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 a simple self-training method that 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.

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 accurate as possible. However, 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. Introduction

Deep learning has shown remarkable successes in image recognition in recent years [42, 75, 70, 28, 78]. 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 (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 using the self-training framework [66] 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.

1965年

3 main steps

Our experiments show that an important element for this 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 accurate as possible, and the noised student is forced to learn harder from the pseudo labels. To noise the student, we use RandAugment data augmentation [17], dropout [71] and stochastic depth [35] during its training. We call the method self-training with Noisy Student to emphasize the role that noise plays in the method and results.

△

Using self-training with Noisy Student, together with 300M unlabeled images, we improve EfficientNet's [78] ImageNet top-1 accuracy to 88.4%. This accuracy is 2.0% 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 [30] top-1 accuracy from 61.0% to 83.7%, ImageNet-C [29] mean corruption error (mCE) from 45.7 to 28.3 and ImageNet-P [29] 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 [80, 51]. 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 for 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. 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 $\theta^t_*$ 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 an unnoised teacher model to generate soft or hard
   pseudo labels for unlabeled images
       $\tilde{y}_i = f(\tilde{x}_i, \theta^t_*), \forall i = 1, \cdots, m$
3: Learn an **equal-or-larger** student model $\theta^s_*$ 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 method.

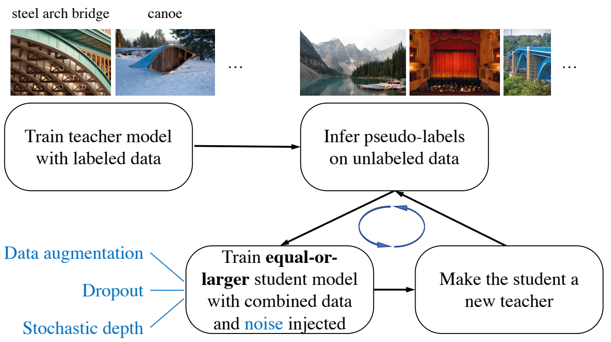

steel arch bridge
canoe
...
...
Train teacher model with labeled data
Infer pseudo-labels on unlabeled data
Data augmentation
Dropout
Stochastic depth
Train **equal-or-larger** student model with combined data and noise injected
Make the student a new teacher

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

The algorithm is fundamentally self-training, a standard method in semi-supervised learning (*e.g.*, [66, 89]). More discussions on how our method is related to prior works are included in Section 5. Our key differences lie in adding more sources of noise to the student and using student models that are as large as (if not larger than) the teacher. This makes our method different from Knowledge Distillation [31] where adding noise is not the core concern and small model is often used as a student 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 more capacity and difficult environments in terms of noise to learn through.

Noising Student – When the student is deliberately noised it is actually trained to be consistent to the more powerful 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 [17]. For model noise, we use dropout [71] and stochastic depth [35].

When applied to unlabeled data, noise has a compound benefit of enforcing local smoothness in the decision function on both labeled and unlabeled data. Different kinds of noise have different effects. With data augmentation noise, the student must ensure that an image, when translated for example, should have the same category as a non-translated image. This invariant encourages the student model to learn beyond the teacher to make predictions with more difficult images. When dropout and stochastic depth function are used as noise, the teacher behaves like an ensemble at inference time (during which 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 also works better with an additional trick: data filtering and balancing. Specifically, we filter images that the teacher model has low confidences on since they are usually out-of-domain images. 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. 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. [1]

Finally, in the above, we say that the pseudo labels can be soft or hard. We observe that both soft and hard pseudo labels work well in our experiments. In particular, soft pseudo labels work slightly better for out of domain unlabeled data. Thus in the following, for consistency, we report results

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

with soft pseudo labels unless otherwise indicated.

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 [41, 62].

Unlabeled dataset. We obtain unlabeled images from the JFT dataset [31, 14], 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 [54]).

We then perform data filtering and balancing on this corpus. First, we run an EfficientNet-B0 trained on ImageNet [78] over the JFT dataset [31, 14] 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 [78] 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 available unlabeled images. Labeled images and unlabeled images are concatenated together to compute the average cross entropy loss.

Lastly, we apply the recently proposed technique to fix train-test resolution discrepancy [80] for EfficientNet-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 [80], 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.

Noise. We use stochastic depth [35], dropout [71], and RandAugment [17] 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 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 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 [42, 75, 28, 78] (see also [62]). As shown in Table 2, Noisy Student with EfficientNet-L2 achieves 88.4% top-1 accuracy which is significantly better than the best reported accuracy on EfficientNet of 85.0%.

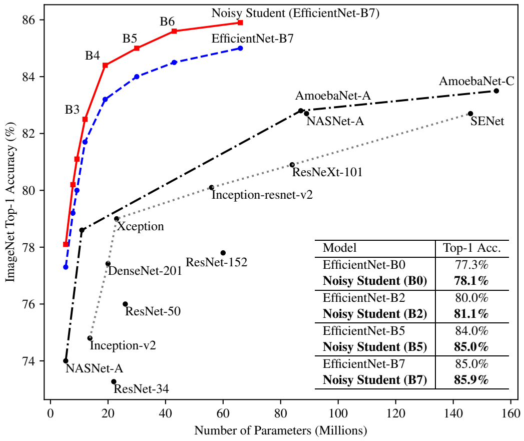
Method	# Params	Extra Data	Top-1 Acc.	Top-5 Acc.
ResNet-50 [28]	26M	-	76.0%	93.0%
ResNet-152 [28]	60M	-	77.8%	93.8%
DenseNet-264 [34]	34M	-	77.9%	93.9%
Inception-v3 [76]	24M	-	78.8%	94.4%
Xception [14]	23M	-	79.0%	94.5%
Inception-v4 [74]	48M	-	80.0%	95.0%
Inception-resnet-v2 [74]	56M	-	80.1%	95.1%
ResNeXt-101 [85]	84M	-	80.9%	95.6%
PolyNet [93]	92M	-	81.3%	95.8%
SENet [33]	146M	-	82.7%	96.2%
NASNet-A [97]	89M	-	82.7%	96.2%
AmoebaNet-A [61]	87M	-	82.8%	96.1%
PNASNet [46]	86M	-	82.9%	96.2%
AmoebaNet-C [16]	155M	-	83.5%	96.5%
GPipe [36]	557M	-	84.3%	97.0%
EfficientNet-B7 [78]	66M	-	85.0%	97.2%
EfficientNet-L2 [78]	480M	-	85.5%	97.5%
ResNet-50 Billion-scale [86]	26M	3.5B images labeled with tags	81.2%	96.0%
ResNeXt-101 Billion-scale [86]	193M		84.8%	-
ResNeXt-101 WSL [51]	829M		85.4%	97.6%
FixRes ResNeXt-101 WSL [80]	829M		86.4%	98.0%
Noisy Student (L2)	480M	300M unlabeled images	**88.4%**	**98.7%**

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.

The total gain of 3.4% comes from two sources: by making the model larger (+0.5%) and by Noisy Student (+2.9%). In other words, using Noisy Student makes a much larger impact on the accuracy than changing the architecture.

Further, Noisy Student outperforms the state-of-the-art accuracy of 86.4% by FixRes ResNeXt-101 WSL [51, 80] 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 for EfficientNet B0-B7 without Iterative Training. In addition to improving state-of-the-art results, we conduct experiments to verify if Noisy Student 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 [78] 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 leads to a consistent improvement of around 0.8% for all model sizes. Over-



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)	**85.0%**
EfficientNet-B7	85.0%
Noisy Student (B7)	**85.9%**

Figure 2: 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.

all, EfficientNets with Noisy Student 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 even without iterative training.

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

Method	Top-1 Acc.	Top-5 Acc.
ResNet-101 [30]	4.7%	-
ResNeXt-101 [30] (32x4d)	5.9%	-
ResNet-152 [30]	6.1%	-
ResNeXt-101 [30] (64x4d)	7.3%	-
DPN-98 [30]	9.4%	-
ResNeXt-101+SE [30] (32x4d)	14.2%	-
ResNeXt-101 WSL [51, 55]	61.0%	-
EfficientNet-L2	49.6%	78.6%
Noisy Student (L2)	**83.7%**	**95.2%**

Table 3: Robustness results on ImageNet-A.

Method	Res.	Top-1 Acc.	mCE
ResNet-50 [29]	224	39.0%	76.7
SIN [22]	224	45.2%	69.3
Patch Gaussian [47]	299	52.3%	60.4
ResNeXt-101 WSL [51, 55]	224	-	45.7
EfficientNet-L2	224	62.6%	47.5
Noisy Student (L2)	224	76.5%	30.0
EfficientNet-L2	299	66.6%	42.5
Noisy Student (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 [29]	224	-	58.0
Low Pass Filter Pooling [92]	224	-	51.2
ResNeXt-101 WSL [51, 55]	224	-	27.8
EfficientNet-L2	224	80.4%	27.2
Noisy Student (L2)	224	85.2%	14.2
EfficientNet-L2	299	81.6%	23.7
Noisy Student (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).

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 [29] include images with common corruptions and perturbations such as blurring, fogging, rotation and scaling. ImageNet-A test set [30] 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 models on two released versions with resolution 224x224 and 299x299 and resize images to the resolution EfficientNet trained on. As shown in Table 3, 4 and 5, Noisy Student yields substantial gains on robustness datasets compared to the previous state-of-the-art model ResNeXt-101 WSL [51, 55] 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.[2] These significant gains in robustness in ImageNet-C and ImageNet-P are surprising because our method was not deliberately optimized for robustness.[3]

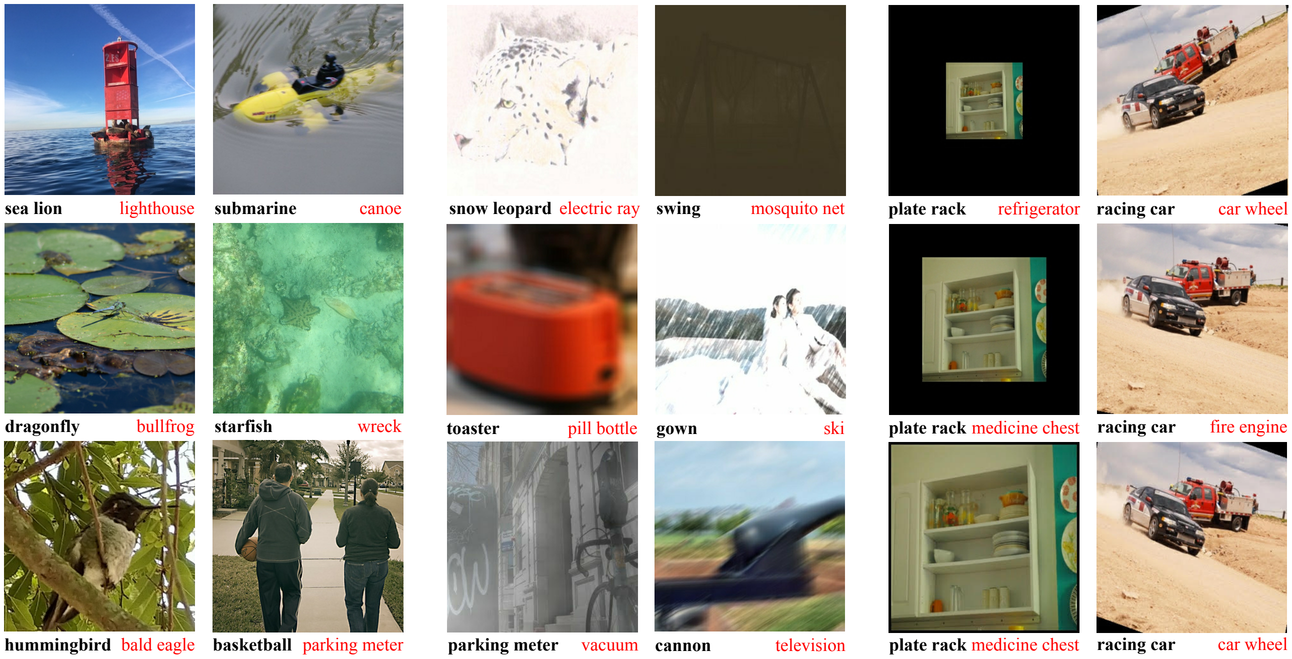
Qualitative Analysis. To intuitively understand the significant improvements on the three robustness benchmarks, we show several images in Figure 3 where the predictions of the standard model are incorrect while the predictions of the Noisy Student model are correct.

Figure 3a 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 Noisy Student model 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 makes correct predictions for images under severe corruptions and perturbations such as

[2] 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.

[3] 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.



(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 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-P, EfficientNet without Noisy Student flips predictions frequently.

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 3c 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 without Noisy Student 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 against an FGSM attack. This attack performs one gradient descent step on the input image [24] with the update on each pixel set to ϵ. As shown in Figure 4, Noisy Student 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 [50], at $\epsilon = 16$, Noisy Student improves EfficientNet-L2's accuracy from 1.1% to 4.4%.

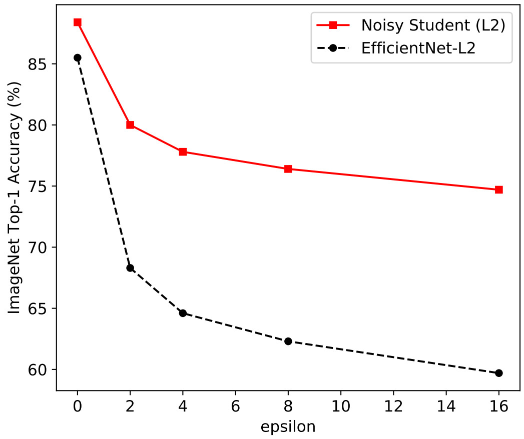


Figure 4: Noisy Student improves adversarial robustness against an FGSM attack though the model is not optimized for adversarial robustness. The accuracy is improved by 11% at $\epsilon = 2$ and gets better as ϵ gets larger.

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 [21, 24, 23, 69].

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.

Model / Unlabeled Set Size	1.3M	130M
EfficientNet-B5	83.3%	84.0%
Noisy Student (B5)	**83.9%**	**84.9%**
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.

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 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	1:14	87.6%
2	EfficientNet-L2	1:14	88.1%
3	EfficientNet-L2	1:28	88.4%

Table 7: Iterative training improves the accuracy, where batch size ratio denotes the ratio between unlabeled data and labeled data. In the first iteration, we use EfficientNet-B7 trained on ImageNet that has accuracy 85.0% as the teacher. In the latter iterations, we use the model of the previous iteration as the teacher.

4.3. Additional Ablation Study Summarization

We also study the importance of various design choices of Noisy Student, 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*., [66, 89, 63]). 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 [86] which is still far from the state-of-the-art accuracy. Yalniz *et al*. [86] 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, we combine these two steps into one because it simplifies the algorithm and leads to better performance in our experiments.

Data Distillation [59], 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*. [57] 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 [64] is related but highly optimized for videos, *e.g*., prediction on which frame to use in a video. The method in [94] ensembles predictions from multiple teacher models, which is more expensive than our method.

Co-training [8] 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 [81], semantic segmentation [4], text classification [38, 73]. Back translation has led to significant improvements in machine translation [67, 19, 27, 13].

Semi-supervised Learning. Apart from self-training, another important line of work in semi-supervised learning [11, 96] is based on consistency training [5, 60, 43, 79, 52, 48, 58, 12, 15, 56, 2, 45, 82, 84, 7, 91]. 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 to 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 [52, 7, 84], the self-training / teacher-student framework is better suited for ImageNet because we can train a good teacher on ImageNet using labeled data.

Works based on pseudo label [44, 37, 68, 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 [95, 83, 87, 40], methods that make use of latent variables as target variables [39, 49, 88] and methods based on low-density separation [25, 65, 18], 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 [9, 3, 31, 20, 6]. 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.* [77, 29, 62, 26], 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 [10, 72, 53, 90]. 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 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, to learn beyond the teacher's knowledge.

Our experiments showed that self-training with Noisy Student and EfficientNet can achieve an accuracy of 88.4% which is 2.9% higher than without Noisy Student. 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 [51, 80].

An important contribution of our work was to show that Noisy Student 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, 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. 8

[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. 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. 9

[4] Yauhen Babakhin, Artsiom Sanakoyeu, and Hiroshi Kitamura. Semi-supervised segmentation of salt bodies in seismic images using an ensemble of convolutional neural networks. *arXiv preprint arXiv:1904.04445*, 2019. 8

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

[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, 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

[8] 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

[9] 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

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

[11] 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

[12] 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

[13] 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. 8

[14] François Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE con-*

ference on computer vision and pattern recognition, pages 1251–1258, 2017. 3, 4

[15] 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

[16] 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

[17] 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

[18] 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

[19] 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. 8

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

[21] 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

[22] 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

[23] 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

[24] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*, 2015. 6, 7

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

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

[27] 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. 8

[28] 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, 3, 4

[29] Dan Hendrycks and Thomas G Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *International Conference on Learning Representations*, 2019. 1, 5, 9, 16

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

[31] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. 2, 3, 9

[32] Andrew G Howard. Some improvements on deep convolutional neural network based image classification. *arXiv preprint arXiv:1312.5402*, 2013. 16

[33] 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

[34] 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

[35] 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

[36] 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

[37] 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

[38] Giannis Karamanolakis, Daniel Hsu, and Luis Gravano. Leveraging just a few keywords for fine-grained aspect detection through weakly supervised co-training. *Empirical Methods in Natural Language Processing (EMNLP)*, 2019. 8

[39] 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

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

[41] 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. 3

[42] 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

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

[44] 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

[45] Yingting Li, Lu Liu, and Robby T Tan. Certainty-driven consistency loss for semi-supervised learning. *arXiv preprint arXiv:1901.05657*, 2019. 8

[46] 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

[47] 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

[48] 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

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

[50] 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. 6

[51] 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, 4, 5, 9, 16

[52] 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

[53] 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. 9

[54] 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

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

[56] 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

[57] 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. 8

[58] 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

[59] 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. 8

[60] 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

[61] 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

[62] Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do imagenet classifiers generalize to imagenet? *International Conference on Machine Learning*, 2019. 3, 9

[63] 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. 8

[64] 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. 8

[65] 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. 9

[66] H Scudder. Probability of error of some adaptive pattern-recognition machines. *IEEE Transactions on Information Theory*, 11(3):363–371, 1965. 1, 2, 8

[67] Rico Sennrich, Barry Haddow, and Alexandra Birch. Improving neural machine translation models with monolingual data. *arXiv preprint arXiv:1511.06709*, 2015. 8

[68] 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

[69] 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. 7

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

[71] 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

[72] Robert Stanforth, Alhussein Fawzi, Pushmeet Kohli, et al. Are labels required for improving adversarial robustness? *arXiv preprint arXiv:1905.13725*, 2019. 9

[73] Qianru Sun, Xinzhe Li, Yaoyao Liu, Shibao Zheng, Tat-Seng Chua, and Bernt Schiele. Learning to self-train for semi-supervised few-shot classification. *arXiv preprint arXiv:1906.00562*, 2019. 8

[74] 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

[75] 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, 16

[76] 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

[77] 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. 9

[78] Mingxing Tan and Quoc V Le. EfficientNet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*, 2019. 1, 3, 4, 13

[79] 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

[80] Hugo Touvron, Andrea Vedaldi, Matthijs Douze, and Hervé Jégou. Fixing the train-test resolution discrepancy. *arXiv preprint arXiv:1906.06423*, 2019. 1, 3, 4, 9

[81] Andreas Veit, Neil Alldrin, Gal Chechik, Ivan Krasin, Abhinav Gupta, and Serge Belongie. Learning from noisy large-scale datasets with minimal supervision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 839–847, 2017. 8

[82] 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

[83] 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

[84] 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

[85] 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

[86] 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, 8, 14

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

[88] 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

[89] David Yarowsky. Unsupervised word sense disambiguation rivaling supervised methods. In *33rd annual meeting of the association for computational linguistics*, pages 189–196, 1995. 2, 8

[90] 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. 9

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

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

[93] 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

[94] Giulio Zhou, Subramanya Dulloor, David G Andersen, and Michael Kaminsky. Edf: Ensemble, distill, and fuse for easy video labeling. *arXiv preprint arXiv:1812.03626*, 2018. 8

[95] 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

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

[97] 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-L2 are listed in Table 8. We also list EfficientNet-B7 as a reference. Scaling width and resolution by c leads to an increase factor of c^2 in training time and scaling depth by c leads to an increase factor of c. The training time of EfficientNet-L2 is around 5 times the training time of EfficientNet-B7.

Architecture Name	w	d	Train Res.	Test Res.	# Params
EfficientNet-B7	2.0	3.1	600	600	66M
EfficientNet-L2	4.3	5.3	475	800	480M

Table 8: Architecture specifications for EfficientNets used in the paper. The width w and depth d are the scaling factors that need to be contextualized in EfficientNet [78]. Train Res. and Test Res. denote training and testing resolutions respectively.

A.2. Ablation Studies

In this section, we provide comprehensive studies of various components of our method. Since iterative training results in longer training time, we conduct ablation without it. To further save training time, we reduce the training epochs for small models from 700 to 350, starting from Study #4. We also set the unlabeled batch size to be the same as the labeled batch size for models smaller than EfficientNet-B7 starting from Study #2.

Study #1: Teacher Model's Capacity. Here, we study if using a larger and better teacher model would lead to better results. We use our best model Noisy Student with EfficientNet-L2, that achieves a top-1 accuracy of 88.4%, to teach student models with sizes ranging from EfficientNet-B0 to EfficientNet-B7. We use the standard augmentation instead of RandAugment on unlabeled data in this experiment to give the student model more capacity. This setting is in principle similar to distillation on unlabeled data.

The comparison is shown in Table 9. Using Noisy Student (EfficientNet-L2) as the teacher leads to another 0.7% to 1.6% improvement on top of the improved results by using the same model as the teacher. For example, we can train a medium-sized model EfficientNet-B4, which has fewer parameters than ResNet-50, to an accuracy of 85.3%. Therefore, *using a large teacher model with better performance leads to better results.*

Study #2: Unlabeled Data Size. Next, we conduct experiments to understand the effects of using different amounts of unlabeled data. We start with the 130M unlabeled images and gradually reduce the unlabeled set. 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 for simplicity, though taking images with highest confidence may lead to better results. We use EfficientNet-B4 as both the teacher and the student.

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)		**78.8%**	**94.5%**
EfficientNet-B1	7.8M	79.2%	94.4%
Noisy Student (B1)		80.2%	95.2%
Noisy Student (B1, L2)		**81.5%**	**95.8%**
EfficientNet-B2	9.2M	80.0%	94.9%
Noisy Student (B2)		81.1%	95.5%
Noisy Student (B2, L2)		**82.4%**	**96.3%**
EfficientNet-B3	12M	81.7%	95.7%
Noisy Student (B3)		82.5%	96.4%
Noisy Student (B3, L2)		**84.1%**	**96.9%**
EfficientNet-B4	19M	83.2%	96.4%
Noisy Student (B4)		84.4%	97.0%
Noisy Student (B4, L2)		**85.3%**	**97.5%**
EfficientNet-B5	30M	84.0%	96.8%
Noisy Student (B5)		85.0%	97.2%
Noisy Student (B5, L2)		**86.1%**	**97.8%**
EfficientNet-B6	43M	84.5%	97.0%
Noisy Student (B6)		85.6%	97.6%
Noisy Student (B6, L2)		**86.4%**	**97.9%**
EfficientNet-B7	66M	85.0%	97.2%
Noisy Student (B7)		85.9%	97.6%
Noisy Student (B7, L2)		**86.9%**	**98.1%**

Table 9: Using our best model with 88.4% accuracy as the teacher (denoted as Noisy Student (X, L2)) leads to more improvements than using the same model as the teacher (denoted as Noisy Student (X)). Models smaller than EfficientNet-B5 are trained for 700 epochs (better than training for 350 epochs as used in Study #4 to Study #8). Models other than EfficientNet-B0 uses an unlabeled batch size of three times the labeled batch size, while other ablation studies set the unlabeled batch size to be the same as labeled batch size by default for models smaller than B7.

As can be seen from Table 10, the performance stays similar when we reduce the data to $\frac{1}{16}$ of the whole data,[4] which amounts to 8.1M images after duplicating. The performance drops when we further reduce it. Hence, *using a large amount of unlabeled data leads to better performance*.

Data	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 10: Noisy Student's performance improves with more unlabeled data. Models are trained for 700 epochs without iterative training. The baseline model achieves an accuracy of 83.2%.

[4] A larger model might benefit from more data while a small model with limited capacity can easily saturate.

Study #3: Hard Pseudo-Label vs. Soft Pseudo-Label on 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 compare hard pseudo-label and soft pseudo-label for 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 each confidence interval $[0.0, 0.1], [0.1, 0.2], \cdots, [0.9, 1.0]$.

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 5 with the following observations: *(1) Soft pseudo labels and hard pseudo labels can both lead to significant 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 lead to robust performance.*

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

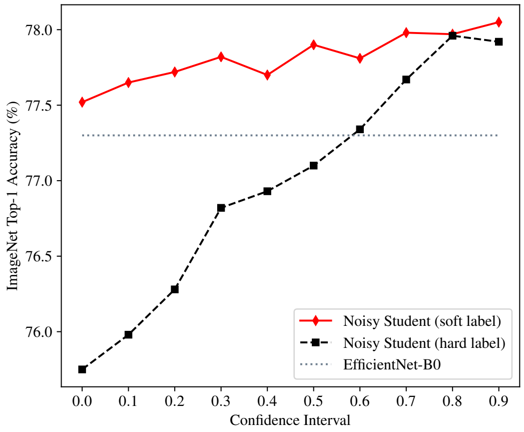


Figure 5: Soft pseudo labels lead to better performance for low confidence data (out-of-domain data). Each dot at p represents a NoisyStudent model trained with 1.3M ImageNet labeled images and 1.3M unlabeled images with confidence scores in $[p, p + 0.1]$.

Study #4: Student Model's Capacity. Then, we investigate the effects of student models with different capacities. For teacher models, we use EfficientNet-B0, B2 and B4 trained on labeled data and EfficientNet-B7 trained using Noisy Student. We compare using a student model with the same size or with a larger size. The comparison is shown in Table 11. With the same teacher, using a larger student model leads to consistently better performance, showing that *using a large student model is important to enable the student to learn a more powerful model.*

Teacher	Teacher Acc.	Student	Student Acc.
B0	77.3%	B0	77.9%
		B1	**79.5%**
B2	80.0%	B2	80.7%
		B3	**82.0%**
B4	83.2%	B4	84.0%
		B5	**84.7%**
B7	86.9%	B7	86.9%
		L2	**87.2%**

Table 11: Using a larger student model leads to better performance. Student models are trained for 350 epochs instead of 700 epochs without iterative training. The B7 teacher with an accuracy of 86.9% is trained by Noisy Student with multiple iterations using B7. The comparison between B7 and L2 as student models is not completely fair for L2, since we use an unlabeled batch size of 3x the labeled batch size for training L2, which is not as good as using an unlabeled batch size of 7x the labeled batch size when training B7 (See Study #7 for more details).

Study #5: Data Balancing. Here, we study the necessity of keeping the unlabeled data balanced across categories. As a comparison, we use all unlabeled data that has a confidence score higher than 0.3. We present results with EfficientNet-B0 to B3 as the backbone models in Table 12. Using data balancing leads to better performance for small models EfficientNet-B0 and B1. Interestingly, the gap becomes smaller for larger models such as EfficientNet-B2 and B3, which shows that more powerful models can learn from unbalanced data effectively. *To enable Noisy Student to work well for all model sizes, we use data balancing by default.*

Model	B0	B1	B2	B3
Supervised Learning	77.3%	79.2%	80.0%	81.7%
Noisy Student	**77.9%**	**79.9%**	**80.7%**	82.1%
w/o Data Balancing	77.6%	79.6%	80.6%	82.1%

Table 12: Data balancing leads to better results for small models. Models are trained for 350 epochs instead of 700 epochs without iterative training.

Study #6: Joint Training. In our algorithm, we train the model with labeled images and pseudo-labeled images jointly. Here, we also compare with an alternative approach used by Yalniz *et al.* [86], which first pretrains the model on pseudo-labeled images and then finetunes it on labeled images. For finetuning, we experiment with different steps and take the best results. The comparison is shown in Table 13.

It is clear that joint training significantly outperforms pretraining + finetuning. Note that pretraining only on pseudo-labeled images leads to a much lower accuracy than supervised learning only on labeled data, which suggests that the distribution of unlabeled data is very different from that of labeled data. *In this case, joint training leads to a better solution that fits both types of data.*

Model	B0	B1	B2	B3
Supervised Learning	77.3%	79.2%	80.0%	81.7%
Pretraining	72.6%	75.1%	75.9%	76.5%
Pretraining + Finetuning	77.5%	79.4%	80.3%	81.7%
Joint Training	**77.9%**	**79.9%**	**80.7%**	**82.1%**

Table 13: Joint training works better than pretraining and finetuning. We vary the finetuning steps and report the best results. Models are trained for 350 epochs instead of 700 epochs without iterative training.

Study #7: Ratio between Unlabeled Batch Size and Labeled Batch Size. Since we use 130M unlabeled images and 1.3M labeled images, if the batch sizes for unlabeled data and labeled data are the same, the model is trained on unlabeled data only for one epoch every time it is trained on labeled data for a hundred epochs. Ideally, we would also like the model to be trained on unlabeled data for more epochs by using a larger unlabeled batch size so that it can fit the unlabeled data better. Hence we study the importance of the ratio between unlabeled batch size and labeled batch size.

Teacher (Acc.)	Batch Size Ratio	Top-1 Acc.
B4 (83.2)	1:1	84.0%
	3:1	84.0%
L2 (87.0)	1:1	86.7%
	3:1	**87.4%**
L2 (87.4)	3:1	87.4%
	6:1	**87.9%**

Table 14: With a fixed labeled batch size, a larger unlabeled batch size leads to better performance for EfficientNet-L2. The Batch Size Ratio denotes the ratio between unlabeled batch size and labeled batch size.

In this study, we try a medium-sized model EfficientNet-B4 as well as a larger model EfficientNet-L2. We use models of the same size as both the teacher and the student. As shown in Table 14, the larger model EfficientNet-L2 benefits from a large ratio while the smaller model EfficientNet-B4 does not. *Using a larger ratio between unlabeled batch size and labeled batch size, leads to substantially better performance for a large model.*

Study #8: Warm-starting the Student Model. Lastly, one might wonder if we should train the student model from scratch when it can be initialized with a converged teacher model with good accuracy. In this ablation, we first train an EfficientNet-B0 model on ImageNet and use it to initialize the student model. We vary the number of epochs for training the student and use the same exponential decay learning rate schedule. Training starts at different learning rates so that the learning rate is decayed to the same value in all experiments. As shown in Table 15, the accuracy drops significantly when we reduce the training epoch from 350 to 70 and drops slightly when reduced to 280 or 140. Hence, the student still needs to be trained for a large number of epochs even with warm-starting.

Further, we also observe that a student initialized with the teacher can sometimes be stuck in a local optimal. For example, when we use EfficientNet-B7 with an accuracy of 86.4% as the teacher, the student model initialized with the teacher achieves an accuracy of 86.4% halfway through the training but gets stuck there when trained for 210 epochs, while a model trained from scratch achieves an accuracy of 86.9%. Hence, though we can save training time by warm-staring, *we train our model from scratch to ensure the best performance.*

Warm-start	Initializing student with teacher				No Init
Epoch	35	70	140	280	350
Top-1 Acc.	77.4%	77.5%	77.7%	77.8%	**77.9%**

Table 15: A student initialized with the teacher still requires at least 140 epochs to perform well. The baseline model, trained with labeled data only, has an accuracy of 77.3%.

A.3. Details of Robustness Benchmarks

Metrics. For completeness, we provide brief descriptions of metrics used in robustness benchmarks ImageNet-A, ImageNet-C and ImageNet-P.

- **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.[5]

[5] https://github.com/hendrycks/natural-adv-

- **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 [29] 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 [29] 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.

On Using RandAugment for ImageNet-C and ImageNet-P. Since Noisy Student leads to significant improvements on ImageNet-C and ImageNet-P, we briefly discuss the influence of RandAugment on robustness results. First, note that our supervised baseline EfficientNet-L2 also uses RandAugment. Noisy Student leads to significant improvements when compared to the supervised baseline as shown in Table 4 and Table 5.

Second, the overlap between transformations of RandAugment and ImageNet-C, P is small. For completeness, we list transformations in RandAugment and corruptions and perturbations in ImageNet-C and ImageNet-P here:

- RandAugment transformations: AutoContrast, Equalize, Invert, Rotate, Posterize, Solarize, Color, Contrast, Brightness, Sharpness, ShearX, ShearY, TranslateX and TranslateY.

- Corruptions in ImageNet-C: Gaussian Noise, Shot Noise, Impulse Noise, Defocus Blur, Frosted Glass Blur, Motion Blur, Zoom Blur, Snow, Frost, Fog, Brightness, Contrast, Elastic, Pixelate, JPEG.

- Perturbations in ImageNet-P: Gaussian Noise, Shot Noise, Motion Blur, Zoom Blur, Snow, Brightness, Translate, Rotate, Tilt, Scale.

The main overlap between RandAugment and ImageNet-C are Contrast, Brightness and Sharpness. Among them, augmentation Contrast and Brightness are also used in ResNeXt-101 WSL [51] and in vision models that uses the Inception preprocessing [32, 75]. The overlap between RandAugment and ImageNet-P includes Brightness, Translate and Rotate.

examples/blob/master/eval.py