Self-training with Noisy Student improves ImageNet classification

Qizhe Xie* [1], Minh-Thang Luong[1], Eduard Hovy[2], Quoc V. Le[1]
[1]Google Research, Brain Team, [2]Carnegie Mellon University
{qizhex, thangluong, qvl}@google.com, hovy@cmu.edu

Abstract

We present Noisy Student Training, a semi-supervised learning approach that works well even when labeled data is abundant. Noisy Student Training achieves 88.4% top-1 accuracy on ImageNet, which is 2.0% better than the state-of-the-art model that requires 3.5B weakly labeled Instagram images. On robustness test sets, it improves ImageNet-A top-1 accuracy from 61.0% to 83.7%, reduces ImageNet-C mean corruption error from 45.7 to 28.3, and reduces ImageNet-P mean flip rate from 27.8 to 12.2.

Noisy Student Training extends the idea of self-training and distillation with the use of equal-or-larger student models and noise added to the student during learning. On ImageNet, we first train an EfficientNet model on labeled images and use it as a teacher to generate pseudo labels for 300M unlabeled images. We then train a larger EfficientNet as a student model on the combination of labeled and pseudo labeled images. We iterate this process by putting back the student as the teacher. During the learning of the student, we inject noise such as dropout, stochastic depth, and data augmentation via RandAugment to the student so that the student generalizes better than the teacher.[1]

1. Introduction

Deep learning has shown remarkable successes in image recognition in recent years [45, 80, 75, 30, 83]. However state-of-the-art (SOTA) vision models are still trained with supervised learning which requires a large corpus of labeled images to work well. By showing the models only labeled images, we limit ourselves from making use of unlabeled images available in much larger quantities to improve accuracy and robustness of SOTA models.

Here, we use unlabeled images to improve the SOTA ImageNet accuracy and show that the accuracy gain has an outsized impact on robustness (out-of-distribution generalization). For this purpose, we use a much larger corpus of unlabeled images, where a large fraction of images do not belong to ImageNet training set distribution (i.e., they do not belong to any category in ImageNet). We train our model with Noisy Student Training, a semi-supervised learning approach, which has three main steps: (1) train a teacher model on labeled images, (2) use the teacher to generate pseudo labels on unlabeled images, and (3) train a student model on the combination of labeled images and pseudo labeled images. We iterate this algorithm a few times by treating the student as a teacher to relabel the unlabeled data and training a new student.

Noisy Student Training improves self-training and distillation in two ways. First, it makes the student larger than, or at least equal to, the teacher so the student can better learn from a larger dataset. Second, it adds noise to the student so the noised student is forced to learn harder from the pseudo labels. To noise the student, we use input noise such as RandAugment data augmentation [18] and model noise such as dropout [76] and stochastic depth [37] during training.

Using Noisy Student Training, together with 300M unlabeled images, we improve EfficientNet's [83] ImageNet top-1 accuracy to 88.4%. This accuracy is 2.0% better than the previous SOTA results which requires 3.5B weakly labeled Instagram images. Not only our method improves standard ImageNet accuracy, it also improves classification robustness on much harder test sets by large margins: ImageNet-A [32] top-1 accuracy from 61.0% to 83.7%, ImageNet-C [31] mean corruption error (mCE) from 45.7 to 28.3 and ImageNet-P [31] mean flip rate (mFR) from 27.8 to 12.2. Our main results are shown in Table 1.

	ImageNet top-1 acc.	ImageNet-A top-1 acc.	ImageNet-C mCE	ImageNet-P mFR
Prev. SOTA	86.4%	61.0%	45.7	27.8
Ours	**88.4%**	**83.7%**	**28.3**	**12.2**

Table 1: Summary of key results compared to previous state-of-the-art models [86, 55]. Lower is better for mean corruption error (mCE) and mean flip rate (mFR).

* This work was conducted at Google.

[1]Models are available at https://github.com/tensorflow/tpu/tree/master/models/official/efficientnet. Code is available at https://github.com/google-research/noisystudent.

2. Noisy Student Training

Algorithm 1 gives an overview of Noisy Student Training. The inputs to the algorithm are both labeled and unlabeled images. We use the labeled images to train a teacher model using the standard cross entropy loss. We then use the teacher model to generate pseudo labels on unlabeled images. The pseudo labels can be soft (a continuous distribution) or hard (a one-hot distribution). We then train a student model which minimizes the combined cross entropy loss on both labeled images and unlabeled images. Finally, we iterate the process by putting back the student as a teacher to generate new pseudo labels and train a new student. The algorithm is also illustrated in Figure 1.

Require: Labeled images $\{(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)\}$ and unlabeled images $\{\tilde{x}_1, \tilde{x}_2, ..., \tilde{x}_m\}$.

1: Learn teacher model θ_*^t which minimizes the cross entropy loss on labeled images

$$\frac{1}{n}\sum_{i=1}^{n} \ell(y_i, f^{noised}(x_i, \theta^t))$$

2: Use a normal (i.e., not noised) teacher model to generate soft or hard pseudo labels for clean (i.e., not distorted) unlabeled images

$$\tilde{y}_i = f(\tilde{x}_i, \theta_*^t), \forall i = 1, \cdots, m$$

3: Learn an **equal-or-larger** student model θ_*^s which minimizes the cross entropy loss on labeled images and unlabeled images with **noise** added to the student model

$$\frac{1}{n}\sum_{i=1}^{n} \ell(y_i, f^{noised}(x_i, \theta^s)) + \frac{1}{m}\sum_{i=1}^{m} \ell(\tilde{y}_i, f^{noised}(\tilde{x}_i, \theta^s))$$

4: Iterative training: Use the student as a teacher and go back to step 2.

Algorithm 1: Noisy Student Training.

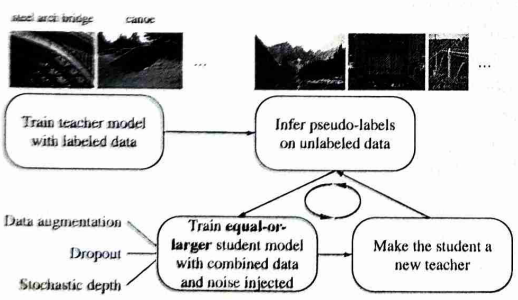


Figure 1: Illustration of the Noisy Student Training. (All shown images are from ImageNet.)

The algorithm is an improved version of self-training, a method in semi-supervised learning (*e.g.*, [71, 96]), and distillation [33]. More discussions on how our method is related to prior works are included in Section 5.

Our key improvements lie in adding noise to the student and using student models that are not smaller than the teacher. This makes our method different from Knowledge Distillation [33] where 1) noise is often not used and 2) a smaller student model is often used to be faster than the teacher. One can think of our method as *knowledge expansion* in which we want the student to be better than the teacher by giving the student model enough capacity and difficult environments in terms of noise to learn through.

Noising Student – When the student is deliberately noised it is trained to be consistent to the teacher that is not noised when it generates pseudo labels. In our experiments, we use two types of noise: input noise and model noise. For input noise, we use data augmentation with RandAugment [18]. For model noise, we use dropout [76] and stochastic depth [37].

When applied to unlabeled data, noise has an important benefit of enforcing invariances in the decision function on both labeled and unlabeled data. First, *data augmentation* is an important noising method in Noisy Student Training because it forces the student to ensure prediction consistency across augmented versions of an image (similar to UDA [91]). Specifically, in our method, the teacher produces high-quality pseudo labels by reading in clean images, while the student is required to reproduce those labels with augmented images as input. For example, the student must ensure that a translated version of an image should have the same category as the original image. Second, when *dropout* and *stochastic depth function* are used as noise, the teacher behaves like an ensemble at inference time (when it generates pseudo labels), whereas the student behaves like a single model. In other words, the student is forced to mimic a more powerful ensemble model. We present an ablation study on the effects of noise in Section 4.1.

Other Techniques – Noisy Student Training also works better with an additional trick, data filtering and balancing, similar to [91, 93]. Specifically, we filter images that the teacher model has low confidences on since they are usually out-of-domain images. To ensure that the distribution of the unlabeled images match that of the training set, we also need to balance the number of unlabeled images for each class, as all classes in ImageNet have a similar number of labeled images. For this purpose, we duplicate images in classes where there are not enough images. For classes where we have too many images, we take the images with the highest confidence.[2]

Finally, we emphasize that our method can be used with

[2]The benefits of data balancing is significant for small models while less significant for larger models. See Study #5 in Appendix A.2 for more details.

soft or hard pseudo labels as both work well in our experiments. Soft pseudo labels, in particular, work slightly better for out-of-domain unlabeled data. Thus in the following, for consistency, we report results with soft pseudo labels unless otherwise indicated.

Comparisons with Existing SSL Methods. Apart from self-training, another important line of work in semi-supervised learning [12, 103] is based on consistency training [3, 64, 67, 84, 89, 91, 8] and pseudo labeling [48, 39, 2, 4]. Although they have produced promising results, in our preliminary experiments, methods based on consistency regularization and pseudo labeling work less well on ImageNet. Instead of using a teacher model trained on labeled data to generate pseudo-labels, these methods do not have a separate teacher model and use the model being trained to generate pseudo-labels. In the early phase of training, the model being trained has low accuracy and high entropy, hence consistency training regularizes the model towards high entropy predictions, and prevents it from achieving good accuracy. A common workaround is to use entropy minimization, to filter examples with low confidence or to ramp up the consistency loss. However, the additional hyperparameters introduced by the ramping up schedule, confidence-based filtering and the entropy minimization make them more difficult to use at scale. The self-training / teacher-student framework is better suited for ImageNet because we can train a good teacher on ImageNet using labeled data.

3. Experiments

In this section, we will first describe our experiment details. We will then present our ImageNet results compared with those of state-of-the-art models. Lastly, we demonstrate the surprising improvements of our models on robustness datasets (such as ImageNet-A, C and P) as well as under adversarial attacks.

3.1. Experiment Details

Labeled dataset. We conduct experiments on ImageNet 2012 ILSVRC challenge prediction task since it has been considered one of the most heavily benchmarked datasets in computer vision and that improvements on ImageNet transfer to other datasets [44, 66].

Unlabeled dataset. We obtain unlabeled images from the JFT dataset [33, 15], which has around 300M images. Although the images in the dataset have labels, we ignore the labels and treat them as unlabeled data. We filter the ImageNet validation set images from the dataset (see [58]).

We then perform data filtering and balancing on this corpus. First, we run an EfficientNet-B0 trained on ImageNet [83] over the JFT dataset [33, 15] to predict a label for each image. We then select images that have confidence of the label higher than 0.3. For each class, we select at most 130K images that have the highest confidence. Finally, for classes that have less than 130K images, we duplicate some images at random so that each class can have 130K images. Hence the total number of images that we use for training a student model is 130M (with some duplicated images). Due to duplications, there are only 81M unique images among these 130M images. We do not tune these hyperparameters extensively since our method is highly robust to them.

To enable fair comparisons with our results, we also experiment with a public dataset YFCC100M [85] and show the results in Appendix A.4. Yahoo Flickr Creative Commons Dataset 100M.

Architecture. We use EfficientNets [83] as our baseline models because they provide better capacity for more data. In our experiments, we also further scale up EfficientNet-B7 and obtain EfficientNet-L2. EfficientNet-L2 is wider and deeper than EfficientNet-B7 but uses a lower resolution, which gives it more parameters to fit a large number of unlabeled images. Due to the large model size, the training time of EfficientNet-L2 is approximately five times the training time of EfficientNet-B7. For more information about EfficientNet-L2, please refer to Table 8 in Appendix A.1.

Training details. For labeled images, we use a batch size of 2048 by default and reduce the batch size when we could not fit the model into the memory. We find that using a batch size of 512, 1024, and 2048 leads to the same performance. We determine the number of training steps and the learning rate schedule by the batch size for labeled images. Specifically, we train the student model for 350 epochs for models larger than EfficientNet-B4, including EfficientNet-L2 and train smaller student models for 700 epochs. The learning rate starts at 0.128 for labeled batch size 2048 and decays by 0.97 every 2.4 epochs if trained for 350 epochs or every 4.8 epochs if trained for 700 epochs.

We use a large batch size for unlabeled images, especially for large models, to make full use of large quantities of unlabeled images. Labeled images and unlabeled images are concatenated together to compute the average cross entropy loss. We apply the recently proposed technique to fix train-test resolution discrepancy [86] for EfficientNet-L2. We first perform normal training with a smaller resolution for 350 epochs. Then we finetune the model with a larger resolution for 1.5 epochs on unaugmented labeled images. Similar to [86], we fix the shallow layers during finetuning.

Our largest model, EfficientNet-L2, needs to be trained for 6 days on a Cloud TPU v3 Pod, which has 2048 cores, if the unlabeled batch size is 14x the labeled batch size.

Method	# Params	Extra Data	Top-1 Acc.	Top-5 Acc.
ResNet-50 [30]	26M	-	76.0%	93.0%
ResNet-152 [30]	60M	-	77.8%	93.8%
DenseNet-264 [36]	34M	-	77.9%	93.9%
Inception-v3 [83]	24M	-	78.8%	94.4%
Xception [5]	23M	-	79.0%	94.5%
Inception-v4 [82]	48M	-	80.0%	95.0%
Inception-resnet-v2 [79]	56M	-	80.1%	95.1%
ResNeXt-101 [85]	84M	-	80.9%	95.6%
PolyNet [98]	92M	-	81.3%	95.8%
SENet [35]	146M	-	82.7%	96.2%
NASNet-A [99]	89M	-	82.7%	96.2%
AmoebaNet-A [68]	87M	-	82.8%	96.1%
PNASNet [50]	86M	-	82.9%	96.2%
AmoebaNet-C [17]	155M	-	83.5%	96.5%
GPipe [38]	557M	-	84.3%	97.0%
EfficientNet-B7 [83]	66M	-	85.0%	97.2%
EfficientNet-L2 [83]	480M	-	85.5%	97.5%
ResNet-50 Billion-scale [93]	26M	3.5B images labeled with tags	81.2%	96.0%
ResNeXt-101 Billion-scale [93]	193M		84.8%	-
ResNeXt-101 WSL [55]	829M		85.4%	97.6%
FixRes ResNeXt-101 WSL [86]	829M		86.4%	98.0%
Big Transfer (BiT-L) [43]†	928M	300M weakly labeled images from JFT	87.5%	98.5%
Noisy Student Training (EfficientNet-L2)	480M	300M unlabeled images from JFT	**88.4%**	**98.7%**

Table 2: Top-1 and Top-5 Accuracy of Noisy Student Training and previous state-of-the-art methods on ImageNet. EfficientNet-L2 with Noisy Student Training is the result of iterative training for multiple iterations by putting back the student model as the new teacher. It has better tradeoff in terms of accuracy and model size compared to previous state-of-the-art models. †: Big Transfer is a concurrent work that performs transfer learning from the JFT dataset.

Noise. We use stochastic depth [37], dropout [76], and RandAugment [18] to noise the student. The hyperparameters for these noise functions are the same for EfficientNet-B7 and L2. In particular, we set the survival probability in stochastic depth to 0.8 for the final layer and follow the linear decay rule for other layers. We apply dropout to the final layer with a dropout rate of 0.5. For RandAugment, we apply two random operations with magnitude set to 27.

Iterative training. The best model in our experiments is a result of three iterations of putting back the student as the new teacher. We first trained an EfficientNet-B7 on ImageNet as the teacher model. Then by using the B7 model as the teacher, we trained an EfficientNet-L2 model with the unlabeled batch size set to 14 times the labeled batch size. Then, we trained a new EfficientNet-L2 model with the EfficientNet-L2 model as the teacher. Lastly, we iterated again and used an unlabeled batch size of 28 times the labeled batch size. The detailed results of the three iterations are available in Section 4.2.

3.2. ImageNet Results

We first report the validation set accuracy on the ImageNet 2012 ILSVRC challenge prediction task as commonly done in literature [45, 80, 30, 83] (see also [66]). As shown in Table 2, EfficientNet-L2 with Noisy Student Training achieves 88.4% top-1 accuracy which is significantly better than the best reported accuracy on EfficientNet of 85.0%. The total gain of 3.4% comes from two sources: by making the model larger (+0.5%) and by Noisy Student Training (+2.9%). In other words, Noisy Student Training makes a much larger impact on the accuracy than changing the architecture.

Further, Noisy Student Training outperforms the state-of-the-art accuracy of 86.4% by FixRes ResNeXt-101 WSL [55, 86] that requires 3.5 Billion Instagram images labeled with tags. As a comparison, our method only requires 300M unlabeled images, which is perhaps more easy to collect. Our model is also approximately twice as small in the number of parameters compared to FixRes ResNeXt-101 WSL.

Model size study: Noisy Student Training for EfficientNet B0-B7 without Iterative Training. In addition to improving state-of-the-art results, we conduct experiments to verify if Noisy Student Training can benefit other EfficienetNet models. In previous experiments, iterative training was used to optimize the accuracy of EfficientNet-L2 but here we skip it as it is difficult to use iterative training for many experiments. We vary the model size from EfficientNet-B0 to EfficientNet-B7 [83] and use the same model as both the teacher and the student. We apply RandAugment to all EfficientNet baselines, leading to more competitive baselines. We set the unlabeled batch size to be three times the batch size of labeled images for all model sizes except for EfficientNet-B0. For EfficientNet-B0, we set the unlabeled batch size to be the same as the batch size of labeled images. As shown in Figure 2, Noisy Student Training leads to a consistent improvement of around 0.8% for all model sizes. Overall, EfficientNets with Noisy Student Training provide a much better tradeoff between model size and accuracy than prior works. The results also confirm that vision models can benefit from Noisy Student Training even without iterative training.

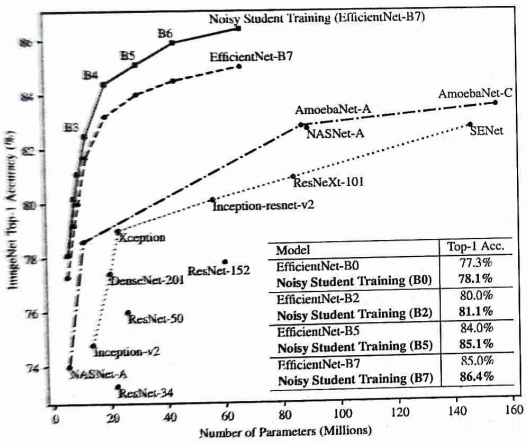


Figure 2: Noisy Student Training leads to significant improvements across all model sizes. We use the same architecture for the teacher and the student and do not perform iterative training.

3.3. Robustness Results on ImageNet-A, ImageNet-C and ImageNet-P

We evaluate the best model, that achieves 88.4% top-1 accuracy, on three robustness test sets: ImageNet-A, ImageNet-C and ImageNet-P. ImageNet-C and P test sets [31] include images with common corruptions and perturbations such as blurring, fogging, rotation and scaling. ImageNet-A test set [32] consists of difficult images that cause significant drops in accuracy to state-of-the-art models. These test sets are considered as "robustness" benchmarks because the test images are either much harder, for ImageNet-A, or the test images are different from the training images, for ImageNet-C and P.

Method	Top-1 Acc.	Top-5 Acc.
ResNet-101 [32]	4.7%	-
ResNeXt-101 [32] (32x4d)	5.9%	-
ResNet-152 [32]	6.1%	-
ResNeXt-101 [32] (64x4d)	7.3%	-
DPN-98 [32]	9.4%	-
ResNeXt-101+SE [32] (32x4d)	14.2%	-
ResNeXt-101 WSL [55, 59]	61.0%	-
EfficientNet-L2	49.6%	78.6%
Noisy Student Training (L2)	**83.7%**	**95.2%**

Table 3: Robustness results on ImageNet-A.

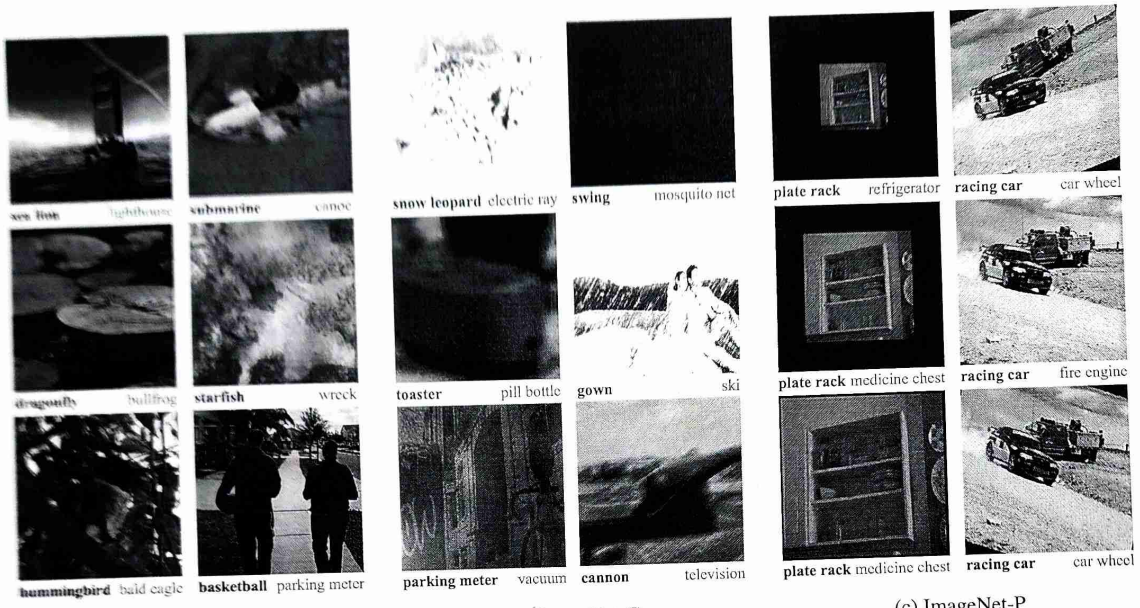
Method	Res.	Top-1 Acc.	mCE
ResNet-50 [31]	224	39.0%	76.7
SIN [23]	224	45.2%	69.3
Patch Gaussian [51]	299	52.3%	60.4
ResNeXt-101 WSL [55, 59]	224	-	45.7
EfficientNet-L2	224	62.6%	47.5
Noisy Student Training (L2)	224	76.5%	30.0
EfficientNet-L2	299	66.6%	42.5
Noisy Student Training (L2)	299	**77.8%**	**28.3**

Table 4: Robustness results on ImageNet-C. mCE is the weighted average of error rate on different corruptions, with AlexNet's error rate as a baseline (lower is better).

Method	Res.	Top-1 Acc.	mFR
ResNet-50 [31]	224	-	58.0
Low Pass Filter Pooling [99]	224	-	51.2
ResNeXt-101 WSL [55, 59]	224	-	27.8
EfficientNet-L2	224	80.4%	27.2
Noisy Student Training (L2)	224	85.2%	14.2
EfficientNet-L2	299	81.6%	23.7
Noisy Student Training (L2)	299	**86.4%**	**12.2**

Table 5: Robustness results on ImageNet-P, where images are generated with a sequence of perturbations. mFR measures the model's probability of flipping predictions under perturbations with AlexNet as a baseline (lower is better).

For ImageNet-C and ImageNet-P, we evaluate models on two released versions with resolution 224x224 and 299x299 and resize images to the resolution EfficientNet trained on.



(a) ImageNet-A (b) ImageNet-C (c) ImageNet-P

Figure 3: Selected images from robustness benchmarks ImageNet-A, C and P. Test images from ImageNet-C underwent artificial transformations (also known as common corruptions) that cannot be found on the ImageNet training set. Test images on ImageNet-P underwent different scales of perturbations. On ImageNet-A, C, EfficientNet with Noisy Student Tranining produces correct top-1 predictions (shown in **bold black** texts) and EfficientNet without Noisy Student Training produces incorrect top-1 predictions (shown in red texts). On ImageNet-P, EfficientNet without Noisy Student Training flips predictions frequently.

As shown in Table 3, 4 and 5, Noisy Student Training yields substantial gains on robustness datasets compared to the previous state-of-the-art model ResNeXt-101 WSL [55, 59] trained on 3.5B weakly labeled images. On ImageNet-A, it improves the top-1 accuracy from 61.0% to 83.7%. On ImageNet-C, it reduces mean corruption error (mCE) from 45.7 to 28.3. On ImageNet-P, it leads to a mean flip rate (mFR) of 14.2 if we use a resolution of 224x224 (direct comparison) and 12.2 if we use a resolution of 299x299.[3] These significant gains in robustness in ImageNet-C and ImageNet-P are surprising because our method was not deliberately optimized for robustness.[4]

Qualitative Analysis. To intuitively understand the significant improvements on the three robustness benchmarks, we show several images in Figure 3 where the predictions of the standard model are incorrect while the predictions of the model with Noisy Student Training are correct.

Figure 3a shows example images from ImageNet-A and the predictions of our models. The model with Noisy Student Training can successfully predict the correct labels of these highly difficult images. For example, without Noisy Student Training, the model predicts *bullfrog* for the image shown on the left of the second row, which might be resulted from the black lotus leaf on the water. With Noisy Student Training, the model correctly predicts *dragonfly* for the image. At the top-left image, the model without Noisy Student Training ignores the *sea lion*s and mistakenly recognizes a buoy as a lighthouse, while the model with Noisy Student Training can recognize the *sea lion*s.

Figure 3b shows images from ImageNet-C and the corresponding predictions. As can be seen from the figure, our model with Noisy Student Training makes correct predictions for images under severe corruptions and perturbations such as snow, motion blur and fog, while the model without

[3] For EfficientNet-L2, we use the model without finetuning with a larger test time resolution, since a larger resolution results in a discrepancy with the resolution of data and leads to degraded performance on ImageNet-C and ImageNet-P.

[4] Note that both our model and ResNeXt-101 WSL use augmentations that have a small overlap with corruptions in ImageNet-C, which might result in better performance. Specifically, RandAugment includes augmentation Brightness, Contrast and Sharpness. ResNeXt-101 WSL uses augmentation of Brightness and Contrast.

Noisy Student Training suffers greatly under these conditions. The most interesting image is shown on the right of the first row. The *swing* in the picture is barely recognizable by human while the model with Noisy Student Training still makes the correct prediction.

Figure 3c shows images from ImageNet-P and the corresponding predictions. As can be seen, our model with Noisy Student Training makes correct and consistent predictions as images undergone different perturbations while the model without Noisy Student Training flips predictions frequently.

3.4. Adversarial Robustness Results

After testing our model's robustness to common corruptions and perturbations, we also study its performance on adversarial perturbations. We evaluate our EfficientNet-L2 models with and without Noisy Student Training against an FGSM attack. This attack performs one gradient descent step on the input image [25] with the update on each pixel set to ϵ. As shown in Figure 4, Noisy Student Training leads to very significant improvements in accuracy even though the model is not optimized for adversarial robustness. Under a stronger attack PGD with 10 iterations [54], at $\epsilon = 16$, Noisy Student Training improves EfficientNet-L2's accuracy from 1.1% to 4.4%.

Note that these adversarial robustness results are not directly comparable to prior works since we use a large input resolution of 800x800 and adversarial vulnerability can scale with the input dimension [22, 25, 24, 74].

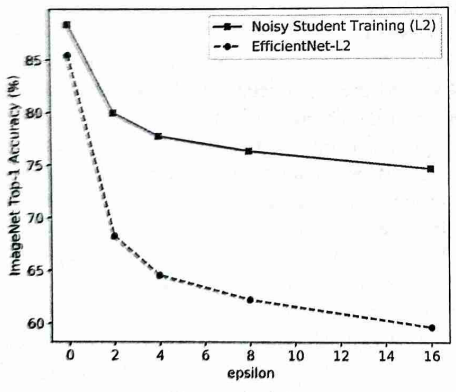


Figure 4: Noisy Student Training improves adversarial robustness against an FGSM attack though the model is not optimized for adversarial robustness. The accuracy is improved by 11% at $\epsilon = 2$ and gets better as ϵ gets larger.

4. Ablation Study

In this section, we study the importance of noise and iterative training and summarize the ablations for other components of our method.

4.1. The Importance of Noise in Self-training

Since we use soft pseudo labels generated from the teacher model, when the student is trained to be exactly the same as the teacher model, the cross entropy loss on unlabeled data would be zero and the training signal would vanish. Hence, a question that naturally arises is why the student can outperform the teacher with soft pseudo labels. As stated earlier, we hypothesize that noising the student is needed so that it does not merely learn the teacher's knowledge. We investigate the importance of noising in two scenarios with different amounts of unlabeled data and different teacher model accuracies. In both cases, we gradually remove augmentation, stochastic depth and dropout for unlabeled images when training the student model, while keeping them for labeled images. This way, we can isolate the influence of noising on unlabeled images from the influence of preventing overfitting for labeled images. In addition, we compare using a noised teacher and an unnoised teacher to study if it is necessary to disable noise when generating pseudo labels.

Here, we show the evidence in Table 6, noise such as stochastic depth, dropout and data augmentation plays an important role in enabling the student model to perform better than the teacher. The performance consistently drops with noise function removed. However, in the case with 130M unlabeled images, when compared to the supervised baseline, the performance is still improved to 84.3% from 84.0% with noise function removed. We hypothesize that the improvement can be attributed to SGD, which introduces stochasticity into the training process.

One might argue that the improvements from using noise can be resulted from preventing overfitting the pseudo labels on the unlabeled images. We verify that this is not the case when we use 130M unlabeled images since the model does not overfit the unlabeled set from the training loss. While removing noise leads to a much lower training loss for labeled images, we observe that, for unlabeled images, removing noise leads to a smaller drop in training loss. This is probably because it is harder to overfit the large unlabeled dataset.

Lastly, adding noise to the teacher model that generates pseudo labels leads to lower accuracy, which shows the importance of having a powerful unnoised teacher model.

4.2. A Study of Iterative Training

Here, we show the detailed effects of iterative training. As mentioned in Section 3.1, we first train an EfficientNet-B7 model on labeled data and then use it as the teacher to

Model / Unlabeled Set Size	1.3M	130M
EfficientNet-B5	83.3%	84.0%
Noisy Student Training (B5)	**83.9%**	**85.1%**
student w/o Aug	83.6%	84.6%
student w/o Aug, SD, Dropout	83.2%	84.3%
teacher w. Aug, SD, Dropout	83.7%	84.4%

Table 6: Ablation study of noising. We use EfficientNet-B5 as the teacher model and study two cases with different numbers of unlabeled images and different augmentations. For the experiment with 1.3M unlabeled images, we use the standard augmentation including random translation and flipping for both the teacher and the student. For the experiment with 130M unlabeled images, we use RandAugment. Aug and SD denote data augmentation and stochastic depth respectively. We remove the noise for unlabeled images while keeping them for labeled images. Here, iterative training is not used and unlabeled batch size is set to be the same as the labeled batch size to save training time.

train an EfficientNet-L2 student model. Then, we iterate this process by putting back the new student model as the teacher model.

As shown in Table 7, the model performance improves to 87.6% in the first iteration and then to 88.1% in the second iteration with the same hyperparameters (except using a teacher model with better performance). These results indicate that iterative training is effective in producing increasingly better models. For the last iteration, we make use of a larger ratio between unlabeled batch size and labeled batch size to boost the final performance to 88.4%.

Iteration	Model	Batch Size Ratio	Top-1 Acc.
1	EfficientNet-L2	14:1	87.6%
2	EfficientNet-L2	14:1	88.1%
3	EfficientNet-L2	28:1	88.4%

Table 7: Iterative training improves the accuracy, where batch size ratio denotes the ratio between unlabeled data and labeled data.

4.3. Additional Ablation Study Summarization

We also study the importance of various design choices of Noisy Student Training, hopefully offering a practical guide for readers. With this purpose, we conduct 8 ablation studies in Appendix A.2. The findings are summarized as follows:

- **Finding #1:** Using *a large teacher model* with better performance leads to better results.
- **Finding #2:** *A large amount of unlabeled data* is necessary for better performance.
- **Finding #3:** *Soft pseudo labels* work better than hard pseudo labels for out-of-domain data in certain cases.
- **Finding #4:** *A large student model* is important to enable the student to learn a more powerful model.
- **Finding #5:** *Data balancing* is useful for small models.
- **Finding #6:** *Joint training* on labeled data and unlabeled data outperforms the pipeline that first pretrains with unlabeled data and then finetunes on labeled data.
- **Finding #7:** Using *a large ratio between unlabeled batch size and labeled batch size* enables models to train longer on unlabeled data to achieve a higher accuracy.
- **Finding #8:** *Training the student from scratch* is sometimes better than initializing the student with the teacher and the student initialized with the teacher still requires a large number of training epochs to perform well.

5. Related works

Self-training. Our work is based on self-training (*e.g.*, [71, 96, 68, 67]). Self-training first uses labeled data to train a good teacher model, then use the teacher model to label unlabeled data and finally use the labeled data and unlabeled data to jointly train a student model. In typical self-training with the teacher-student framework, noise injection to the student is not used by default, or the role of noise is not fully understood or justified. The main difference between our work and prior works is that we identify the importance of noise, and aggressively inject noise to make the student better.

Self-training was previously used to improve ResNet-50 from 76.4% to 81.2% top-1 accuracy [93] which is still far from the state-of-the-art accuracy. Yalniz *et al.* [93] also did not show significant improvements in terms of robustness on ImageNet-A, C and P as we did. In terms of methodology, they proposed to first only train on unlabeled images and then finetune their model on labeled images as the final stage. In Noisy Student Training, we combine these two steps into one because it simplifies the algorithm and leads to better performance in our experiments.

Data Distillation [63], which ensembles predictions for an image with different transformations to strengthen the teacher, is the opposite of our approach of weakening the student. Parthasarathi *et al.* [61] find a small and fast speech recognition model for deployment via knowledge distillation on unlabeled data. As noise is not used and the student is also small, it is difficult to make the student better than teacher. The domain adaptation framework in [69] is related but highly optimized for videos, *e.g.*, prediction on

which frame to use in a video. The method in [101] ensembles predictions from multiple teacher models, which is more expensive than our method.

Co-training [3] divides features into two disjoint partitions and trains two models with the two sets of features using labeled data. Their source of "noise" is the feature partitioning such that two models do not always agree on unlabeled data. Our method of injecting noise to the student model also enables the teacher and the student to make different predictions and is more suitable for ImageNet than partitioning features.

Self-training / co-training has also been shown to work well for a variety of other tasks including leveraging noisy data [3], semantic segmentation [4], text classification [40, 78]. Back translation and self-training have led to significant improvements in machine translation [72, 20, 28, 14, 6, 29].

Semi-supervised Learning. Apart from self-training, another important line of work in semi-supervised learning [12, 83] is based on consistency training [5, 64, 47, 84, 76, 52, 62, 13, 10, 60, 2, 49, 88, 91, 8, 98, 46, 7]. They constrain model predictions to be invariant to noise injected to the input, hidden states or model parameters. As discussed in Section 2, consistency regularization works less well on ImageNet because consistency regularization uses a model being trained to generate the pseudo-labels. In the early phase of training, they regularize the model towards high entropy predictions, and prevents it from achieving good accuracy.

Works based on pseudo label [48, 39, 73, 1] are similar to self-training, but also suffer the same problem with consistency training, since they rely on a model being trained instead of a converged model with high accuracy to generate pseudo labels. Finally, frameworks in semi-supervised learning also include graph-based methods [102, 89, 94, 42], methods that make use of latent variables as target variables [41, 53, 95] and methods based on low-density separation [26, 70, 19], which might provide complementary benefits to our method.

Knowledge Distillation. Our work is also related to methods in Knowledge Distillation [10, 3, 33, 21, 6] via the use of soft targets. The main use of knowledge distillation is model compression by making the student model smaller. The main difference between our method and knowledge distillation is that knowledge distillation does not consider unlabeled data and does not aim to improve the student model.

Robustness. A number of studies, *e.g.* [82, 31, 66, 27], have shown that vision models lack robustness. Addressing the lack of robustness has become an important research direction in machine learning and computer vision in recent years. Our study shows that using unlabeled data improves accuracy and general robustness. Our finding is consistent with arguments that using unlabeled data can improve *adversarial* robustness [11, 77, 57, 97]. The main difference between our work and these works is that they directly optimize adversarial robustness on unlabeled data, whereas we show that Noisy Student Training improves robustness greatly even without directly optimizing robustness.

6. Conclusion

Prior works on weakly-supervised learning required billions of weakly labeled data to improve state-of-the-art ImageNet models. In this work, we showed that it is possible to use unlabeled images to significantly advance both accuracy and robustness of state-of-the-art ImageNet models. We found that self-training is a simple and effective algorithm to leverage unlabeled data at scale. We improved it by adding noise to the student, hence the name Noisy Student Training, to learn beyond the teacher's knowledge.

Our experiments showed that Noisy Student Training and EfficientNet can achieve an accuracy of 88.4% which is 2.9% higher than without Noisy Student Training. This result is also a new state-of-the-art and 2.0% better than the previous best method that used an order of magnitude more weakly labeled data [55, 86].

An important contribution of our work was to show that Noisy Student Training boosts robustness in computer vision models. Our experiments showed that our model significantly improves performances on ImageNet-A, C and P.

Acknowledgement

We thank the Google Brain team, Zihang Dai, Jeff Dean, Hieu Pham, Colin Raffel, Ilya Sutskever and Mingxing Tan for insightful discussions, Cihang Xie, Dan Hendrycks and A. Emin Orhan for robustness evaluation, Sergey Ioffe, Guokun Lai, Jiquan Ngiam, Jiateng Xie and Adams Wei Yu for feedbacks on the draft, Yanping Huang, Pankaj Kanwar, Naveen Kumar, Sameer Kumar and Zak Stone for great help with TPUs, Ekin Dogus Cubuk and Barret Zoph for help with RandAugment, Tom Duerig, Victor Gomes, Paul Haahr, Pandu Nayak, David Price, Janel Thamkul, Elizabeth Trumbull, Jake Walker and Wenlei Zhou for help with model releases, Yanan Bao, Zheyun Feng and Daiyi Peng for help with the JFT dataset, Ola Spyra and Olga Wichrowska for help with infrastructure.

References

[1] Eric Arazo, Diego Ortego, Paul Albert, Noel E O'Connor, and Kevin McGuinness. Pseudo-labeling and confirmation bias in deep semi-supervised learning. *arXiv preprint arXiv:1908.02983*, 2019. 3, 9

[2] Ben Athiwaratkun, Marc Finzi, Pavel Izmailov, and Andrew Gordon Wilson. There are many consistent explanations of unlabeled data: Why you should average. In *International Conference on Learning Representations*, 2018. 9

[3] Jimmy Ba and Rich Caruana. Do deep nets really need to be deep? In *Advances in Neural Information Processing Systems*, pages 2654–2662, 2014. 9

[4] Yauhen Babakhin, Artsiom Sanakoyeu, and Hirotoshi Kitamura. Semi-supervised segmentation of salt bodies in seismic images using an ensemble of convolutional neural networks. *arXiv preprint arXiv:1904.04445*, 2019. 9

[5] Philip Bachman, Ouais Alsharif, and Doina Precup. Learning with pseudo-ensembles. In *Advances in Neural Information Processing Systems*, pages 3365–3373, 2014. 3, 9

[6] Anoop Korattikara Balan, Vivek Rathod, Kevin P Murphy, and Max Welling. Bayesian dark knowledge. In *Advances in Neural Information Processing Systems*, pages 3438–3446, 2015. 9

[7] 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. *arXiv preprint arXiv:1911.09785*, 2019. 9

[8] 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. 3, 9

[9] Avrim Blum and Tom Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the eleventh annual conference on Computational learning theory*, pages 92–100. ACM, 1998. 9

[10] Cristian Bucilu, Rich Caruana, and Alexandru Niculescu-Mizil. Model compression. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 535–541. ACM, 2006. 9

[11] Yair Carmon, Aditi Raghunathan, Ludwig Schmidt, Percy Liang, and John C Duchi. Unlabeled data improves adversarial robustness. *arXiv preprint arXiv:1905.13736*, 2019. 9

[12] Olivier Chapelle, Bernhard Scholkopf, and Alexander Zien. Semi-supervised learning (chapelle, o. et al., eds.; 2006)[book reviews]. *IEEE Transactions on Neural Networks*, 20(3):542–542, 2009. 3, 9

[13] Yanbei Chen, Xiatian Zhu, and Shaogang Gong. Semi-supervised deep learning with memory. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 268–283, 2018. 9

[14] Yong Cheng, Wei Xu, Zhongjun He, Wei He, Hua Wu, Maosong Sun, and Yang Liu. Semi-supervised learning for neural machine translation. *arXiv preprint arXiv:1606.04596*, 2016. 9

[15] François Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1251–1258, 2017. 3, 4

[16] Kevin Clark, Minh-Thang Luong, Christopher D Manning, and Quoc V Le. Semi-supervised sequence modeling with cross-view training. In *Empirical Methods in Natural Language Processing (EMNLP)*, 2018. 9

[17] Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. AutoAugment: Learning augmentation strategies from data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 4

[18] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical data augmentation with no separate search. *arXiv preprint arXiv:1909.13719*, 2019. 1, 2, 4

[19] Zihang Dai, Zhilin Yang, Fan Yang, William W Cohen, and Ruslan R Salakhutdinov. Good semi-supervised learning that requires a bad gan. In *Advances in Neural Information Processing Systems*, pages 6510–6520, 2017. 9

[20] Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. Understanding back-translation at scale. In *Proceedings of the 2018 conference on Empirical methods in natural language processing*, pages 489–500, 2018. 9

[21] Tommaso Furlanello, Zachary C Lipton, Michael Tschannen, Laurent Itti, and Anima Anandkumar. Born again neural networks. In *International Conference on Machine Learning*, 2018. 9

[22] Angus Galloway, Anna Golubeva, Thomas Tanay, Medhat Moussa, and Graham W Taylor. Batch normalization is a cause of adversarial vulnerability. *arXiv preprint arXiv:1905.02161*, 2019. 7

[23] Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A Wichmann, and Wieland Brendel. ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. In *International Conference on Learning Representations*, 2019. 5

[24] Justin Gilmer, Luke Metz, Fartash Faghri, Samuel S Schoenholz, Maithra Raghu, Martin Wattenberg, and Ian Goodfellow. Adversarial spheres. *arXiv preprint arXiv:1801.02774*, 2018. 7

[25] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*, 2015. 7

[26] Yves Grandvalet and Yoshua Bengio. Semi-supervised learning by entropy minimization. In *Advances in neural information processing systems*, pages 529–536, 2005. 9

[27] Keren Gu, Brandon Yang, Jiquan Ngiam, Quoc Le, and Jonathan Shlens. Using videos to evaluate image model robustness. In *ICLR Workshop*, 2019. 9

[28] Di He, Yingce Xia, Tao Qin, Liwei Wang, Nenghai Yu, Tie-Yan Liu, and Wei-Ying Ma. Dual learning for machine translation. In *Advances in Neural Information Processing Systems*, pages 820–828, 2016. 9

[29] Junxian He, Jiatao Gu, Jiajun Shen, and Marc'Aurelio Ranzato. Revisiting self-training for neural sequence generation. *arXiv preprint arXiv:1909.13788*, 2019. 9

[30] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 1, 4, 16