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

Xiaohua Zhai, Avital Oliver, Alexander Kolesnikov, Lucas Beyer

Google Research, Brain Team

Contribution

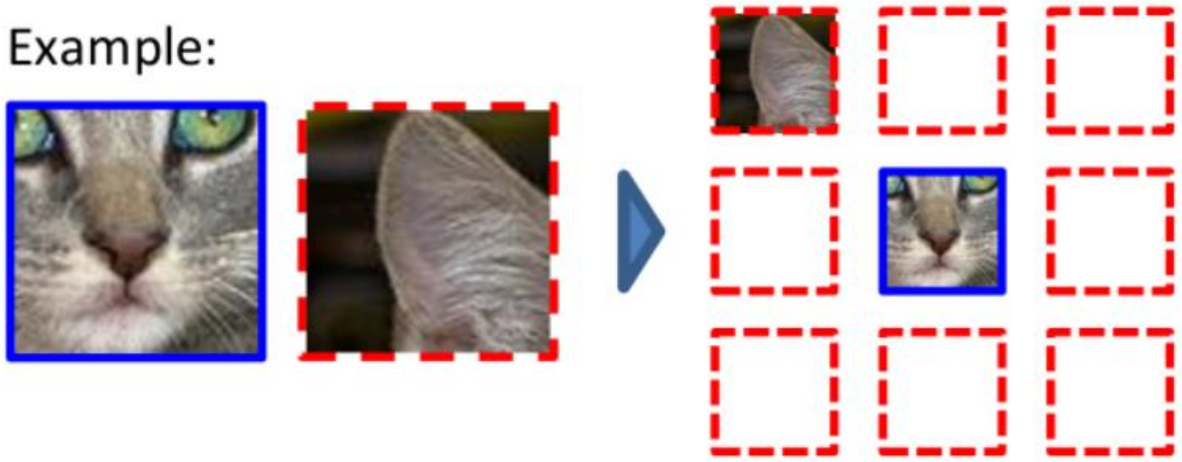
- Propose a general technique to merge semi-supervised learning with self-supervised representation learning (S^4L)
- Demonstrates that this S^4L method is competitive in image-based classification.
- Demonstrates state-of-the-art performance (image classification on) by leveraging both S^4L and traditional semi-supervised learning.

Motivation

- Deep Neural Networks are commonly used for image-based tasks (e.g. classification).
- Generating adequate labeled data for training is often impractical, especially for highly specific tasks.
- Better performance from unsupervised training methods is desired.

Self-supervised vs Semi-supervised Learning

- Both forms of unsupervised learning (no explicit training labels).
- Self-supervised: Training network with a pretext task for which data labels can easily be generated
  - Training on this task teaches network features that are important to the primary task.
  - Ex) Train network to determine relative location of two non-overlapping image patches



Semi-supervised Learning cont.

- Network learning that involves training with both labelled and unlabeled data
- Ex) *Pseudo-Labeling*
  - Train the network with labeled data first
  - Make network predictions on unlabeled data
  - Assign predicted labels to samples with high-confidence predictions
  - Retrain with labeled data
- Evaluation Standard: 10/90 labelled/unlabeled data split

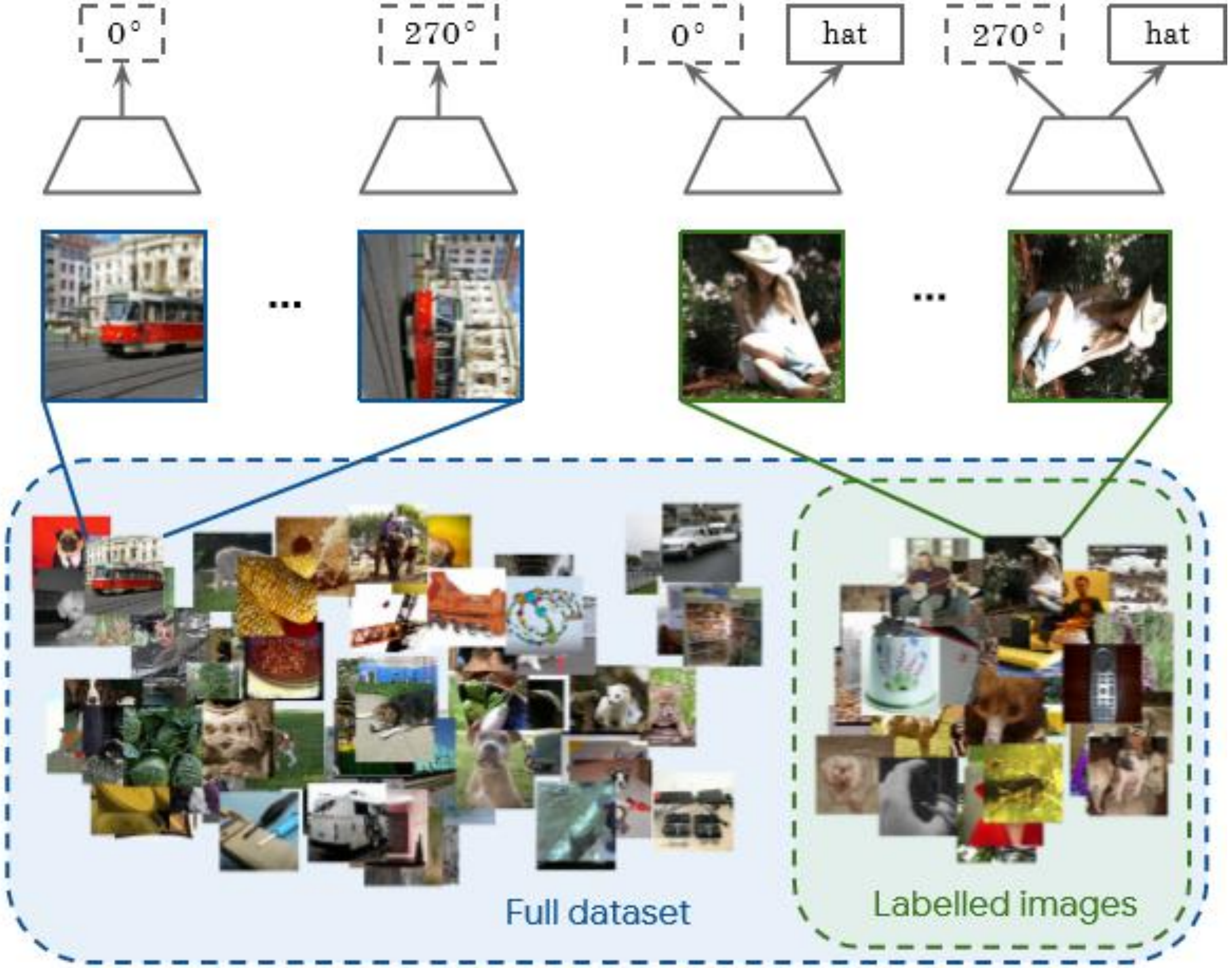
General Method of S^4L

- Self-Supervised Semi-Supervised Learning
  - Essentially including a self-supervised learning objective in a semi-supervised objective/framework
- Objective function for test batch:

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

- $\mathcal{L}_l$ *: Standard cross-entropy classification loss for labeled data*
- $\mathcal{L}_u$ *: Loss function for unsupervised images*
- *w : scalar weighting factor*

S^4L Example



- *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 unlabeled 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.

Self-supervised approaches tested with S^4L

- S^4L-Rotation
  - Unlabeled and labeled images are rotated in 1 of 4 cardinal directions.
  - Part of label is the degree of rotation.

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

  - r: degree of rotation
  - *R*: set of all four rotations

Self-supervised approaches tested with S4L

- S4L-Exemplar
  - Unlabeled images undergo a range of image transformations
  - Different transformations of the same image share the same label.
  - This paper used 8 different transformations total.

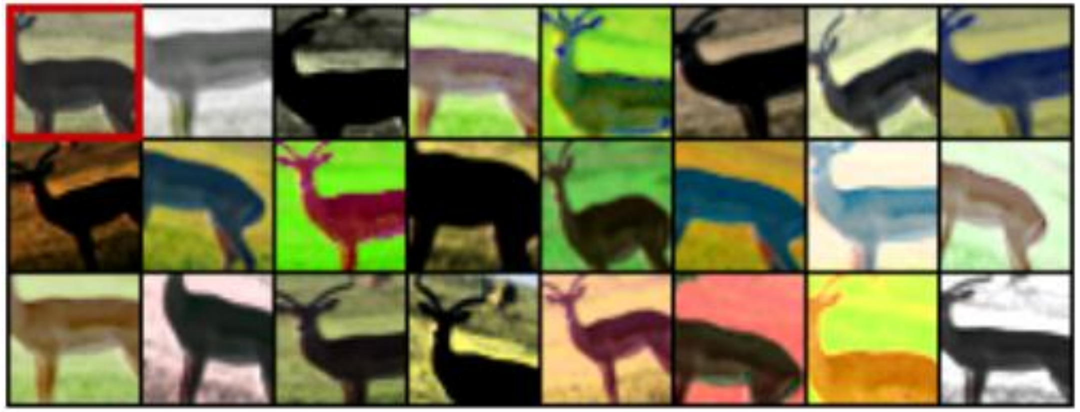


Figure 2: Several random transformations applied to one of the patches extracted from the STL unlabeled dataset. The original ('seed') patch is in the top left corner.

Experimental Objectives

- Test S^4L with image classification and compare performance to supervised, unsupervised, and self-supervised methods.
- Investigate if combining S^4L with other semi-supervised methods produces even greater performance.

General Experiment Parameters

- Database used: ILSVRC-2012
  - Over 1 million images!
  - Different experiments done with labeled/unlabeled data splits of **10/90** and **1/99**.

- Network Architecture: ResNet50v2

- Training ‘dry runs’ first done to optimize hyperparameters

Methods Tested

- Supervised only Baseline:
  - For 1% data case, random color data augmentation used
- Semi-Supervised Baselines:
  - Pseudo-Label
  - Virtual Adversarial Training (VAT)
  - VAT+ Entropy Minimization (VAT+EntMin)
- Self-Supervised Baselines:
  - Rotation
  - Exemplar
  - After pre-training, either “linear” or “fine-tune” supervised learning was used

Methods Tested (cont.)

- S^4L
  - Rotation
  - Exemplar
  - Set self-supervised loss weight (w) to 1
  - Self-supervised loss applied to labeled and unlabeled examples
  - Supervised loss not applied to ‘copies’ of images
- Hyperparameters fine-tuned optimally for each model
  - Weight decay
  - Learning Rate
  - Number of Training Epochs

Results

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

Testing combined S^4L and semi-supervised model

- *Mix of All Models* (MOAM)
- Step 1: Combine S^4L-Rotation with VAT+EntMin with 4x wider ResNet50v2
- Step 2: Retrain model using Pseudo-labels
- Step 3: Fine-tune model with 10% of labeled data

Results

	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:

	labels	Top-5	Top-1
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.*

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.

Transfer of Learned Representations

- Objective is to investigate how generally useful the learned feature representation for S^4L is.
- Method: Take trained model as fixed feature extractor, then train linear logistic regression model on top of extractor.
  - Train on entirely different dataset (Places205) and measure accuracy/ convergence of regression model.

Results of Logistic Regression

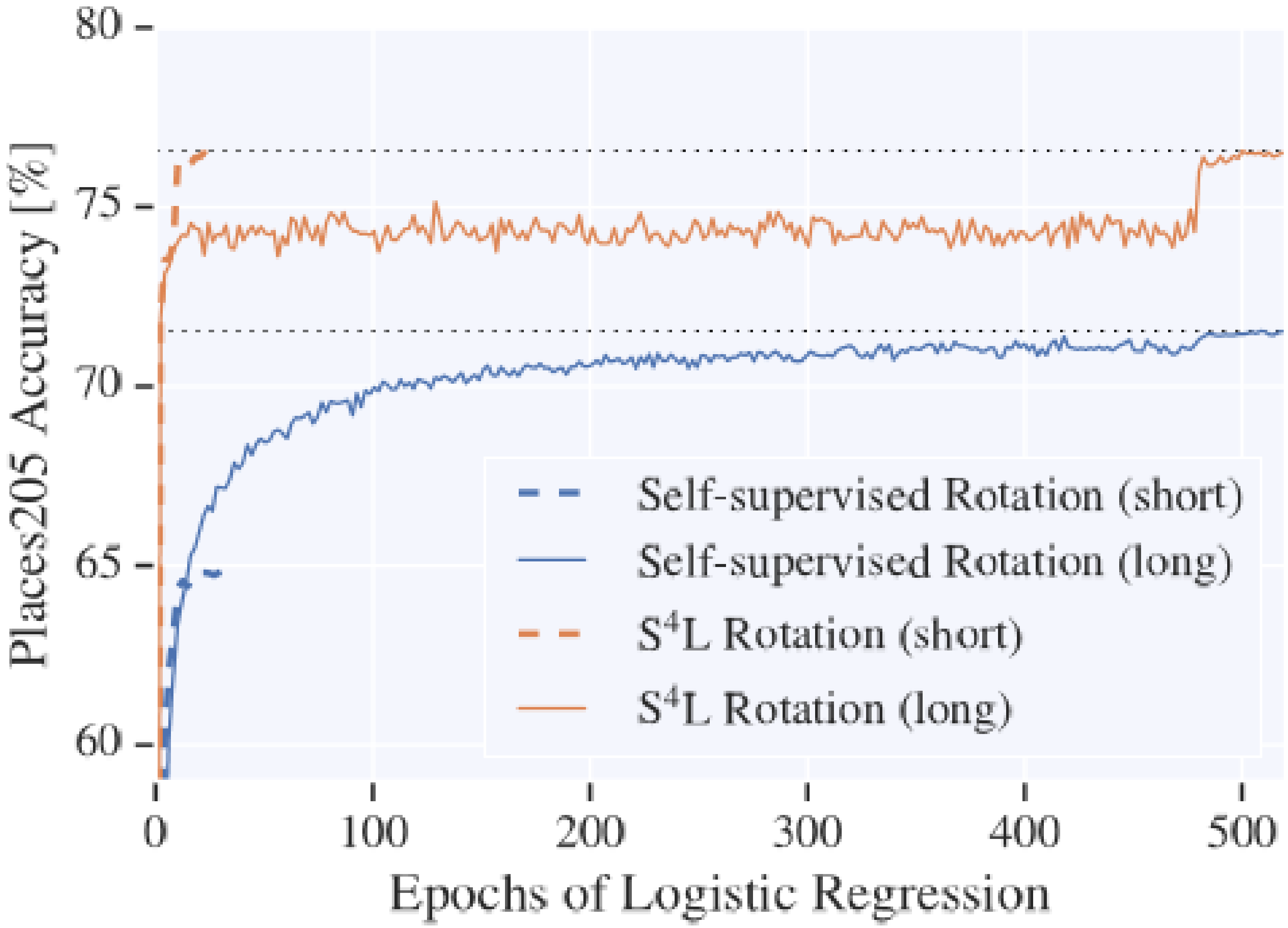


Figure 2. Places205 learning curves of logistic regression on top of the features learned by pre-training a self-supervised versus S^4LRotation model on ILSVRC-2012. The significantly faster convergence (“long” training schedule vs. “short” one) suggests that more easily separable features are learned.

Is Tiny Validation Set Enough

- Problem with many unsupervised methods is that they are still validated on large labelled datasets.
  - Validation feedback is used to fine-tune model.
- Major issue: methods still rely on big labelled data for validation.
- Can accurate feedback still be achieved with a smaller validation set?

Validation Results Mostly Independent of Set Size

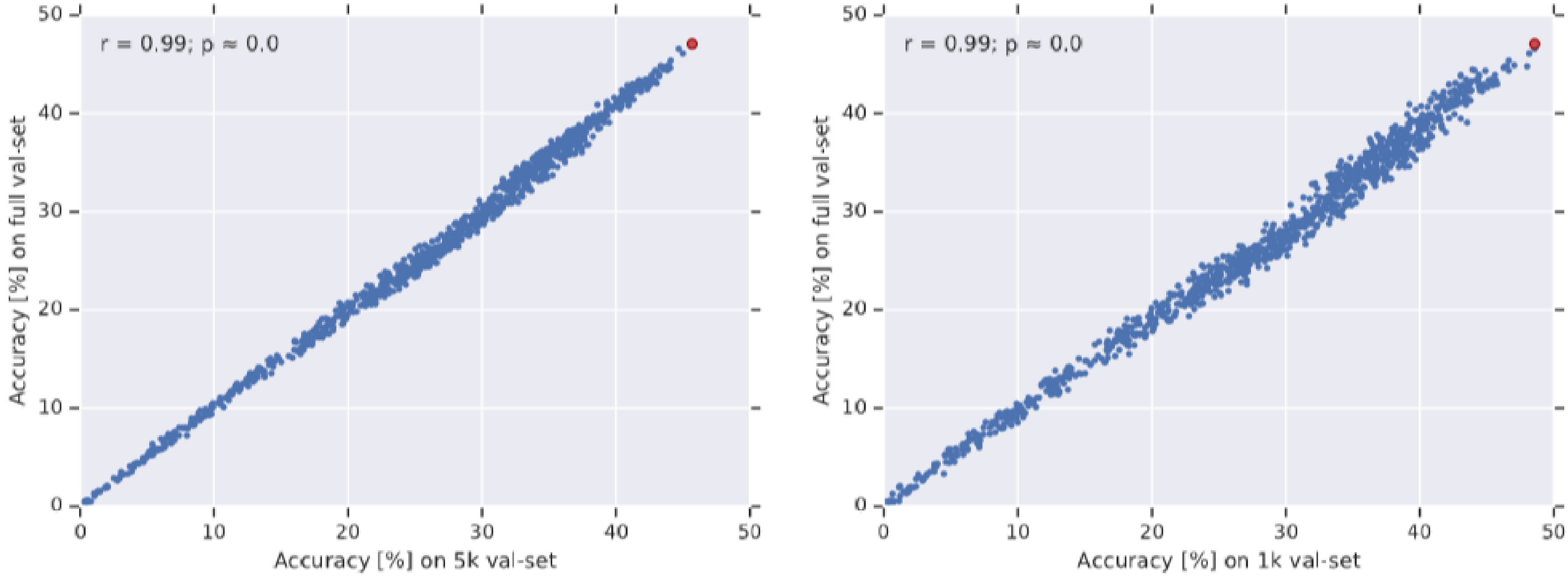


Figure 3. Correlation between validation score on a (custom) validation set of 1000, 5000, and 50 046 images on ILSVRC-2012. Each point corresponds to a *trained model* during a sweep for plain supervised baseline for the 1 % labeled case. The best model according to the validation set of 1 000 is marked 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.

Conclusion

- S^4L method can merge self-supervised and supervised training methods into one semi-supervised training model.
- Training from S^4L is somewhat complementary to other semi-supervised training methods.
- Features learned with S^4L appear to be generally useful when compared to self-supervised feature learning.
- Small validation sets are enough to provide S^4L model feedback.