A. Experiments

A.1. Architecture Details

The architecture specifications of EfficientNet-L2 are listed in Table 7. 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 7: 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 [82]. Train Res. and Test Res. denote training and testing resolutions respectively.

A.2. Details 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 8, the model performance improves to 87.6% in the first iteration and then to 88.1% in the second iteration with the same hyperparameters (except using a teacher model with better performance). These results indicate that iterative training is effective in producing increasingly better models. For the last iteration, we make use of a larger ratio between unlabeled batch size and labeled batch size to boost the final performance to 88.4%.

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

Table 8: Iterative training improves the accuracy, where batch size ratio denotes the ratio between unlabeled data and labeled data.

A.3. Ablation Studies

We also study the importance of various design choices of NoisyStudent, hopefully offering a practical guide for readers. With this purpose, we conduct 8 ablation studies. 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.

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 NoisyStudent 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 NoisyStudent (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

Model	# Params	Top-1 Acc.	Top-5 Acc.
EfficientNet-B0	5.3M	77.3%	93.4%
NoisyStudent (B0)		78.1%	94.2%
NoisyStudent (B0, L2)		**78.8%**	**94.5%**
EfficientNet-B1	7.8M	79.2%	94.4%
NoisyStudent (B1)		80.2%	95.2%
NoisyStudent (B1, L2)		**81.5%**	**95.8%**
EfficientNet-B2	9.2M	80.0%	94.9%
NoisyStudent (B2)		81.1%	95.5%
NoisyStudent (B2, L2)		**82.4%**	**96.3%**
EfficientNet-B3	12M	81.7%	95.7%
NoisyStudent (B3)		82.5%	96.4%
NoisyStudent (B3, L2)		**84.1%**	**96.9%**
EfficientNet-B4	19M	83.2%	96.4%
NoisyStudent (B4)		84.4%	97.0%
NoisyStudent (B4, L2)		**85.3%**	**97.5%**
EfficientNet-B5	30M	84.0%	96.8%
NoisyStudent (B5)		85.0%	97.2%
NoisyStudent (B5, L2)		**86.1%**	**97.8%**
EfficientNet-B6	43M	84.5%	97.0%
NoisyStudent (B6)		85.6%	97.6%
NoisyStudent (B6, L2)		**86.4%**	**97.9%**
EfficientNet-B7	66M	85.0%	97.2%
NoisyStudent (B7)		85.9%	97.6%
NoisyStudent (B7, L2)		**86.9%**	**98.1%**

Table 9: Using our best model with 88.4% accuracy as the teacher (denoted as NoisyStudent (X, L2)) leads to more improvements than using the same model as the teacher (denoted as NoisyStudent (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.

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.

As can be seen from Table 10, the performance stays similar when we reduce the data to $\frac{1}{16}$ of the whole data,[5] 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*.

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.*,

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

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: NoisyStudent'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%.

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 NoisyStudent 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.*

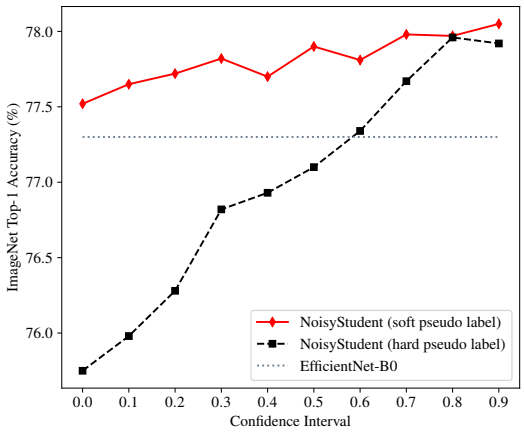

Figure 5: Soft pseudo labels lead to better performance for low confidence data (out-of-domain data). Each dot at p represents a NoisyStudent model trained with 1.3M ImageNet labeled images and 1.3M unlabeled images with confidence scores in $[p, p + 0.1]$.

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.

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 NoisyStudent. 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 NoisyStudent 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 NoisyStudent to work well for all model sizes, we use data balancing by default.*

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*. [91], 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.

Model	B0	B1	B2	B3
Supervised Learning	77.3%	79.2%	80.0%	81.7%
NoisyStudent	**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.

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.

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.*

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.

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 Epoch	Initializing student with teacher 35	70	140	280	No Init 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.4. Results with a Different Architecture and Dataset

Results with ResNet-50. To study whether other architectures can benefit from NoisyStudent, we conduct experiments with ResNet-50 [30]. We use the full ImageNet as the labeled data and the 130M images from JFT as the unlabeled data. We train a ResNet-50 model on ImageNet and use it as our teacher model. We use RandAugment with the magnitude set to 9 as the noise.

The results are shown in Table 16. NoisyStudent leads to an improvement of 1.3% on the baseline model, which shows that NoisyStudent is effective for architectures other than EfficientNet.

Method	Top-1 Acc.	Top-5 Acc.
ResNet-50	77.6%	93.8%
NoisyStudent (ResNet-50)	**78.9%**	**94.3%**

Table 16: Experiments on ResNet-50.

Results on SVHN. We also evaluate NoisyStudent on a smaller dataset SVHN. We use the core set with 73K images as the training set and the validation set. The extra set with 531K images are used as the unlabeled set. We use EfficientNet-B0 with strides of the second and the third blocks set to 1 so that the final feature map is 4x4 when the input image size is 32x32.

As shown in Table 17, NoisyStudent improves the baseline accuracy from 98.1% to 98.6% and outperforms the previous state-of-the-art results achieved by RandAugment with Wide-ResNet-28-10.

Method	Accuracy
RandAugment (WRN)	98.3%
EfficientNet-B0	98.1%
NoisyStudent (B0)	98.6%

Table 17: Results on SVHN.

A.5. 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.[6]
- **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 [31] for details about mCE

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

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 [31] 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 NoisyStudent 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. NoisyStudent 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 [55] and in vision models that uses the Inception preprocessing [34, 79]. The overlap between RandAugment and ImageNet-P includes Brightness, Translate and Rotate.