Meta Pseudo Labels

Hieu Pham, Zihang Dai, Qizhe Xie, Quoc V. Le
Google AI, Brain Team, Mountain View, CA 94043
{hyhieu,zihangd,qizhex,qvl}@google.com

Abstract

We present Meta Pseudo Labels, a semi-supervised learning method that achieves a new state-of-the-art top-1 accuracy of 90.2% on ImageNet, which is 1.6% better than the existing state-of-the-art [16]. Like Pseudo Labels, Meta Pseudo Labels has a teacher network to generate pseudo labels on unlabeled data to teach a student network. However, unlike Pseudo Labels where the teacher is fixed, the teacher in Meta Pseudo Labels is constantly adapted by the feedback of the student's performance on the labeled dataset. As a result, the teacher generates better pseudo labels to teach the student.[1]

1. Introduction

The methods of Pseudo Labels or self-training [57, 81, 55, 36] have been applied successfully to improve state-of-the-art models in many computer vision tasks such as image classification (e.g., [79, 77]), object detection, and semantic segmentation (e.g., [89, 51]). Pseudo Labels methods work by having a pair of networks, one as a teacher and one as a student. The teacher generates pseudo labels on unlabeled images. These pseudo labeled images are then combined with labeled images to train the student. Thanks to the abundance of pseudo labeled data and the use of regularization methods such as data augmentation, the student learns to become better than the teacher [77].

Despite the strong performance of Pseudo Labels methods, they have one main drawback: if the pseudo labels are inaccurate, the student will learn from inaccurate data. As a result, the student may not get significantly better than the teacher. This drawback is also known as the problem of confirmation bias in pseudo-labeling [2].

In this paper, we design a systematic mechanism for the teacher to correct the bias by observing how its pseudo labels would affect the student. Specifically, we propose Meta Pseudo Labels, which utilizes the feedback from the student to inform the teacher to generate better pseudo labels. In our implementation, the feedback signal is the performance of the student on the labeled dataset. This feedback signal is used as a reward to train the teacher throughout the course of the student's learning. In summary, the teacher and student of Meta Pseudo Labels are trained in parallel: (1) the student learns from a minibatch of pseudo labeled data annotated by the teacher, and (2) the teacher learns from the reward signal of how well the student performs on a minibatch drawn from the labeled dataset.

We experiment with Meta Pseudo Labels, using the ImageNet [56] dataset as labeled data and the JFT-300M dataset [26, 60] as unlabeled data. We train a pair of EfficientNet-L2 networks, one as a teacher and one as a student, using Meta Pseudo Labels. The resulting student network achieves the top-1 accuracy of 90.2% on the ImageNet ILSVRC 2012 validation set [56], which is 1.6% better than the previous record of 88.6% [16]. This student model also generalizes to the ImageNet-ReaL test set [6], as summarized in Table 1. Small scale semi-supervised learning experiments with standard ResNet models on CIFAR-10-4K, SVHN-1K, and ImageNet-10% also show that Meta Pseudo Labels outperforms a range of other recently proposed methods such as FixMatch [58] and Unsupervised Data Augmentation [76].

Datasets	**ImageNet** Top-1 Accuracy	**ImageNet-ReaL** Precision@1
Previous SOTA [16, 14]	88.6	90.72
Ours	**90.2**	**91.02**

Table 1: Summary of our key results on ImageNet ILSVRC 2012 validation set [56] and the ImageNet-ReaL test set [6].

2. Meta Pseudo Labels

An overview of the contrast between Pseudo Labels and Meta Pseudo Labels is presented in Figure 1. The main difference is that in Meta Pseudo Labels, the teacher receives feedback of the student's performance on a labeled dataset.

[1]Code is available at https://github.com/google-research/google-research/tree/master/meta_pseudo_labels.

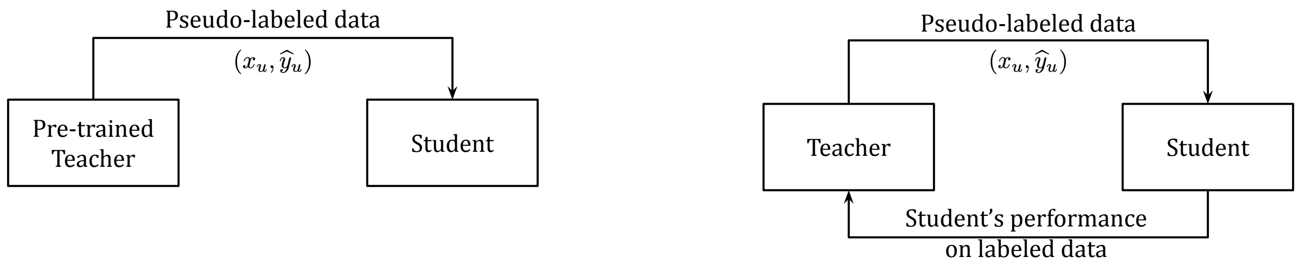

Figure 1: The difference between Pseudo Labels and Meta Pseudo Labels. **Left:** Pseudo Labels, where a fixed pre-trained teacher generates pseudo labels for the student to learn from. **Right:** Meta Pseudo Labels, where the teacher is trained along with the student. The student is trained based on the pseudo labels generated by the teacher (top arrow). The teacher is trained based on the performance of the student on labeled data (bottom arrow).

Notations. Let T and S respectively be the teacher network and the student network in Meta Pseudo Labels. Let their corresponding parameters be θ_T and θ_S. We use (x_l, y_l) to refer to a batch of images and their corresponding labels, e.g., ImageNet training images and their labels, and use x_u to refer to a batch of unlabeled images, e.g., images from the internet. We denote by $T(x_u; \theta_T)$ the *soft* predictions of the teacher network on the batch x_u of unlabeled images and likewise for the student, e.g. $S(x_l; \theta_S)$ and $S(x_u; \theta_S)$. We use $\text{CE}(q, p)$ to denote the cross-entropy loss between two distributions q and p; if q is a label then it is understood as a one-hot distribution; if q and p have multiple instances in them then $\text{CE}(q, p)$ is understood as the *average* of all instances in the batch. For example, $\text{CE}\big(y_l, S(x_l; \theta_S)\big)$ is the canonical cross-entropy loss in supervised learning.

Pseudo Labels as an optimization problem. To introduce Meta Pseudo Labels, let's first review Pseudo Labels. Specifically, Pseudo Labels (PL) trains the student model to minimize the cross-entropy loss on unlabeled data:

$$\theta_S^{\text{PL}} = \underset{\theta_S}{\text{argmin}} \; \underbrace{\mathbb{E}_{x_u}\Big[\text{CE}\big(T(x_u; \theta_T), S(x_u; \theta_S)\big)\Big]}_{:=\mathcal{L}_u\left(\theta_T, \theta_S\right)} \tag{1}$$

where the pseudo target $T(x_u; \theta_T)$ is produced by a well pre-trained teacher model with *fixed* parameter θ_T. Given a good teacher, the hope of Pseudo Labels is that the obtained θ_S^{PL} would ultimately achieve a low loss on labeled data, i.e. $\mathbb{E}_{x_l, y_l}\Big[\text{CE}\big(y_l, S(x_l; \theta_S^{\text{PL}})\big)\Big] := \mathcal{L}_l\big(\theta_S^{\text{PL}}\big)$.

Under the framework of Pseudo Labels, notice that the optimal student parameter θ_S^{PL} always depends on the teacher parameter θ_T via the pseudo targets $T(x_u; \theta_T)$. To facilitate the discussion of Meta Pseudo Labels, we can explicitly express the dependency as $\theta_S^{\text{PL}}(\theta_T)$. As an immediate observation, the ultimate student loss on labeled data $\mathcal{L}_l\big(\theta_S^{\text{PL}}(\theta_T)\big)$ is also a "function" of θ_T. Therefore, we could further optimize $\mathcal{L}_l$ with respect to θ_T:

$$\begin{aligned} \min_{\theta_T} \quad & \mathcal{L}_l\left(\theta_S^{\text{PL}}(\theta_T)\right), \\ \text{where} \quad & \theta_S^{\text{PL}}(\theta_T) = \underset{\theta_S}{\text{argmin}}\, \mathcal{L}_u\big(\theta_T, \theta_S\big). \end{aligned} \tag{2}$$

Intuitively, by optimizing the teacher's parameter according to the performance of the student on labeled data, the pseudo labels can be adjusted accordingly to further improve student's performance. As we are effectively trying to optimize the teacher on a meta level, we name our method *Meta Pseudo Labels*. However, the dependency of $\theta_S^{\text{PL}}(\theta_T)$ on θ_T is extremely complicated, as computing the gradient $\nabla_{\theta_T}\theta_S^{\text{PL}}(\theta_T)$ requires unrolling the entire student training process (i.e. argmin_{θ_S}).

Practical approximation. To make Meta Pseudo Labels feasible, we borrow ideas from previous work in meta learning [40, 15] and approximate the multi-step argmin_{θ_S} with the one-step gradient update of θ_S:

$$\theta_S^{\text{PL}}(\theta_T) \approx \theta_S - \eta_S \cdot \nabla_{\theta_S}\mathcal{L}_u\big(\theta_T, \theta_S\big),$$

where η_S is the learning rate. Plugging this approximation into the optimization problem in Equation 2 leads to the practical teacher objective in Meta Pseudo Labels:

$$\min_{\theta_T} \quad \mathcal{L}_l\Big(\theta_S - \eta_S \cdot \nabla_{\theta_S}\mathcal{L}_u\big(\theta_T, \theta_S\big)\Big). \tag{3}$$

Note that, if *soft* pseudo labels are used, i.e. $T(x_u; \theta_T)$ is the full distribution predicted by teacher, the objective above is fully differentiable with respect to θ_T and we can perform standard back-propagation to get the gradient.[2] However, in this work, we sample the *hard* pseudo labels from the teacher distribution to train the student. We use hard pseudo labels because they result in smaller computational graphs which

[2]When optimizing Equation (3), we always treat θ_S as fixed parameters and ignore its higher-order dependency on θ_T.

are necessary for our large-scale experiments in Section 4. For smaller experiments where we can use either soft pseudo labels or hard pseudo labels, we do not find significant performance difference between them. A caveat of using hard pseudo labels is that we need to rely on a slightly modified version of REINFORCE to obtain the approximated gradient of $\mathcal{L}_l$ in Equation 3 with respect to θ_T. We defer the detailed derivation to Appendix A.

On the other hand, the student's training still relies on the objective in Equation 1, except that the teacher parameter is *not fixed* anymore. Instead, θ_T is constantly changing due to the teacher's optimization. More interestingly, the student's parameter update can be reused in the one-step approximation of the teacher's objective, which naturally gives rise to an alternating optimization procedure between the student update and the teacher update:

- Student: draw a batch of unlabeled data x_u, then sample $T(x_u; \theta_T)$ from teacher's prediction, and optimize objective 1 with SGD: $\theta'_S = \theta_S - \eta_S \nabla_{\theta_S} \mathcal{L}_u(\theta_T, \theta_S)$,
- Teacher: draw a batch of labeled data (x_l, y_l), and "reuse" the student's update to optimize objective 3 with SGD: $\theta'_T = \theta_T - \eta_T \nabla_{\theta_T} \mathcal{L}_l\big(\underbrace{\theta_S - \nabla_{\theta_S} \mathcal{L}_u(\theta_T, \theta_S)}_{= \theta'_S \text{ reused from student's update}}\big)$.

Teacher's auxiliary losses. We empirically observe that Meta Pseudo Labels works well on its own. Moreover, it works even better if the teacher is jointly trained with other auxiliary objectives. Therefore, in our implementation, we augment the teacher's training with a supervised learning objective and a semi-supervised learning objective. For the supervised objective, we train the teacher on labeled data. For the semi-supervised objective, we additionally train the teacher on unlabeled data using the UDA objective [76]. For the full pseudo code of Meta Pseudo Labels when it is combined with supervised and UDA objectives for the teacher, please see Appendix B, Algorithm 1.

Finally, as the student in Meta Pseudo Labels only learns from unlabeled data with pseudo labels generated by the teacher, we can take a student model that has converged after training with Meta Pseudo Labels and finetune it on labeled data to improve its accuracy. Details of the student's finetuning are reported in our experiments.

Next, we will present the experimental results of Meta Pseudo Labels, and organize them as follows:

- Section 3 presents small scale experiments where we compare Meta Pseudo Labels against other state-of-the-art semi-supervised learning methods on widely used benchmarks.
- Section 4 presents large scale experiments of Meta Pseudo Labels where we push the limits of ImageNet accuracy.

3. Small Scale Experiments

In this section, we present our empirical studies of Meta Pseudo Labels at small scales. We first study the role of feedback in Meta Pseudo Labels on the simple TwoMoon dataset [7]. This study visually illustrates Meta Pseudo Labels' behaviors and benefits. We then compare Meta Pseudo Labels against state-of-the-art semi-supervised learning methods on standard benchmarks such as CIFAR-10-4K, SVHN-1K, and ImageNet-10%. We conclude the section with experiments on the standard ResNet-50 architecture with the full ImageNet dataset.

3.1. TwoMoon Experiment

To understand the role of feedback in Meta Pseudo Labels, we conduct an experiment on the simple and classic TwoMoon dataset [7]. The 2D nature of the TwoMoon dataset allows us to visualize how Meta Pseudo Labels behaves compared to Supervised Learning and Pseudo Labels.

Dataset. For this experiment, we generate our own version of the TwoMoon dataset. In our version, there are 2,000 examples forming two clusters each with 1,000 examples. Only 6 examples are labeled, 3 examples for each cluster, while the remaining examples are unlabeled. Semi-supervised learning algorithms are asked to use these 6 labeled examples and the clustering assumption to separate the two clusters into correct classes.

Training details. Our model architecture is a feed-forward fully-connected neural network with two hidden layers, each has 8 units. The sigmoid non-linearity is used at each layer. In Meta Pseudo Labels, both the teacher and the student share this architecture but have independent weights. All networks are trained with SGD using a constant learning rate of 0.1. The networks' weights are initialized with the uniform distribution between -0.1 and 0.1. We do not apply any regularization.

Results. We randomly generate the TwoMoon dataset for a few times and repeat the three methods: Supervised Learning, Pseudo Labels, and Meta Pseudo Labels. We observe that Meta Pseudo Labels has a much higher success rate of finding the correct classifier than Supervised Learning and Pseudo Labels. Figure 2 presents a typical outcome of our experiment, where the red and green regions correspond to the classifiers' decisions. As can be seen from the figure, Supervised Learning finds a bad classifier which classifies the labeled instances correctly but fails to take advantage of the clustering assumption to separate the two "moons". Pseudo Labels uses the bad classifier from Supervised Learning and hence receives incorrect pseudo labels on the unlabeled data. As a result, Pseudo Labels finds a classifier that misclassifies

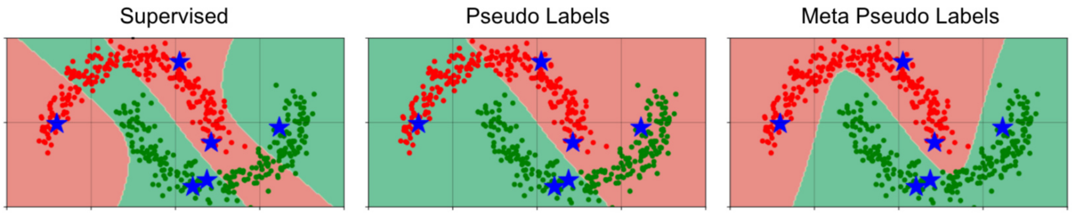

Figure 2: An illustration of the importance of feedback in Meta Pseudo Labels (right). In this example, Meta Pseudo Labels works better than Supervised Learning (left) and Pseudo Labels (middle) on the simple TwoMoon dataset. More details are in Section 3.1.

half of the data, including a few labeled instances. Meta Pseudo Labels, on the other hand, uses the feedback from the student model's loss on the labeled instances to adjust the teacher to generate better pseudo labels. As a result, Meta Pseudo Labels finds a good classifier for this dataset. In other words, Meta Pseudo Labels can address the problem of confirmation bias [2] of Pseudo Labels in this experiment.

3.2. CIFAR-10-4K, SVHN-1K, and ImageNet-10% Experiments

Datasets. We consider three standard benchmarks: CIFAR-10-4K, SVHN-1K, and ImageNet-10%, which have been widely used in the literature to fairly benchmark semi-supervised learning algorithms. These benchmarks were created by keeping a small fraction of the training set as labeled data while using the rest as unlabeled data. For CIFAR-10 [34], 4,000 labeled examples are kept as labeled data while 41,000 examples are used as unlabeled data. The test set for CIFAR-10 is standard and consists of 10,000 examples. For SVHN [46], 1,000 examples are used as labeled data whereas about 603,000 examples are used as unlabeled data. The test set for SVHN is also standard, and has 26,032 examples. Finally, for ImageNet [56], 128,000 examples are used as labeled data which is approximately 10% of the whole ImageNet training set while the rest of 1.28 million examples are used as unlabeled data. The test set for ImageNet is the standard ILSVRC 2012 version that has 50,000 examples. We use the image resolution of 32x32 for CIFAR-10 and SVHN, and 224x224 for ImageNet.

Training details. In our experiments, our teacher and our student share the same architecture but have independent weights. For CIFAR-10-4K and SVHN-1K, we use a WideResNet-28-2 [84] which has 1.45 million parameters. For ImageNet, we use a ResNet-50 [24] which has 25.5 million parameters. These architectures are also commonly used by previous works in this area. During the Meta Pseudo Labels training phase where we train both the teacher and the student, we use the default hyper-parameters from previous work for all our models, except for a few modifications in RandAugment [13] which we detail in Appendix C.2. All hyper-parameters are reported in Appendix C.4. After training both the teacher and student with Meta Pseudo Labels, we finetune the student on the labeled dataset. For this finetuning phase, we use SGD with a fixed learning rate of 10^{-5} and a batch size of 512, running for 2,000 steps for ImageNet-10% and 1,000 steps for CIFAR-10 and SVHN. Since the amount of labeled examples is limited for all three datasets, we do not use any heldout validation set. Instead, we return the model at the final checkpoint.

Baselines. To ensure a fair comparison, we only compare Meta Pseudo Labels against methods that use the same architectures and do not compare against methods that use larger architectures such as Larger-WideResNet-28-2 and PyramidNet+ShakeDrop for CIFAR-10 and SVHN [5, 4, 72, 76], or ResNet-50$\times${2,3,4}, ResNet-101, ResNet-152, etc. for ImageNet-10% [25, 23, 10, 8, 9]. We also do not compare Meta Pseudo Labels with training procedures that include self-distillation or distillation from a larger teacher [8, 9]. We enforce these restrictions on our baselines since it is known that larger architectures and distillation can improve any method, possibly including Meta Pseudo Labels.

We directly compare Meta Pseudo Labels against two baselines: Supervised Learning with full dataset and Unsupervised Data Augmentation (UDA [76]). Supervised Learning with full dataset represents the headroom because it unfairly makes use of all labeled data (e.g., for CIFAR-10, it uses all 50,000 labeled examples). We also compare against UDA because our implementation of Meta Pseudo Labels uses UDA in training the teacher. Both of these baselines use the same experimental protocols and hence ensure a fair comparison. We follow [48]'s train/eval/test splitting, and we use the same amount of resources to tune hyper-parameters for our baselines as well as for Meta Pseudo Labels. More details are in Appendix C.

Additional baselines. In addition to these two baselines, we also include a range of other semi-supervised baselines in two categories: Label Propagation and Self-Supervised. Since these methods do not share the same controlled environment, the comparison to them is not direct, and should be contextualized as suggested by [48]. More controlled experiments comparing Meta Pseudo Labels to other baselines

	Method	CIFAR-10-4K (mean ± std)	SVHN-1K (mean ± std)	ImageNet-10% Top-1	Top-5
Label Propagation Methods	Temporal Ensemble [35]	83.63 ± 0.63	92.81 ± 0.27	–	
	Mean Teacher [64]	84.13 ± 0.28	94.35 ± 0.47	–	
	VAT + EntMin [44]	86.87 ± 0.39	94.65 ± 0.19	–	83.39
	LGA + VAT [30]	87.94 ± 0.19	93.42 ± 0.36	–	
	ICT [71]	92.71 ± 0.02	96.11 ± 0.04	–	
	MixMatch [5]	93.76 ± 0.06	96.73 ± 0.31	–	
	ReMixMatch [4]	94.86 ± 0.04	97.17 ± 0.30	–	
	EnAET [72]	94.65	97.08	–	
	FixMatch [58]	95.74 ± 0.05	97.72 ± 0.38	71.5	89.1
	UDA* [76]	94.53 ± 0.18	97.11 ± 0.17	68.07	88.19
Self-Supervised Methods	SimCLR [8, 9]	–	–	71.7	90.4
	MOCOv2 [10]	–	–	71.1	–
	PCL [38]	–	–	–	85.6
	PIRL [43]	–	–	–	84.9
	BYOL [21]	–	–	68.8	89.0
	Meta Pseudo Labels	$\mathbf{96.11 \pm 0.07}$	$\mathbf{98.01 \pm 0.07}$	**73.89**	**91.38**
	Supervised Learning with full dataset*	94.92 ± 0.17	97.41 ± 0.16	76.89	93.27

Table 2: Image classification accuracy on CIFAR-10-4K, SVHN-1K, and ImageNet-10%. Higher is better. For CIFAR-10-4K and SVHN-1K, we report mean ± std over 10 runs, while for ImageNet-10%, we report Top-1/Top-5 accuracy of a single run. For fair comparison, we only include results that share the same model architecture: WideResNet-28-2 for CIFAR-10-4K and SVHN-1K, and ResNet-50 for ImageNet-10%. * indicates our implementation which uses the same experimental protocols. Except for UDA, results in the first two blocks are from representative important papers, and hence do not share the same controlled environment with ours.

are presented in Appendix D.

Results. Table 2 presents our results with Meta Pseudo Labels in comparison with other methods. The results show that under strictly fair comparisons (as argued by [48]), Meta Pseudo Labels significantly improves over UDA. Interestingly, on CIFAR-10-4K, Meta Pseudo Labels even exceeds the headroom supervised learning on full dataset. On ImageNet-10%, Meta Pseudo Labels outperforms the UDA teacher by more than 5% in top-1 accuracy, going from 68.07% to 73.89%. For ImageNet, such relative improvement is very significant.

Comparing to existing state-of-the-art methods. Compared to results reported from past papers, Meta Pseudo Labels has achieved the best accuracies *among the same model architectures* on all the three datasets: CIFAR-10-4K, SVHN-1K, and ImageNet-10%. On CIFAR-10-4K and SVHN-1K, Meta Pseudo Labels leads to almost 10% relative error reduction compared to the highest reported baselines [58]. On ImageNet-10%, Meta Pseudo Labels outperforms SimCLR [8, 9] by 2.19% top-1 accuracy.

While better results on these datasets exist, to our knowledge, such results are all obtained with larger models, stronger regularization techniques, or extra distillation procedures. For example, the best reported accuracy on CIFAR-10-4K is 97.3% [76] but this accuracy is achieved with a PyramidNet which has 17x more parameters than our WideResNet-28-2 and uses the complex ShakeDrop regularization [80]. On the other hand, the best reported top-1 accuracy for ImageNet-10% is 80.9%, achieved by SimCLRv2 [9] using a self-distillation training phase and a ResNet-152×3 which has 32x more parameters than our ResNet-50. Such enhancements on architectures, regularization, and distillation can also be applied to Meta Pseudo Labels to further improve our results.

3.3. ResNet-50 Experiment

The previous experiments show that Meta Pseudo Labels outperforms other semi-supervised learning methods on CIFAR-10-4K, SVHN-1K, and ImageNet-10%. In this experiment, we benchmark Meta Pseudo Labels on the entire ImageNet dataset plus unlabeled images from the JFT dataset. The purpose of this experiment is to verify if Meta Pseudo Labels works well on the widely used ResNet-50 architecture [24] before we conduct more large scale experiments on EfficientNet (Section 4).

Datasets. As mentioned, we experiment with all labeled examples from the ImageNet dataset. We reserve 25,000 examples from the ImageNet dataset for hyper-parameter tuning and model selection. Our test set is the ILSVRC 2012 validation set. Additionally, we take 12.8 million unlabeled images from the JFT dataset. To obtain these 12.8 million unlabeled images, we first train a ResNet-50 on the entire ImageNet training set and then use the resulting ResNet-50 to assign class probabilities to images in the JFT dataset. We then select 12,800 images of highest probability for each of the 1,000 classes of ImageNet. This selection results in 12.8 million images. We also make sure that none of the 12.8 million images that we use overlaps with the ILSVRC 2012 validation set of ImageNet. This procedure of filtering extra unlabeled data has been used by UDA [76] and Noisy Student [77].

Implementation details. We implement Meta Pseudo Labels the same as in Section 3.2 but we use a larger batch size and more training steps, as the datasets are much larger for this experiment. Specifically, for both the student and the teacher, we use the batch size of 4,096 for labeled images and the batch size of 32,768 for unlabeled images. We train for 500,000 steps which equals to about 160 epochs on the unlabeled dataset. After training the Meta Pseudo Labels phase on ImageNet+JFT, we finetune the resulting student on ImageNet for 10,000 SGD steps, using a fixed learning rate of 10^{-4}. Using 512 TPUv2 cores, our training procedure takes about 2 days.

Method	Unlabeled Images	Accuracy (top-1/top-5)
Supervised [24]	None	76.9/93.3
AutoAugment [12]	None	77.6/93.8
DropBlock [18]	None	78.4/94.2
FixRes [68]	None	79.1/94.6
FixRes+CutMix [83]	None	79.8/94.9
NoisyStudent [77]	JFT	78.9/94.3
UDA [76]	JFT	79.0/94.5
Billion-scale SSL [68, 79]	YFCC	82.5/**96.6**
Meta Pseudo Labels	JFT	**83.2**/96.5

Table 3: Top-1 and Top-5 accuracy of Meta Pseudo Labels and other representative supervised and semi-supervised methods on ImageNet with ResNet-50.

Baselines. We compare Meta Pseudo Labels against two groups of baselines. The first group contains supervised learning methods with data augmentation or regularization methods such as AutoAugment [12], DropBlock[18], and CutMix [83]. These baselines represent state-of-the-art supervised learning methods on ResNet-50. The second group of baselines consists of three recent semi-supervised learning methods that leverage the labeled training images from ImageNet and unlabeled images elsewhere. Specifically, billion-scale semi-supervised learning [79] uses unlabeled data from the YFCC100M dataset [65], while UDA [76] and Noisy Student [77] both use JFT as unlabeled data like Meta Pseudo Labels. Similar to Section 3.2, we only compare Meta Pseudo Labels to results that are obtained with ResNet-50 and without distillation.

Results. Table 3 presents the results. As can be seen from the table, Meta Pseudo Labels boosts the top-1 accuracy of ResNet-50 from 76.9% to 83.2%, which is a large margin of improvement for ImageNet, outperforming both UDA and Noisy Student. Meta Pseudo Labels also outperforms Billion-scale SSL [68, 79] in top-1 accuracy. This is particularly impressive since Billion-scale SSL pre-trains their ResNet-50 on weakly-supervised images from Instagram.

4. Large Scale Experiment: Pushing the Limits of ImageNet Accuracy

In this section, we scale up Meta Pseudo Labels to train on a large model and a large dataset to push the limits of ImageNet accuracy. Specifically, we use the EfficientNet-L2 architecture because it has a higher capacity than ResNets. EfficientNet-L2 was also used by Noisy Student [77] to achieve the top-1 accuracy of 88.4% on ImageNet.

Datasets. For this experiment, we use the entire ImageNet training set as labeled data, and use the JFT dataset as unlabeled data. The JFT dataset has 300 million images, and then is filtered down to 130 million images by Noisy Student using confidence thresholds and up-sampling [77]. We use the same 130 million images as Noisy Student.

Model architecture. We experiment with EfficientNet-L2 since it has the state-of-the-art performance on ImageNet [77] without extra labeled data. We use the same hyper-parameters with Noisy Student, except that we use the training image resolution of 512x512 instead of 475x475. We increase the input image resolution to be compatible with our model parallelism implementation which we discuss in the next paragraph. In addition to EfficientNet-L2, we also experiment with a smaller model, which has the same depth with EfficientNet-B6 [63] but has the width factor increased from 2.1 to 5.0. This model, termed EfficientNet-B6-Wide, has 390 million parameters. We adopt all hyper-parameters of EfficientNet-L2 for EfficientNet-B6-Wide. We find that EfficientNet-B6-Wide has almost the same performance with EfficientNet-L2, but is faster to compile and train.

Model parallelism. Due to the memory footprint of our networks, keeping two networks in memory for the teacher and the student vastly exceeds the available memory of our accelerators. We thus design a hybrid model-data parallelism framework to run Meta Pseudo Labels. Specifically, we use a cluster of 2,048 TPUv3 cores. We divide these cores into 128 identical replicas to run with standard data parallelism with

Method	# Params	Extra Data	ImageNet Top-1	ImageNet Top-5	ImageNet-ReaL [6] Precision@1
ResNet-50 [24]	26M	–	76.0	93.0	82.94
ResNet-152 [24]	60M	–	77.8	93.8	84.79
DenseNet-264 [28]	34M	–	77.9	93.9	–
Inception-v3 [62]	24M	–	78.8	94.4	83.58
Xception [11]	23M	–	79.0	94.5	–
Inception-v4 [61]	48M	–	80.0	95.0	–
Inception-resnet-v2 [61]	56M	–	80.1	95.1	–
ResNeXt-101 [78]	84M	–	80.9	95.6	85.18
PolyNet [87]	92M	–	81.3	95.8	–
SENet [27]	146M	–	82.7	96.2	–
NASNet-A [90]	89M	–	82.7	96.2	82.56
AmoebaNet-A [52]	87M	–	82.8	96.1	–
PNASNet [39]	86M	–	82.9	96.2	–
AmoebaNet-C + AutoAugment [12]	155M	–	83.5	96.5	–
GPipe [29]	557M	–	84.3	97.0	–
EfficientNet-B7 [63]	66M	–	85.0	97.2	–
EfficientNet-B7 + FixRes [70]	66M	–	85.3	97.4	–
EfficientNet-L2 [63]	480M	–	85.5	97.5	–
ResNet-50 Billion-scale SSL [79]	26M	3.5B labeled Instagram	81.2	96.0	–
ResNeXt-101 Billion-scale SSL [79]	193M	3.5B labeled Instagram	84.8	–	–
ResNeXt-101 WSL [42]	829M	3.5B labeled Instagram	85.4	97.6	88.19
FixRes ResNeXt-101 WSL [69]	829M	3.5B labeled Instagram	86.4	98.0	89.73
Big Transfer (BiT-L) [33]	928M	300M labeled JFT	87.5	98.5	90.54
Noisy Student (EfficientNet-L2) [77]	480M	300M unlabeled JFT	88.4	98.7	90.55
Noisy Student + FixRes [70]	480M	300M unlabeled JFT	88.5	98.7	–
Vision Transformer (ViT-H) [14]	632M	300M labeled JFT	88.55	–	90.72
EfficientNet-L2-NoisyStudent + SAM [16]	480M	300M unlabeled JFT	88.6	98.6	–
Meta Pseudo Labels (EfficientNet-B6-Wide)	390M	300M unlabeled JFT	90.0	98.7	**91.12**
Meta Pseudo Labels (EfficientNet-L2)	480M	300M unlabeled JFT	**90.2**	**98.8**	91.02

Table 4: Top-1 and Top-5 accuracy of Meta Pseudo Labels and previous state-of-the-art methods on ImageNet. With EfficientNet-L2 and EfficientNet-B6-Wide, Meta Pseudo Labels achieves an improvement of 1.6% on top of the state-of-the-art [16], despite the fact that the latter uses 300 million *labeled* training examples from JFT.

synchronized gradients. Within each replica, which runs on 2,048/128=16 cores, we implement two types of model parallelism. First, each input image of resolution 512x512 is split along the width dimension into 16 patches of equal size 512x32 and is distributed to 16 cores to process. Note that we choose the input resolution of 512x512 because 512 is close to the resolution 475x475 used by Noisy Student and 512 keeps the dimensions of the network's intermediate outputs divisible by 16. Second, each weight tensor is also split equally into 16 parts that are assigned to the 16 cores. We implement our hybrid data-model parallelism in the XLA-Sharding framework [37]. With this parallelism, we can fit a batch size of 2,048 labeled images and 16,384 unlabeled images into each training step. We train the model for 1 million steps in total, which takes about 11 days for EfficientNet-L2 and 10 days for EfficientNet-B6-Wide. After finishing the Meta Pseudo Labels training phase, we finetune the models on our labeled dataset for 20,000 steps. Details of the finetuning procedures are in Appendix C.4.

Results. Our results are presented in Table 4. From the table, it can be seen that Meta Pseudo Labels achieves 90.2% top-1 accuracy on ImageNet, which is a new state-of-the-art on this dataset. This result is 1.8% better than the same EfficientNet-L2 architecture trained with Noisy Student [77] and FixRes [69, 70]. Meta Pseudo Labels also outperforms the recent results by BiT-L [33] and the previous state-of-the-art by Vision Transformer [14]. The important contrast here is that both Bit-L and Vision Transformer pre-train on 300 million *labeled* images from JFT, while our method only uses *unlabeled* images from this dataset. At this level of accuracy, our gain of 1.6% over [16] is a very significant margin of improvement compared to recent gains. For instance, the gain of Vision Transformer [14] over Noisy Student + FixRes was only 0.05%, and the gain of FixRes over Noisy Student was only 0.1%.

Finally, to verify that our model does not simply overfit to the ImageNet ILSVRC 2012 validation set, we test it on the ImageNet-ReaL test set [6]. On this test set, our model also works well and achieves 91.02% Precision@1 which is

0.4% better than Vision Transformer [14]. This gap is also bigger than the gap between Vision Transformer and Noisy Student which is only 0.17%.

A lite version of Meta Pseudo Labels. Given the expensive training cost of Meta Pseudo Labels, we design a lite version of Meta Pseudo Labels, termed *Reduced Meta Pseudo Labels*. We describe this lite version in Appendix E, where we achieve 86.9% top-1 accuracy on the ImageNet ILSRVC 2012 validation set with EfficentNet-B7. To avoid using proprietary data like JFT, we use the ImageNet training set as labeled data and the YFCC100M dataset [65] as unlabeled data. Reduced Meta Pseudo Labels allows us to implement the feedback mechanism of Meta Pseudo Labels while avoiding the need to keep two networks in memory.

5. Related Works

Pseudo Labels. The method of Pseudo Labels, also known as self-training, is a simple Semi-Supervised Learning (SSL) approach that has been successfully applied to improve the state-of-the-art of many tasks, such as: image classification [79, 77], object detection, semantic segmentation [89], machine translation [22], and speech recognition [31, 49]. Vanilla Pseudo Labels methods keep a pre-trained teacher fixed during the student's learning, leading to a confirmation bias [2] when the pseudo labels are inaccurate. Unlike vanilla Pseudo Labels, Meta Pseudo Labels continues to adapt the teacher to improve the student's performance on a labeled dataset. This extra adaptation allows the teacher to generate better pseudo labels to teach the student as shown in our experiments.

Other SSL approaches. Other typical SSL methods often train a single model by optimizing an objective function that combines a supervised loss on labeled data and an unsupervised loss on unlabeled data. The supervised loss is often the cross-entropy computed on the labeled data. Meanwhile, the unsupervised loss is typically either a self-supervised loss or a label propagation loss. Self-supervised losses typically encourage the model to develop a common sense about images, such as in-painting [50], solving jigsaw puzzles [47], predicting the rotation angle [19], contrastive prediction [25, 10, 8, 9, 38], or bootstraping the latent space [21]. On the other hand, label propagation losses typically enforce that the model is invariant against certain transformations of the data such as data augmentations, adversarial attacks, or proximity in the latent space [35, 64, 44, 5, 76, 30, 71, 58, 32, 51, 20]. Meta Pseudo Labels is distinct from the aforementioned SSL methods in two notable ways. First, the student in Meta Pseudo Labels never learns directly from labeled data, which helps to avoid overfitting, especially when labeled data is limited. Second, the signal that the teacher in Meta Pseudo Labels receives from the student's performance on labeled data is a novel way of utilizing labeled data.

Knowledge Distillation and Label Smoothing. The teacher in Meta Pseudo Labels uses its softmax predictions on unlabeled data to teach the student. These softmax predictions are generally called the soft labels, which have been widely utilized in the literature on knowledge distillation [26, 17, 86]. Outside the line of work on distillation, manually designed soft labels, such as label smoothing [45] and temperature sharpening or dampening [76, 77], have also been shown to improve models' generalization. Both of these methods can be seen as adjusting the labels of the training examples to improve optimization and generalization. Similar to other SSL methods, these adjustments do not receive any feedback from the student's performance as proposed in this paper. An experiment comparing Meta Pseudo Labels to Label Smoothing is presented in Appendix D.2.

Bi-level optimization algorithms. We use *Meta* in our method name because our technique of deriving the teacher's update rule from the student's feedback is based on a bi-level optimization problem which appears frequently in the literature of meta-learning. Similar bi-level optimization problems have been proposed to optimize a model's learning process, such as learning the learning rate schedule [3], designing architectures [40], correcting wrong training labels [88], generating training examples [59], and re-weighting training data [73, 74, 54, 53]. Meta Pseudo Labels uses the same bi-level optimization technique in this line of work to derive the teacher's gradient from the student's feedback. The difference between Meta Pseudo Labels and these methods is that Meta Pseudo Labels applies the bi-level optimization technique to improve the pseudo labels generated by the teacher model.

6. Conclusion

In this paper, we proposed the Meta Pseudo Labels method for semi-supervised learning. Key to Meta Pseudo Labels is the idea that the teacher learns from the student's feedback to generate pseudo labels in a way that best helps student's learning. The learning process in Meta Pseudo Labels consists of two main updates: updating the student based on the pseudo labeled data produced by the teacher and updating the teacher based on the student's performance. Experiments on standard low-resource benchmarks such as CIFAR-10-4K, SVHN-1K, and ImageNet-10% show that Meta Pseudo Labels is better than many existing semi-supervised learning methods. Meta Pseudo Labels also scales well to large problems, attaining 90.2% top-1 accuracy on ImageNet, which is 1.6% better than the previous state-of-the-art [16]. The consistent gains confirm the benefit of the student's feedback to the teacher.

References

[1] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, Manjunath Kudlur, Josh Levenberg, Rajat Monga, Sherry Moore, Benoit G, Derek . Murrayand Steiner, Paul Tucker, Vijay Vasudevan, Pete Warden, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. Tensorflow: A system for large-scale machine learning. In *USENIX Symposium on Operating Systems Design and Implementation*, 2016. 17

[2] Eric Arazo, Diego Ortego, Paul Albert, Noel E. O'Connor, and Kevin McGuinness. Pseudo-labeling and confirmation bias in deep semi-supervised learning. *Arxiv, 1908.02983*, 2019. 1, 4, 8, 19

[3] Atilim Gunes Baydin, Robert Cornish, David Martinez Rubio, Mark Schmidt, and Frank Wood. Online learning rate adaptation with hypergradient descent. In *International Conference on Learning Representations*, 2018. 8, 16

[4] David Berthelot, Nicholas Carlini, Ekin D. Cubuk, Alex Kurakin, Kihyuk Sohn, Han Zhang, and Colin Raffel. Remixmatch: Semi-supervised learning with distribution alignment and augmentation anchoring. In *International Conference on Learning Representations*, 2020. 4, 5

[5] David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin Raffel. MixMatch: A holistic approach to semi-supervised learning. In *Advances in Neural Information Processing Systems*, 2019. 4, 5, 8

[6] Lucas Beyer, Olivier J Hénaff, Alexander Kolesnikov, Xiaohua Zhai, and Aäron van den Oord. Are we done with ImageNet? *arXiv preprint arXiv:2006.07159*, 2020. 1, 7, 8

[7] Olivier Chapelle, Bernhard Schlkopf, and Alexander Zien. *Semi-Supervised Learning*. The MIT Press, 2010. 3

[8] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning*, 2020. 4, 5, 8

[9] Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey Hinton. Big self-supervised models are strong semi-supervised learners. In *Advances in Neural Information Processing Systems*, 2020. 4, 5, 8

[10] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *Arxiv, 2003.04297*, 2020. 4, 5, 8

[11] Francois Chollet. Xception: Deep learning with depthwise separable convolutions. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 7

[12] Ekin D. Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V. Le. AutoAugment: Learning augmentation policies from data. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 6, 7, 16, 17

[13] Ekin D. Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V. Le. Randaugment: Practical data augmentation with no separate search. In *Advances in Neural Information Processing Systems*, 2020. 4, 16

[14] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *Arxiv, 2010.11929*, 2020. 1, 7, 8

[15] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, 2017. 2

[16] Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. *Arxiv, 2010.01412*, 2020. 1, 7, 8

[17] Tommaso Furlanello, Zachary C. Lipton, Michael Tschannen, Laurent Itti, and Anima Anandkumar. Born again neural networks. In *International Conference on Machine Learning*, 2018. 8

[18] Golnaz Ghiasi, Tsung-Yi Lin, and Quoc V. Le. Dropblock: A regularization method for convolutional networks. In *Advances in Neural Information Processing Systems*, 2018. 6

[19] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 8

[20] Yves Grandvalet and Yoshua Bengio. Semi-supervised learning by entropy minimization. In *International Conference on Computer Vision*, 2005. 8

[21] 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, Remi Munos, and Michal Valko. Bootstrap your own latent: A new approach to self-supervised learning. In *Advances in Neural Information Processing Systems*, 2020. 5, 8

[22] Junxian He, Jiatao Gu, Jiajun Shen, and Marc'Aurelio Ranzato. Revisiting self-training for neural sequence generation. In *International Conference on Learning Representations*, 2020. 8

[23] Kaiming He, Haoqi Fan, Yuxin Wu, Saining He, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 4

[24] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 4, 5, 6, 7

[25] Olivier J. Henaff, Aravind Srinivas, Jeffrey De Fauw, Ali Razavi, Carl Doersch, S. M. Ali Eslami, and Aaron van den Oord. Data-efficient image recognition with contrastive predictive coding. *Arxiv, 2003.04297*, 2020. 4, 8

[26] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *Arxiv, 1503.02531*, 2015. 1, 8

[27] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018. 7

[28] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 7

[29] Yanping Huang, Yonglong Cheng, Dehao Chen, HyoukJoong Lee, Jiquan Ngiam, Quoc V Le, and Zhifeng Chen. GPipe: Efficient training of giant neural networks using pipeline parallelism. In *Advances in Neural Information Processing Systems*, 2019. 7

[30] Jacob Jackson and John Schulman. Semi-supervised learning by label gradient alignment. *Arxiv 1902.02336*, 2019. 5, 8

[31] Jacob Kahn, Ann Lee, and Awni Hannun. Self-training for end-to-end speech recognition. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2020. 8

[32] Zhanghan Ke, Daoye Wang, Qiong Yan, Jimmy Ren, and Rynson W. H. Lau. Dual student: Breaking the limits of the teacher in semi-supervised learning. In *International Conference in Computer Vision*, 2019. 8

[33] Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Joan Puigcerver, Jessica Yung, Sylvain Gelly, and Neil Houlsby. Big transfer (bit): General visual representation learning. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020. 7

[34] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009. 4

[35] Samuli Laine and Timo Aila. Temporal ensembling for semi-supervised learning. In *International Conference on Learning Representations*, 2017. 5, 8

[36] Dong-Hyun Lee. Pseudo-Label: The simple and efficient semi-supervised learning method for deep neural networks. In *International Conference on Machine Learning Workshop*, 2013. 1, 20

[37] Dmitry Lepikhin, HyoukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam Shazeer, and Zhifeng Chen. Gshard: Scaling giant models with conditional computation and automatic sharding. *Arxiv, 2006.16668*, 2020. 7

[38] Junnan Li, Pan Zhou, Caiming Xiong, Richard Socher, and Steven CH Hoi. Prototypical contrastive learning of unsupervised representations. *Arxiv, 2005.04966*, 2020. 5, 8

[39] Chenxi Liu, Barret Zoph, Maxim Neumann, Jonathon Shlens, Wei Hua, Li-Jia Li, Li Fei-Fei, Alan Yuille, Jonathan Huang, and Kevin Murphy. Progressive neural architecture search. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018. 7

[40] Hanxiao Liu, Karen Simonyan, and Yiming Yang. Darts: Differentiable architecture search. In *International Conference on Learning Representations*, 2019. 2, 8

[41] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. In *International Conference on Learning Representations*, 2017. 17

[42] 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. *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018. 7

[43] Ishan Misra and Laurens van der Maaten. Self-supervised learning of pretext-invariant representations. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 5

[44] Takeru Miyato, Shin-ichi Maeda, Shin Ishii, and Masanori Koyama. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018. 5, 8

[45] Rafael Müller, Simon Kornblith, and Geoffrey Hinton. When does label smoothing help? In *Advances in Neural Information Processing Systems*, 2019. 8

[46] Yuval Netzer, Tao Wang, Alessandro Coates, Adamand Bissacco, Bo Wu, and Andrew Y. Ng. Reading digits in natural images with unsupervised feature learning. In *Advances in Neural Information Processing Systems Workshop on Deep Learning and Unsupervised Feature Learning*, 2011. 4

[47] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 8

[48] Avital Oliver, Augustus Odena, Colin Raffel, Ekin D. Cubuk, and Ian J. Goodfellow. Realistic evaluation of deep semi-supervised learning algorithms. In *Advances in Neural Information Processing Systems*, 2018. 4, 5, 17

[49] Daniel S. Park, Yu Zhang, Ye Jia, Wei Han, Chung-Cheng Chiu, Bo Li, Yonghui Wu, and Quoc V. Le. Improved noisy student training for automatic speech recognition. In *Interspeech*, 2020. 8

[50] Deepak Pathak, Philipp Krahenbühl, Jeff Donahue, Trevor Darrell, and Alexei A. Efrös. Context encoders: Feature learning by inpainting. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 8

[51] Ilija Radosavovic, Piotr Dollár, Ross Girshick, Georgia Gkioxari, and Kaiming He. Data distillation: Towards omni-supervised learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 1, 8

[52] Esteban Real, Alok Aggarwal, Yanping Huang, and Quoc V Le. Regularized evolution for image classifier architecture search. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, 2019. 7

[53] Mengye Ren, Wenyuan Zeng, Bin Yang, and Raquel Urtasun. Learning to reweight examples for robust deep learning. In *International Conference on Machine Learning*, 2018. 8

[54] Zhongzheng Ren, Raymond A. Yeh, and Alexander G. Schwing. Not all unlabeled data are equal: Learning to weight data in semi-supervised learning. 2020. 8

[55] Ellen Riloff. Automatically generating extraction patterns from untagged text. In *Proceedings of the national conference on artificial intelligence*, 1996. 1

[56] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*, 2015. 1, 4

[57] H Scudder. Probability of error of some adaptive pattern-recognition machines. *IEEE Transactions on Information Theory*, 11(3), 1965. 1

[58] Kihyuk Sohn, David Berthelot, Zizhao Li, Chun-Liang Zhang, Nicholas Carlini, Ekin D. Cubuk, Alex Kurakin, Han Zhang, and Colin Raffel. Fixmatch: Simplifying semi-supervised

learning with consistency and confidence. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 1, 5, 8

[59] Felipe Petroski Such, Aditya Rawal, Joel Lehman, Kenneth O. Stanley, and Jeff Clune. Generative teaching networks: Accelerating neural architecture search by learning to generate synthetic training data. 2020. 8

[60] 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*, 2017. 1

[61] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017. 7

[62] 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*, 2016. 7

[63] Mingxing Tan and Quoc V. Le. EfficientNet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*, 2019. 6, 7

[64] 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*, 2017. 5, 8

[65] Bart Thomee, David A. Shamma, Gerald Friedland, Benjamin Elizalde, Karl Ni, Douglas Poland, Damian Borth, and Li-Jia Li. YFCC100M: The new data in multimedia research. *Communications of the ACM*, 2016. 6, 8, 21

[66] T. Tieleman and G. Hinton. RmsProp: Divide the gradient by a running average of its recent magnitude. COURSERA: Neural Networks for Machine Learning, 2012. 17

[67] Antonio Torralba, Rob Fergus, and William T. Freeman. 80 million tiny images: a large dataset for non-parametric object and scene recognition. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2008. 21

[68] Hugo Touvron, Andrea Vedaldi, Matthijs Douze, and Herve Jegou. Fixing the train-test resolution discrepancy. In *Advances in Neural Information Processing Systems*, 2019. 6

[69] Hugo Touvron, Andrea Vedaldi, Matthijs Douze, and Hervé Jégou. Fixing the train-test resolution discrepancy. In *Advances in Neural Information Processing Systems*, 2019. 7

[70] Hugo Touvron, Andrea Vedaldi, Matthijs Douze, and Hervé Jégou. Fixing the train-test resolution discrepancy: Fixefficientnet. *arXiv preprint arXiv:2003.08237*, 2020. 7

[71] Vikas Verma, Alex Lamb, Juho Kannala, Yoshua Bengio, and David Lopez-Paz. Interpolation consistency training for semi-supervised learning. In *International Joint Conference on Artificial Intelligence*, 2019. 5, 8

[72] Xiao Wang, Daisuke Kihara, Jiebo Luo, and Guo-Jun Qi. Enaet: Self-trained ensemble autoencoding transformations for semi-supervised learning. *Arxiv 1911.09265*, 2019. 4, 5

[73] Xinyi Wang, Hieu Pham, Paul Mitchel, Antonis Anastasopoulos, Jaime Carbonell, and Graham Neubig. Optimizing data usage via differentiable rewards. In *International Conference on Machine Learning*, 2020. 8

[74] Yulin Wang, Jiayi Guo, Shiji Song, and Gao Huang. Meta-semi: A meta-learning approach for semi-supervised learning. *Arxiv, 2007.02394*, 2020. 8

[75] Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 1992. 14

[76] Qizhe Xie, Zihang Dai, Eduard Hovy, Minh-Thang Luong, and Quoc V. Le. Unsupervised data augmentation for consistency training. In *Advances in Neural Information Processing Systems*, 2020. 1, 3, 4, 5, 6, 8, 15, 17, 18

[77] Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V Le. Self-training with noisy student improves imagenet classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020. 1, 6, 7, 8, 17, 22

[78] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 7

[79] I. Zeki Yalniz, Herv'e J'egou, Kan Chen, Manohar Paluri, and Dhruv Mahajan. Billion-scale semi-supervised learning for image classification. *Arxiv 1905.00546*, 2019. 1, 6, 7, 8

[80] Yoshihiro Yamada, Masakazu Iwamura, Takuya Akiba, and Koichi Kise. Shakedrop regularization for deep residual learning. *Arxiv, 1802.0237*, 2018. 5

[81] David Yarowsky. Unsupervised word sense disambiguation rivaling supervised methods. In *33rd annual meeting of the association for computational linguistics*, 1995. 1

[82] Yang You, Igor Gitman, and Boris Ginsburg. Large batch training of convolutional networks. *Arxiv, 1708.03888*, 2017. 17

[83] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. CutMix: Regularization strategy to train strong classifiers with localizable features. In *International Conference on Computer Vision*, 2019. 6

[84] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In *British Machine Vision Conference*, 2016. 4

[85] Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*, 2018. 20

[86] Linfeng Zhang, Jiebo Song, Anni Gao, Jingwei Chen, Chenglong Bao, and Kaisheng Ma. Be your own teacher: Improve the performance of convolutional neural networks via self distillation. In *International Conference on Computer Vision*, 2019. 8

[87] Xingcheng Zhang, Zhizhong Li, Chen Change Loy, and Dahua Lin. Polynet: A pursuit of structural diversity in very deep networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 7

[88] Guoqing Zheng, Ahmed Hassan Awadallah, and Susan Dumais. Meta label correction for learning with weak supervision. *Arxiv, 1911.03809*, 2019. 8

[89] Barret Zoph, Golnaz Ghiasi, Tsung-Yi Lin, Yin Cui, Hanxiao Liu, Ekin D Cubuk, and Quoc V Le. Rethinking pre-training and self-training. In *Advances in Neural Information Processing Systems*, 2020. 1, 8

[90] Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc V Le. Learning transferable architectures for scalable image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018. 7