L performance for 9 runs with random image subsets. Top-5 accuracies [%] are reported as mean±standard deviation.

Method	10% ImageNet	1% ImageNet
S^4L-Rotation	83.91 ± 0.13	53.47 ± 0.22
S^4L-Exemplar	83.76 ± 0.06	46.61 ± 0.25

There are two factors of randomness of a semi supervised model: (1) labeled subset sampling, (2) run with different seeds. In order to estimate the randomness in the performance we train 9 models with random data subsets and random seeds for our proposed S^4L method. Table 3 presents the detailed results. Overall, we observe that standard deviation is fairly small across both subsets and different runs and, therefore, our empirical evaluation provides robust comparison of various techniques.

C. More Results in the Transfer Setup

In this section we present more results from the transfer evaluation task on Places205 [45]. Table 4 shows the results for the models mentioned in our main paper. For each method, we select the best model and evaluate its transfer to Places205.

Table 4. Accuracy (in percent) obtained by various individual methods when transferring their representation to the Places205 dataset using linear models on frozen representations. All methods use the same plain ResNet50v2 base model, except for the ones marked by *, which use a $4\times$ wider network. When it was necessary, a $^+$ marks longer transfer training of 520 epochs. The "%-labels" column shows the percentage of ILSVRC-2012 labels that was used for training the model.

Method	%-labels	top-5	top-1
Supervised	1	65.4	36.2
Supervised	10	75.0	44.7
Supervised	100	81.9	52.5
Supervised*	100	83.1	53.7
SS Rotation$^+$ [17]	0	71.4	41.7
SS Exemplar$^+$ [17]	0	69.0	39.8
Pseudolabels [20]	1	71.6	41.8
VAT [24]	1	64.9	35.9
VAT + EntMin [11]	1	65.9	36.4
Pseudolabels [20]	10	78.1	48.2
VAT [24]	10	76.4	45.8
VAT + EntMin [11]	10	76.4	46.2
SS Rotation [17] + Fine-tune	1	66.1	36.3
SS Exemplar [17] + Fine-tune	1	60.0	31.1
SS Rotation [17] + Fine-tune	10	75.4	45.9
SS Exemplar [17] + Fine-tune	10	75.6	45.9
S^4L-Rotation	1	67.3	38.0
S^4L-Exemplar	1	61.2	32.2
S^4L-Rotation	10	76.4	46.6
S^4L-Exemplar	10	75.9	45.9
MOAM* full	10	83.3	54.2
MOAM* + pseudo label	10	83.3	54.2
MOAM*	10	79.2	49.5

We follow the same setup as [17] to train a linear models with SGD on top of frozen representations. The only difference is the training epochs, we train for 30 epochs in total with learning rate decayed at 10 and 20 epochs respectively. The learning rate is linearly ramped up for the first epoch. Kolesnikov et.al. [17] train for 520 epochs with learning rate decays at 480 and 500 epochs. The schedule used in our paper is much shorter because of our finding that representation learned with labels are more separable and converges significantly faster. (See in Section 6 of the main paper for details.) To make fair comparison with the self-supervised models, results in Table 4 with 0% labels are trained for 520 epochs to ensure their convergence.

From the plain supervised baselines, we observe that either more labels or wider networks lead to more transferable representations. Surprisingly, we found that pseudo labels outperforms the other two semi-supervised baselines in the transfer setup. On the 1% labels evaluation setup, pseudo labels achieves the best result comparing to the other methods. With 10% labels, S^4L is comparable to the semi-supervised baselines, and our MOAM clearly outperforms all other models trained on 10% of labels. More interestingly, the *MOAM (full)* model on 10% is slightly better than the 100% supervised baseline with the same $4\times$ wider network. This indicates that learning a model with multiple losses may lead to representations that generalize better to unseen tasks.