# Self-training with Noisy Student improves ImageNet classification

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

## Abstract

*We present a simple self-training method that achieves 87.4% top-1 accuracy on ImageNet, which is 1.0% 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 16.6% to 74.2%, reduces ImageNet-C mean corruption error from 45.7 to 31.2, and reduces ImageNet-P mean flip rate from 27.8 to 16.1.*

*To achieve this result, we first train an EfficientNet model on labeled ImageNet images and use it as a teacher to generate pseudo labels on 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 generation of the pseudo labels, the teacher is not noised so that the pseudo labels are as good as possible. But during the learning of the student, we inject noise such as data augmentation, dropout, stochastic depth to the student so that the noised student is forced to learn harder from the pseudo labels.*

## 1. Introduction

Deep learning has shown remarkable successes in image recognition in recent years [35, 66, 62, 23, 69]. However state-of-the-art 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 state-of-the-art models.

Here we use unlabeled images to improve the state-of-the-art ImageNet accuracy and show that the accuracy gain has an outsized impact on robustness. For this purpose, we use a much larger corpus of unlabeled images, where some images may not belong to any category in ImageNet. We train our model using the self-training framework [59] 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. Finally, we iterate the algorithm a few times by treating the student as a teacher to generate new pseudo labels and train a new student.

Our experiments show that an important element for this simple method to work well at scale is that the student model should be noised during its training while the teacher should not be noised during the generation of pseudo labels. This way, the pseudo labels are as good as possible, and the noised student is forced to learn harder from the pseudo labels. To noise the student, we use dropout [63], data augmentation [14] and stochastic depth [29] during its training. We call the method self-training with Noisy Student to emphasize the role that noise plays in the method and results. To achieve strong results on ImageNet, the student model also needs to be large, typically larger than common vision models, so that it can leverage a large number of unlabeled images.

Using self-training with Noisy Student, together with 300M unlabeled images, we improve EfficientNet's [69] ImageNet top-1 accuracy to 87.4%. This accuracy is 1.0% better than the previous state-of-the-art ImageNet accuracy 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 [25] top-1 accuracy from 16.6% to 74.2%, ImageNet-C [24] mean corruption error (mCE) from 45.7 to 31.2 and ImageNet-P [24] mean flip rate (mFR) from 27.8 to 16.1. Our main results are shown in Table 1.

| | ImageNet<br>top-1 acc. | ImageNet-A<br>top-1 acc. | ImageNet-C<br>mCE | ImageNet-P<br>mFR |
|---|---|---|---|---|
| Prev. SOTA | 86.4% | 16.6% | 45.7 | 27.8 |
| Ours | **87.4%** | **74.2%** | **31.2** | **16.1** |

Table 1: Summary of key results compared to previous state-of-the-art models [71, 44]. Lower is better for mean corruption error (mCE) and mean flip rate (mFR).

[*] This work was conducted at Google.

## 2. Self-training with Noisy Student

Algorithm 1 gives an overview of self-training with Noisy Student (or Noisy Student in short). 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.

```
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 $\theta_*$ which minimizes the cross entropy
    loss on labeled images
        $\frac{1}{n}\sum_{i=1}^{n} \ell(y_i, f^{noised}(x_i, \theta))$
2: Use an unnoised teacher model to generate soft or hard
    pseudo labels for unlabeled images
        $\tilde{y}_i = f(\tilde{x}_i, \theta_*), \forall i = 1, \cdots, m$
3: Learn student model $\theta'_*$ which minimizes the cross entropy
    loss on labeled images and unlabeled images with noise
    added to the student model
        $\frac{1}{n}\sum_{i=1}^{n} \ell(y_i, f^{noised}(x_i, \theta')) + \frac{1}{m}\sum_{i=1}^{m} \ell(\tilde{y}_i, f^{noised}(\tilde{x}_i, \theta'))$
4: Iterative training: Use the student as a teacher and go back to
    step 2.
```

**Algorithm 1:** Noisy Student method

The algorithm is basically self-training, a method in semi-supervised learning (*e.g.*, [59, 79]). We will discuss how our method is related to prior works in Section 5. Our main change is to add more sources of noise to the student to significantly improve it while removing the noise in the teacher when the teacher generates the pseudo labels.

When the student model is deliberately noised it is actually trained to be consistent to the more powerful teacher model that is not noised when it generates pseudo labels. In our experiments, we use dropout [63], stochastic depth [29], data augmentation [14] to noise the student.

Although noise may appear to be limited and uninteresting, when it is applied to unlabeled data, it has a compound benefit of enforcing local smoothness in the decision function on both labeled and unlabeled data. Different kinds of noise, however, may have different effects. When data augmentation noise is used, the student must ensure that a translated image, for example, should have the same category with a non-translated image. This invariance constraint reduces the degrees of freedom in the model. When dropout and stochastic depth are used, the teacher model behaves like an ensemble of models (when it generates the pseudo labels, dropout is not used), whereas the student behaves like a single model. In other words, the student is forced to mimic a more powerful ensemble model.

The architectures for the student and teacher models can be the same or different. However an important requirement for Noisy Student to work well is that the student model needs to be sufficiently large to fit more data (labeled and pseudo labeled). For this purpose, we use the recently developed EfficientNet architectures [69] because they have a larger capacity than ResNet architectures [23]. Secondly, to enable the student to learn a more powerful model, we also make the student model larger than the teacher model. This is an important difference between our work and prior works on teacher-student framework whose main goal is model compression.

We find that Noisy Student is better with an additional trick: data balancing. Specifically, as all classes in ImageNet have a similar number of labeled images, we also need to balance the number of unlabeled images for each class. 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.

Finally, in the above, we say that the pseudo labels can be soft or hard. In our experiments, we observe that soft pseudo labels are usually more stable and lead to faster convergence, especially when the teacher model has low accuracy. Hence we use soft pseudo labels for our experiments unless otherwise specified.

## 3. Experiments

In the following, we will first describe experiment details to achieve our results. We will then show our results on ImageNet and compare them with state-of-the-art models. Lastly, we will show the results of benchmarking our model on robustness datasets such as ImageNet-A, C and P and adversarial robustness.

### 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 [34, 55].

**Unlabeled dataset.** We obtain unlabeled images from the JFT dataset [26, 11], which has around 300M images. Although the images in the dataset have labels, we ignore the labels and treat them as unlabeled data. We used the version from [47], which filtered the validation set of ImageNet.

We then perform data filtering and balancing on this corpus. First, we run an EfficientNet-B0 trained on ImageNet [69] over the JFT dataset 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.

**Architecture.** We use EfficientNets [69] 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-L0, L1 and L2. EfficientNet-L0 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 with similar training speed. Then, EfficientNet-L1 is scaled up from EfficientNet-L0 by increasing width. Lastly, we follow the idea of compound scaling [69] and scale all dimensions to obtain EfficientNet-L2. 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 the large architectures, please refer to Table 7 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 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-L0, L1 and L2 and train the student model for 700 epochs for smaller models. 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.

For unlabeled images, we set the batch size to be three times the batch size of labeled images for large models, including EfficientNet-B7, L0, L1 and L2. For smaller models, we set the batch size of unlabeled images to be the same as the batch size of labeled images. In our implementation, labeled images and unlabeled images are concatenated together and we compute the average cross entropy loss.

Lastly, we apply the recently proposed technique to fix train-test resolution discrepancy [71] for EfficientNet-L0, L1 and L2. In particular, 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 [71], we fix the shallow layers during finetuning.

Our largest model, EfficientNet-L2, needs to be trained for 3.5 days on a Cloud TPU v3 Pod, which has 2048 cores.

**Noise.** We use stochastic depth [29], dropout [63] and RandAugment [14] to noise the student. The hyperparameters for these noise functions are the same for EfficientNet-B7, L0, L1 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 classification layer with a dropout rate of 0.5. For RandAugment, we apply two random operations with the magnitude set to 27.

**Iterative training.** The best model in our experiments is a result of iterative training of teacher and student by putting back the student as the new teacher to generate new pseudo labels. During this process, we kept increasing the size of the student model to improve the performance. Our procedure went as follows. We first improved the accuracy of EfficientNet-B7 using EfficientNet-B7 as both the teacher and the student. Then by using the improved B7 model as the teacher, we trained an EfficientNet-L0 student model. Next, with the EfficientNet-L0 as the teacher, we trained a student model EfficientNet-L1, a wider model than L0. Afterward, we further increased the student model size to EfficientNet-L2, with the EfficientNet-L1 as the teacher. Lastly, we trained another EfficientNet-L2 student by using the EfficientNet-L2 model as the teacher.

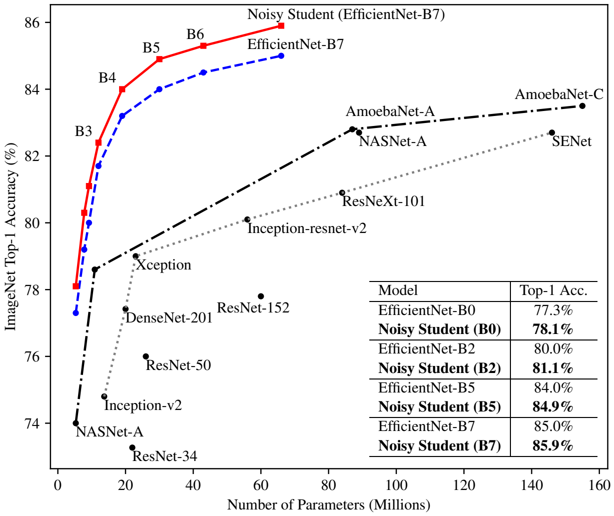
### 3.2. ImageNet Results

We first report the validation set accuracy on the ImageNet 2012 ILSVRC challenge prediction task as commonly done in literature [35, 66, 23, 69] (see also [55]). As shown in Table 2, Noisy Student with EfficientNet-L2 achieves 87.4% top-1 accuracy which is significantly better than the best previously reported accuracy on EfficientNet of 85.0%. The total gain of 2.4% comes from two sources: by making the model larger (+0.5%) and by Noisy Student (+1.9%). In other words, using Noisy Student makes a much larger impact to the accuracy than changing the architecture.

Further, Noisy Student outperforms the state-of-the-art accuracy of 86.4% by FixRes ResNeXt-101 WSL [44, 71] 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.

| Method | # Params | Extra Data | Top-1 Acc. | Top-5 Acc. |
|---|---|---|---|---|
| ResNet-50 [23] | 26M | - | 76.0% | 93.0% |
| ResNet-152 [23] | 60M | - | 77.8% | 93.8% |
| DenseNet-264 [28] | 34M | - | 77.9% | 93.9% |
| Inception-v3 [67] | 24M | - | 78.8% | 94.4% |
| Xception [11] | 23M | - | 79.0% | 94.5% |
| Inception-v4 [65] | 48M | - | 80.0% | 95.0% |
| Inception-resnet-v2 [65] | 56M | - | 80.1% | 95.1% |
| ResNeXt-101 [75] | 84M | - | 80.9% | 95.6% |
| PolyNet [83] | 92M | - | 81.3% | 95.8% |
| SENet [27] | 146M | - | 82.7% | 96.2% |
| NASNet-A [86] | 89M | - | 82.7% | 96.2% |
| AmoebaNet-A [54] | 87M | - | 82.8% | 96.1% |
| PNASNet [39] | 86M | - | 82.9% | 96.2% |
| AmoebaNet-C [13] | 155M | - | 83.5% | 96.5% |
| GPipe [30] | 557M | - | 84.3% | 97.0% |
| EfficientNet-B7 [69] | 66M | - | 85.0% | 97.2% |
| EfficientNet-L2 [69] | 480M | - | 85.5% | 97.5% |
| ResNet-50 Billion-scale [76] | 26M | 3.5B images labeled with tags | 81.2% | 96.0% |
| ResNeXt-101 Billion-scale [76] | 193M | | 84.8% | - |
| ResNeXt-101 WSL [44] | 829M | | 85.4% | 97.6% |
| FixRes ResNeXt-101 WSL [71] | 829M | | 86.4% | 98.0% |
| **Noisy Student (L2)** | 480M | 300M unlabeled images | **87.4%** | **98.2%** |

Table 2: Top-1 and Top-5 Accuracy of Noisy Student and previous state-of-the-art methods on ImageNet. EfficientNets trained with Noisy Student have better tradeoff in terms of accuracy and model size compared to previous state-of-the-art models. Noisy Student (EfficientNet-L2) is the result of iterative training for multiple iterations.



| Model | Top-1 Acc. |
|---|---|
| EfficientNet-B0 | 77.3% |
| **Noisy Student (B0)** | **78.1%** |
| EfficientNet-B2 | 80.0% |
| **Noisy Student (B2)** | **81.1%** |
| EfficientNet-B5 | 84.0% |
| **Noisy Student (B5)** | **84.9%** |
| EfficientNet-B7 | 85.0% |
| **Noisy Student (B7)** | **85.9%** |

Figure 1: Noisy Student leads to significant improvements across all model sizes for EfficientNet. We use the same architecture for the teacher and the student and do not perform iterative training.

**Model size study: Noisy Student for EfficientNet B0-B7 without Iterative Training.** In addition to improving state-of-the-art results, we conduct additional experiments to verify if Noisy Student can benefit other EfficienetNet models. In the above 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 [69] and use the same model as both the teacher and the student. We apply RandAugment to all EfficientNet baselines, leading to more competitive baselines. As shown in Figure 1, Noisy Student leads to a consistent improvement of around 0.8% for all model sizes. Overall, EfficientNets with Noisy Student provide a much better tradeoff between model size and accuracy when compared with prior works. The results also confirm that vision models can benefit from Noisy Student even without iterative training.

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

We evaluate the best model, that achieves 87.4% top-1 accuracy, on three robustness test sets: ImageNet-A, ImageNet-C and ImageNet-P. ImageNet-C and P test sets [24] include images with common corruptions and perturbations such as blurring, fogging, rotation and scaling. ImageNet-A test set [25] 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.

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

| Method | Top-1 Acc. | Top-5 Acc. |
|---|---|---|
| ResNet-101 [25] | 4.7% | - |
| ResNeXt-101 [25] (32x4d) | 5.9% | - |
| ResNet-152 [25] | 6.1% | - |
| ResNeXt-101 [25] (64x4d) | 7.3% | - |
| DPN-98 [25] | 9.4% | - |
| ResNeXt-101+SE [25] (32x4d) | 14.2% | - |
| ResNeXt-101 WSL [44, 48] | 16.6% | - |
| EfficientNet-L2 | 49.6% | 78.6% |
| **Noisy Student (L2)** | **74.2%** | **91.3%** |

Table 3: Robustness results on ImageNet-A.

| Method | Res. | Top-1 Acc. | mCE |
|---|---|---|---|
| ResNet-50 [24] | 224 | 39.0% | 76.7 |
| SIN [18] | 224 | 45.2% | 69.3 |
| Patch Guassian [40] | 299 | 52.3% | 60.4 |
| ResNeXt-101 WSL [44, 48] | 224 | - | 45.7 |
| EfficientNet-L2 | 224 | 62.6% | 47.5 |
| Noisy Student (L2) | 224 | 72.8% | 34.7 |
| EfficientNet-L2 | 299 | 66.6% | 42.5 |
| **Noisy Student (L2)** | 299 | **75.5%** | **31.2** |

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 [24] | 224 | - | 58.0 |
| Low Pass Filter Pooling [82] | 224 | - | 51.2 |
| ResNeXt-101 WSL [44, 48] | 224 | - | 27.8 |
| EfficientNet-L2 | 224 | 80.4% | 27.2 |
| Noisy Student (L2) | 224 | 83.1% | 17.8 |
| EfficientNet-L2 | 299 | 81.6% | 23.7 |
| **Noisy Student (L2)** | 299 | **84.3%** | **16.1** |

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).

As shown in Table 3, 4 and 5, when compared with the previous state-of-the-art model ResNeXt-101 WSL [44, 48] trained on 3.5B weakly labeled images, Noisy Student yields substantial gains on robustness datasets. On ImageNet-C, it reduces mean corruption error (mCE) from 45.7 to 31.2. On ImageNet-P, it leads to an mean flip rate (mFR) of 17.8 if we use a resolution of 224x224 (direct comparison) and 16.1 if we use a resolution of 299x299.[1] These significant gains in robustness in ImageNet-C and ImageNet-P are surprising because our models were not deliberately optimizing for robustness (*e.g.*, via data augmentation).

The biggest gain is observed on ImageNet-A: our method achieves 3.5x higher accuracy on ImageNet-A, going from 16.6% of the previous state-of-the-art to 74.2% top-1 accuracy. In contrast, changing architectures or training with weakly labeled data give modest gains in accuracy from 4.7% to 16.6%.

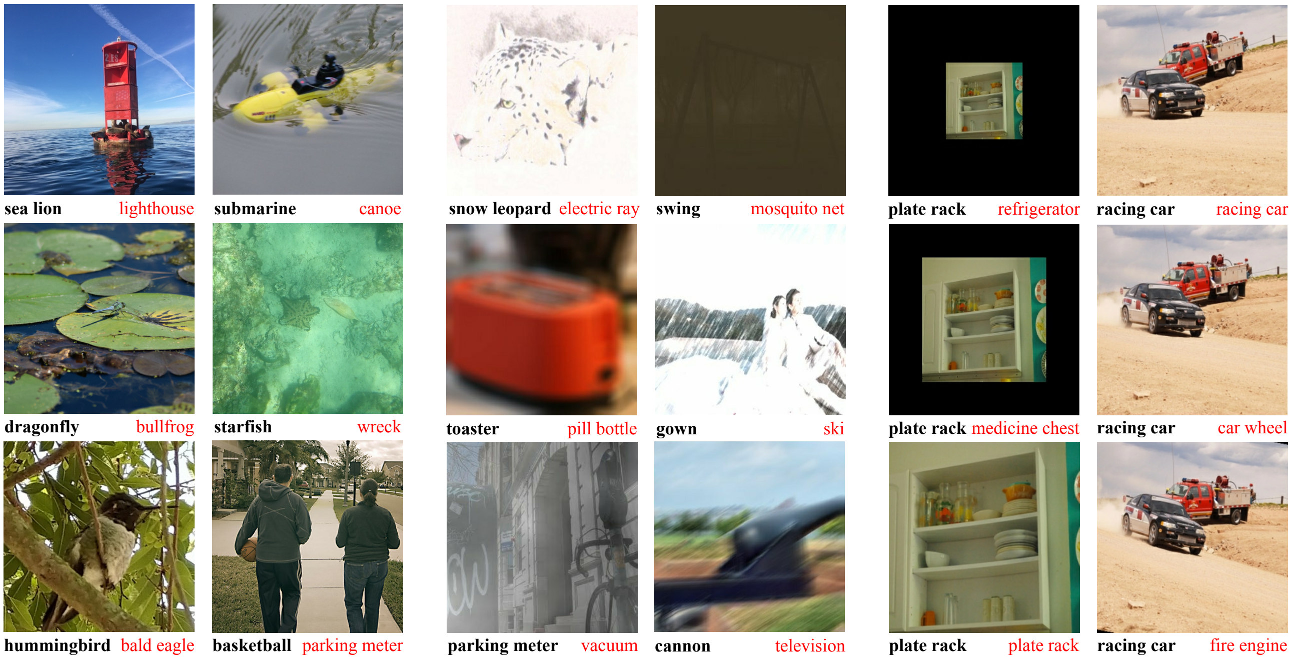
**Qualitative Analysis.** To intuitively understand the significant improvements on the three robustness benchmarks, we show several images in Figure 2 where the predictions of the standard model are incorrect and the predictions of the Noisy Student model are correct.

Figure 2a shows example images from ImageNet-A and the predictions of our models. The model with Noisy Student can successfully predict the correct labels of these highly difficult images. For example, without Noisy Student, 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, the model correctly predicts *dragonfly* for the image. At the top-left image, the model without Noisy Student ignores the *sea lion*s and mistakenly recognizes a buoy as a lighthouse, while the model with Noisy Student can recognize the *sea lion*s.

Figure 2b shows images from ImageNet-C and the corresponding predictions. As can be seen from the figure, our model with Noisy Student makes correct predictions for images under severe corruptions and perturbations such as snow, motion blur and fog, while the model without Noisy Student 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 Noisy Student model still makes the correct prediction.

Figure 2c shows images from ImageNet-P and the corresponding predictions. As can be seen, our model with Noisy Student makes correct and consistent predictions as images undergone different perturbations while the model

[1] 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.



(a) ImageNet-A

(b) ImageNet-C

(c) ImageNet-P

Figure 2: 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. EfficientNet with Noisy Student produces correct top-1 predictions (shown in **bold black** texts) and EfficientNet without Noisy Student produces incorrect top-1 predictions (shown in red texts) on ImageNet-A, C and flips predictions frequently on ImageNet-P.

without Noisy Student flips predictions frequently. For instance, on the right column, as the image of the car undergone a small rotation, the standard model changes its prediction from *racing car* to *car wheel* to *fire engine*. In contrast, the predictions of the model with Noisy Student remain quite stable.

### 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 against an FGSM attack. This attack performs one gradient descent step on the input image [20] with the update on each pixel set to $\epsilon$. As shown in Figure 3, Noisy Student leads to approximately 10% improvement in accuracy even though the model is not optimized for adversarial robustness.

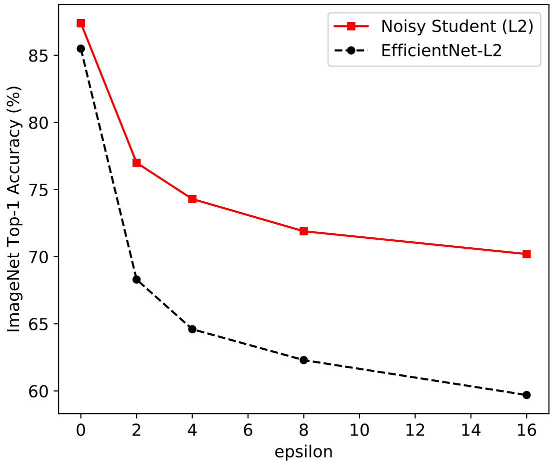


Figure 3: Noisy Student improves adversarial robustness against an FGSM attack though the model is not optimized for adversarial robustness. The accuracy is improved by about 10% in most settings. We use a resolution of 800x800 in this experiment.

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 [17, 20, 19, 61]. Probably due to the same reason, at $\epsilon = 16$, EfficientNet-L2 achieves an accuracy of 1.1% under a stronger attack PGD with 10 iterations [43], which is far from the SOTA results. Noisy Student can still improve the accuracy to 1.6%.

## 4. Ablation Study: The Importance of Noise in Self-training

In this section, we study the importance of noise and the effect of several noise methods used in our model. 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, 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.

| Model / Unlabeled Set Size | 1.3M | 130M |
|---|---|---|
| EfficientNet-B5 | 83.3% | 84.0% |
| Noisy Student (B5) | **83.9%** | **84.9%** |
| w/o Aug | 83.6% | 84.6% |
| w/o Aug, SD, Dropout | 83.2% | 84.3% |

Table 6: Ablation study on noising. We use EfficientNet-B5 as the teacher model and study two cases with different number of unlabeled images and different augmentations. For the experiment with 1.3M unlabeled images, we use 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. Iterative training is not used in these experiments for simplicity.

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. For example, with all noise removed, the accuracy drops from 84.9% to 84.3% in the case with 130M unlabeled images and drops from 83.9% to 83.2% in the case with 1.3M unlabeled images. However, in the case with 130M unlabeled images, with noise function removed, the performance is still improved to 84.3% from 84.0% when compared to the supervised baseline. 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.

## 5. Related works

**Self-training.** Our work is based on self-training (*e.g.*, [59, 79, 56]). 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 [76] which is still far from the state-of-the-art accuracy. They did not show significant improvements in terms of robustness on ImageNet-A, C and P as we did. In terms of methodology, Yalniz *et al.* [76] also proposed to first only train on unlabeled images and then finetune their model on labeled images as the final stage. In Noisy Student, we combine these two steps into one because it simplifies the algorithm and leads to better performance in our preliminary experiments.

Also related to our work is Data Distillation [52], which ensembled predictions for an image with different transformations to teach a student network. The main difference between Data Distillation and our method is that we use the noise to weaken the student, which is the opposite of their approach of strengthening the teacher by ensembling.

Parthasarathi *et al.* [50] used knowledge distillation on unlabeled data to teach a small student model for speech recognition. Their main goal is to find a small and fast model for deployment. As noise injection methods are not used in the student model, and the student model was also small, it is more difficult to make the student better than teacher.

Chowdhury *et al.* [57] used self-training for domain adaptation. Their purpose is different from ours: to adapt a teacher model on one domain to another. Their noise model is video specific and not relevant for image classification.

Their framework is highly optimized for videos, *e.g.*, prediction on which frame to use in a video, which is not as general as our work.

**Semi-supervised Learning.** Apart from self-training, another important line of work in semi-supervised learning [9, 85] is based on consistency training [6, 4, 53, 36, 70, 45, 41, 51, 10, 12, 49, 2, 38, 72, 74, 5, 81]. These works constrain model predictions to be invariant to noise injected to the input, hidden states or model parameters. Although they have produced promising results, in our preliminary experiments, consistency regularization works less well on ImageNet because consistency regularization in the early phase of ImageNet training regularizes the model towards high entropy predictions, and prevents it from achieving good accuracy. A common workaround is to use entropy minimization or ramp up the consistency loss. However, the additional hyperparameters introduced by the ramping up schedule and the entropy minimization make them more difficult to use at scale. Compared to consistency training [45, 5, 74], the self-training / teacher-student framework is better suited for ImageNet because we can train a good teacher on ImageNet using label data.

Works based on pseudo label [37, 31, 60, 1] are similar to self-training, but also suffers the same problem with consistency training, since it relies 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 [84, 73, 77, 33], methods that make use of latent variables as target variables [32, 42, 78] and methods based on low-density separation [21, 58, 15], which might provide complementary benefits to our method.

**Knowledge Distillation.** As we use soft targets, our work is also related to methods in Knowledge Distillation [7, 3, 26, 16]. The main use case 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.* [68, 24, 55, 22], have shown that computer vision models lack robustness. In other words, small changes in the input image can cause large changes to the predictions. 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 similar arguments that using unlabeled data can improve *adversarial* robustness [8, 64, 46, 80]. The main difference between our work and these works is that they directly optimize adversarial robustness on unlabeled data, whereas we show that self-training with Noisy Student improves robustness greatly even without directly optimizing robustness.

## 6. Conclusion

Prior works on weakly-supervised learning require 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 to learn beyond the teacher's knowledge. The method, named self-training with Noisy Student, also benefits from the large capacity of EfficientNet family.

Our experiments showed that self-training with Noisy Student and EfficientNet can achieve an accuracy of 87.4% which is 1.9% higher than without Noisy Student. This result is also a new state-of-the-art and 1% better than the previous best method that used an order of magnitude more weakly labeled data [44, 71].

An important contribution of our work was to show that Noisy Student can potentially help addressing the lack of robustness in computer vision models. Our experiments showed that our model significantly improves accuracy on ImageNet-A, C and P without the need for deliberate data augmentation. For instance, on ImageNet-A, Noisy Student achieves 74.2% top-1 accuracy which is approximately 57% more accurate than the previous state-of-the-art model.

### 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 for robustness evaluation, Guokun Lai, Jiquan Ngiam, Jiateng Xie and Adams Wei Yu for feedbacks on the draft, Yanping Huang and Sameer Kumar for improving TPU implementation, Ekin Dogus Cubuk and Barret Zoph for help with RandAugment, Yanan Bao, Zheyun Feng and Daiyi Peng for help with the JFT dataset, Olga Wichrowska and Ola Spyra 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. 8

[2] Ben Athiwaratkun, Marc Finzi, Pavel Izmailov, and Andrew Gordon Wilson. There are many consistent explana-

tions of unlabeled data: Why you should average. In *International Conference on Learning Representations*, 2018. 8

[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. 8

[4] Philip Bachman, Ouais Alsharif, and Doina Precup. Learning with pseudo-ensembles. In *Advances in Neural Information Processing Systems*, pages 3365–3373, 2014. 8

[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. 8

[6] 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. 8

[7] 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. 8

[8] Yair Carmon, Aditi Raghunathan, Ludwig Schmidt, Percy Liang, and John C Duchi. Unlabeled data improves adversarial robustness. *arXiv preprint arXiv:1905.13736*, 2019. 8

[9] 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. 8

[10] 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. 8

[11] 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. 2, 4

[12] 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. 8

[13] 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

[14] 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, 3

[15] 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. 8

[16] Tommaso Furlanello, Zachary C Lipton, Michael Tschannen, Laurent Itti, and Anima Anandkumar. Born again neural networks. *arXiv preprint arXiv:1805.04770*, 2018. 8

[17] 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. 6

[18] 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

[19] Justin Gilmer, Luke Metz, Fartash Faghri, Samuel S Schoenholz, Maithra Raghu, Martin Wattenberg, and Ian Goodfellow. Adversarial spheres. *arXiv preprint arXiv:1801.02774*, 2018. 6

[20] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*, 2015. 6

[21] Yves Grandvalet and Yoshua Bengio. Semi-supervised learning by entropy minimization. In *Advances in neural information processing systems*, pages 529–536, 2005. 8

[22] Keren Gu, Brandon Yang, Jiquan Ngiam, Quoc Le, and Jonathan Shlens. Using videos to evaluate image model robustness. In *ICLR Workshop*, 2019. 8

[23] 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, 2, 3, 4

[24] Dan Hendrycks and Thomas G Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *International Conference on Learning Representations*, 2019. 1, 4, 5, 8, 13

[25] Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. *arXiv preprint arXiv:1907.07174*, 2019. 1, 4, 5

[26] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. 2, 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*, pages 7132–7141, 2018. 4

[28] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017. 4

[29] Gao Huang, Yu Sun, Zhuang Liu, Daniel Sedra, and Kilian Q Weinberger. Deep networks with stochastic depth. In *European conference on computer vision*, pages 646–661. Springer, 2016. 1, 2, 3

[30] 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. 4

[31] Ahmet Iscen, Giorgos Tolias, Yannis Avrithis, and Ondrej Chum. Label propagation for deep semi-supervised learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5070–5079, 2019. 8

[32] Durk P Kingma, Shakir Mohamed, Danilo Jimenez Rezende, and Max Welling. Semi-supervised learning with deep generative models. In *Advances in neural information processing systems*, pages 3581–3589, 2014. 8

[33] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016. 8

[34] Simon Kornblith, Jonathon Shlens, and Quoc V Le. Do better imagenet models transfer better? In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2661–2671, 2019. 2

[35] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 1097–1105, 2012. 1, 3

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

[37] Dong-Hyun Lee. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on Challenges in Representation Learning, ICML*, volume 3, page 2, 2013. 8

[38] Yingting Li, Lu Liu, and Robby T Tan. Certainty-driven consistency loss for semi-supervised learning. *arXiv preprint arXiv:1901.05657*, 2019. 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)*, pages 19–34, 2018. 4

[40] Raphael Gontijo Lopes, Dong Yin, Ben Poole, Justin Gilmer, and Ekin D Cubuk. Improving robustness without sacrificing accuracy with patch gaussian augmentation. *arXiv preprint arXiv:1906.02611*, 2019. 5

[41] Yucen Luo, Jun Zhu, Mengxi Li, Yong Ren, and Bo Zhang. Smooth neighbors on teacher graphs for semi-supervised learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8896–8905, 2018. 8

[42] Lars Maaløe, Casper Kaae Sønderby, Søren Kaae Sønderby, and Ole Winther. Auxiliary deep generative models. *arXiv preprint arXiv:1602.05473*, 2016. 8

[43] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *International Conference on Learning Representations*, 2018. 7

[44] Dhruv Mahajan, Ross Girshick, Vignesh Ramanathan, Kaiming He, Manohar Paluri, Yixuan Li, Ashwin Bharambe, and Laurens van der Maaten. Exploring the limits of weakly supervised pretraining. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 181–196, 2018. 1, 3, 4, 5, 8

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

[46] Amir Najafi, Shin-ichi Maeda, Masanori Koyama, and Takeru Miyato. Robustness to adversarial perturbations in learning from incomplete data. In *Advances in Neural Information Processing Systems*, 2019. 8

[47] Jiquan Ngiam, Daiyi Peng, Vijay Vasudevan, Simon Kornblith, Quoc V Le, and Ruoming Pang. Domain adaptive transfer learning with specialist models. *arXiv preprint arXiv:1811.07056*, 2018. 3

[48] A Emin Orhan. Robustness properties of facebook's resnext wsl models. *arXiv preprint arXiv:1907.07640*, 2019. 5

[49] Sungrae Park, JunKeon Park, Su-Jin Shin, and Il-Chul Moon. Adversarial dropout for supervised and semi-supervised learning. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018. 8

[50] Sree Hari Krishnan Parthasarathi and Nikko Strom. Lessons from building acoustic models with a million hours of speech. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6670–6674. IEEE, 2019. 7

[51] Siyuan Qiao, Wei Shen, Zhishuai Zhang, Bo Wang, and Alan Yuille. Deep co-training for semi-supervised image recognition. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 135–152, 2018. 8

[52] 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*, pages 4119–4128, 2018. 7

[53] Antti Rasmus, Mathias Berglund, Mikko Honkala, Harri Valpola, and Tapani Raiko. Semi-supervised learning with ladder networks. In *Advances in neural information processing systems*, pages 3546–3554, 2015. 8

[54] 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, pages 4780–4789, 2019. 4

[55] Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do imagenet classifiers generalize to imagenet? *International Conference on Machine Learning*, 2019. 2, 3, 8

[56] Ellen Riloff and Janyce Wiebe. Learning extraction patterns for subjective expressions. In *Proceedings of the 2003 conference on Empirical methods in natural language processing*, pages 105–112, 2003. 7

[57] Aruni Roy Chowdhury, Prithvijit Chakrabarty, Ashish Singh, SouYoung Jin, Huaizu Jiang, Liangliang Cao, and Erik G. Learned-Miller. Automatic adaptation of object detectors to new domains using self-training. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 7

[58] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *Advances in neural information processing systems*, pages 2234–2242, 2016. 8

[59] H Scudder. Probability of error of some adaptive pattern-recognition machines. *IEEE Transactions on Information Theory*, 11(3):363–371, 1965. 1, 2, 7

[60] Weiwei Shi, Yihong Gong, Chris Ding, Zhiheng MaXiaoyu Tao, and Nanning Zheng. Transductive semi-supervised deep learning using min-max features. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 299–315, 2018. 8

[61] Carl-Johann Simon-Gabriel, Yann Ollivier, Leon Bottou, Bernhard Schölkopf, and David Lopez-Paz. First-order adversarial vulnerability of neural networks and input dimension. In *International Conference on Machine Learning*, pages 5809–5817, 2019. 6

[62] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015. 1

[63] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014. 1, 2, 3

[64] Robert Stanforth, Alhussein Fawzi, Pushmeet Kohli, et al. Are labels required for improving adversarial robustness? *arXiv preprint arXiv:1905.13725*, 2019. 8

[65] 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. 4

[66] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015. 1, 3

[67] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016. 4

[68] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013. 8

[69] Mingxing Tan and Quoc V Le. EfficientNet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*, 2019. 1, 2, 3, 4, 12

[70] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *Advances in Neural Information Processing Systems*, pages 1195–1204, 2017. 8

[71] Hugo Touvron, Andrea Vedaldi, Matthijs Douze, and Hervé Jégou. Fixing the train-test resolution discrepancy. *arXiv preprint arXiv:1906.06423*, 2019. 1, 3, 4, 8

[72] Vikas Verma, Alex Lamb, Juho Kannala, Yoshua Bengio, and David Lopez-Paz. Interpolation consistency training for semi-supervised learning. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence (IJCAI-19)*, 2019. 8

[73] Jason Weston, Frédéric Ratle, Hossein Mobahi, and Ronan Collobert. Deep learning via semi-supervised embedding. In *Neural Networks: Tricks of the Trade*, pages 639–655. Springer, 2012. 8

[74] Qizhe Xie, Zihang Dai, Eduard Hovy, Minh-Thang Luong, and Quoc V Le. Unsupervised data augmentation for consistency training. *arXiv preprint arXiv:1904.12848*, 2019. 8

[75] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1492–1500, 2017. 4

[76] 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. 4, 7

[77] Zhilin Yang, William W Cohen, and Ruslan Salakhutdinov. Revisiting semi-supervised learning with graph embeddings. *arXiv preprint arXiv:1603.08861*, 2016. 8

[78] Zhilin Yang, Junjie Hu, Ruslan Salakhutdinov, and William W Cohen. Semi-supervised qa with generative domain-adaptive nets. *arXiv preprint arXiv:1702.02206*, 2017. 8

[79] David Yarowsky. Unsupervised word sense disambiguation rivaling supervised methods. In *33rd annual meeting of the association for computational linguistics*, pages 189–196, 1995. 2, 7

[80] Runtian Zhai, Tianle Cai, Di He, Chen Dan, Kun He, John Hopcroft, and Liwei Wang. Adversarially robust generalization just requires more unlabeled data. *arXiv preprint arXiv:1906.00555*, 2019. 8

[81] Xiaohua Zhai, Avital Oliver, Alexander Kolesnikov, and Lucas Beyer. $S^4$L: Self-supervised semi-supervised learning. In *Proceedings of the IEEE international conference on computer vision*, 2019. 8

[82] Richard Zhang. Making convolutional networks shift-invariant again. In *International Conference on Machine Learning*, 2019. 5

[83] 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*, pages 718–726, 2017. 4

[84] Xiaojin Zhu, Zoubin Ghahramani, and John D Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In *Proceedings of the 20th International conference on Machine learning (ICML-03)*, pages 912–919, 2003. 8

[85] Xiaojin Jerry Zhu. Semi-supervised learning literature survey. Technical report, University of Wisconsin-Madison Department of Computer Sciences, 2005. 8

[86] 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*, pages 8697–8710, 2018. 4

# A. Experiments

## A.1. Architecture Details

The architecture specifications of EfficientNet-L0, L1 and L2 are listed in Table 7. We also list EfficientNet-B7 as a reference. Scaling width and resolution by $c$ leads to $c^2$ times training time and scaling depth by $c$ leads to $c$ times training time. Hence, EfficientNet-L0 has around the same training speed with EfficientNet-B7 but more parameters that give it a larger capacity. EfficientNet-L1 approximately doubles the training time of EfficientNet-L0. Finally, the training time of EfficientNet-L2 is around 2.72 times the training time of EfficientNet-L1.

| Architecture name | $w$ | $d$ | Train Res. | Test Res. | # Params |
|---|---|---|---|---|---|
| EfficientNet-B7 | 2.0 | 3.1 | 600 | 600 | 66M |
| EfficientNet-L0 | 2.8 | 3.7 | 380 | 600 | 140M |
| EfficientNet-L1 | 3.9 | 3.7 | 380 | 600 | 273M |
| EfficientNet-L2 | 4.3 | 5.3 | 475 | 800 | 480M |

Table 7: Architecture specifications for EfficientNet used in the paper. The width $w$ and depth $d$ are the scaling factor that needs to be contextualized in EfficientNet [69]. Train Res. and Test res. denote training and test resolution respectively.

### A.2. Study on Using Out-of-domain Data

Unlike previous studies in semi-supervised learning that use in-domain unlabeled data (*e.g.*, CIFAR-10 images as unlabeled data for a small CIFAR-10 training set), to improve ImageNet, we must use out-of-domain unlabeled data. Here we study how to effectively use out-of-domain data. Since a teacher model's confidence on an image can be a good indicator of whether it is an out-of-domain image, we consider the high-confidence images as in-domain images and the low-confidence images as out-of-domain images. We sample 1.3M images in confidence intervals $[0.0, 0.1], [0.1, 0.2], \cdots, [0.9, 1.0]$.

We use EfficientNet-B0 as both the teacher model and the student model and compare using Noisy Student with soft pseudo labels and hard pseudo labels. The results are shown in Figure 4 with the following observations: (1) Soft pseudo labels and hard pseudo labels can both lead to great improvements with in-domain unlabeled images *i.e.*, high-confidence images. (2) With out-of-domain unlabeled images, hard pseudo labels can hurt the performance while soft pseudo labels leads to robust performance.

We have also observed that using hard pseudo labels can achieve as good results or slightly better results when a larger teacher is used. Hence, whether soft pseudo labels or hard pseudo labels work better might need to be determined on a case-by-case basis.

### A.3. Study on Unlabeled Data Size

We also study the effects of using different amounts of unlabeled data. We start with the 130M unlabeled images and gradually reduce the number of images. For simplicity, we experiment with using $\frac{1}{128}, \frac{1}{64}, \frac{1}{32}, \frac{1}{16}, \frac{1}{4}$ of the whole data by uniformly sampling images from the the unlabeled set though taking the images with highest confidence leads to better results. We use EfficientNet-B4 as both the teacher and the student. As can be seen from Table 8, the performance stays similar when we reduce the data to $\frac{1}{16}$ of the total data, which amounts to 8.1M images after duplicating. The performance drops when we further reduce it. Whether the model benefits from more unlabeled data depends on the capacity of the model since a small model can easily saturate, while a larger model can benefit from more data.

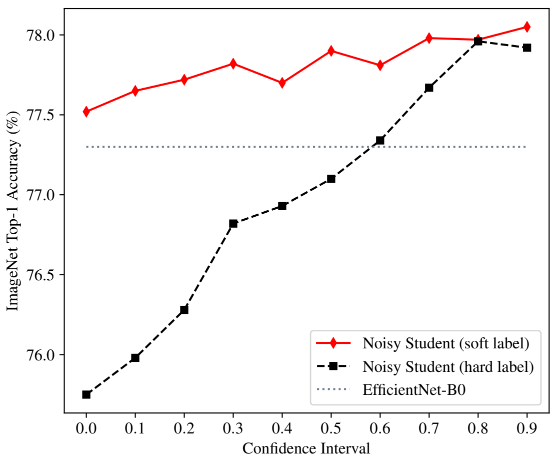


Figure 4: Soft pseudo labels lead to better performance for low confidence data.

| Data Reduction | 1/128 | 1/64 | 1/32 | 1/16 | 1/4 | 1 |
|---|---|---|---|---|---|---|
| Top-1 Acc. | 83.4 | 83.3 | 83.7 | 83.9 | 83.8 | 84.0 |

Table 8: Noisy Student's performance improves with more unlabeled data. The baseline model achieves an accuracy of 83.2.

### A.4. Study on Teacher Model's Capacity

In all previous experiments, the student's capacity is as large as or larger than the capacity of the teacher model. Here we study if it is possible to improve performance on small models by using a larger teacher model, since small models are useful when there are constraints for model size and latency in real-world applications. We use our best model Noisy Student with EfficientNet-L2 to teach student models with sizes ranging from EfficientNet-B0 to EfficientNet-B7. Iterative training is not used here for simplicity. We use the standard augmentation instead of RandAugment in this experiment.

The comparison is shown in Table 9. Using Noisy Student (EfficientNet-L2) as the teacher leads to another 0.8% improvement on top of the improved results. Notably, EfficientNet-B7 achieves an accuracy of 86.8%, which is 1.8% better than the supervised model. This shows that it is helpful to train a large model with high accuracy using Noisy Student when small models are needed for deployment.

| Model | # Params | Top-1 Acc. | Top-5 Acc. |
|---|---|---|---|
| EfficientNet-B0 | 5.3M | 77.3 | 93.4 |
| Noisy Student (B0) | | 78.1 | 94.2 |
| **Noisy Student (B0, L2)** | | **79.0** | **94.6** |
| EfficientNet-B1 | 7.8M | 79.2 | 94.4 |
| Noisy Student (B1) | | 80.3 | 95.1 |
| **Noisy Student (B1, L2)** | | **81.4** | **95.6** |
| EfficientNet-B2 | 9.2M | 80.0 | 94.9 |
| Noisy Student (B2) | | 81.1 | 95.5 |
| **Noisy Student (B2, L2)** | | **82.2** | **96.0** |
| EfficientNet-B3 | 12M | 81.7 | 95.7 |
| Noisy Student (B3) | | 82.4 | 96.2 |
| **Noisy Student (B3, L2)** | | **83.4** | **96.6** |
| EfficientNet-B4 | 19M | 83.2 | 96.4 |
| Noisy Student (B4) | | 84.0 | 96.8 |
| **Noisy Student (B4, L2)** | | **84.9** | **97.2** |
| EfficientNet-B5 | 30M | 84.0 | 96.8 |
| Noisy Student (B5) | | 84.9 | 97.2 |
| **Noisy Student (B5, L2)** | | **85.6** | **97.6** |
| EfficientNet-B6 | 43M | 84.5 | 97.0 |
| Noisy Student (B6) | | 85.3 | 97.5 |
| **Noisy Student (B6, L2)** | | **86.0** | **97.7** |
| EfficientNet-B7 | 66M | 85.0 | 97.2 |
| Noisy Student (B7) | | 85.9 | 97.6 |
| **Noisy Student (B7, L2)** | | **86.8** | **98.0** |

Table 9: Noisy Student (B7) means to use EfficientNet-B7 for both the student and the teacher. Noisy Student (B7, L2) means to use EfficientNet-B7 as the student and use our best model with 87.4% accuracy as the teacher model. For a small student model, using our best model Noisy Student (EfficientNet-L2) as the teacher model leads to more improvements than using the same model as the teacher, which shows that it is helpful to push the performance with our method when small models are needed for deployment.

## A.5. Details for Metrics on Robustness Benchmarks

**ImageNet-A** The top-1 and top-5 accuracy are measured on the 200 classes that ImageNet-A includes. The mapping from the 200 classes to the original ImageNet classes are available online.[2]

**ImageNet-C** mCE (mean corruption error) is the weighted average of error rate on different corruptions, with AlexNet's error rate as a baseline. The score is normalized by AlexNet's error rate so that corruptions with different difficulties lead to scores of a similar scale. Please refer to [24] for details about mCE and AlexNet's error rate. The top-1 accuracy is simply the average top-1 accuracy for all corruptions and all severity degrees. The top-1 accuracy of prior methods are computed from their reported corruption error on each corruption.

**ImageNet-P** Flip probability is the probability that the model changes top-1 prediction for different perturbations. mFR (mean flip rate) is the weighted average of flip probability on different perturbations, with AlexNet's flip probability as a baseline. Please refer to [24] for details about mFR and AlexNet's flip probability. The top-1 accuracy reported in this paper is the average accuracy for all images included in ImageNet-P.

[2] https://github.com/hendrycks/natural-adv-examples/blob/master/eval.py