Design Choices for Enhancing Noisy Student Self-Training

Aswathnarayan Radhakrishnan[1] Jim Davis[1] Zachary Rabin[1]
Benjamin Lewis[2] Matthew Scherreik[2] Roman Ilin[2]
[1] Ohio State University {radhakrishnan.39, davis.1719, rabin.30}@osu.edu
[2] AFRL, Wright-Patterson AFB {benjamin.lewis.13, matthew.scherreik.1, roman.ilin.1}@us.af.mil

Abstract

Semi-supervised learning approaches train on small sets of labeled data in addition to large sets of unlabeled data. Self-training is a semi-supervised teacher-student approach that often suffers from "confirmation bias" that occurs when the student model repeatedly overfits to incorrect pseudo-labels given by the teacher model for the unlabeled data. This bias impedes improvements in pseudo-label accuracy across self-training iterations, leading to unwanted saturation in model performance after just a few iterations. In this work, we study multiple design choices to improve the Noisy Student self-training pipeline and reduce confirmation bias. We showed that our proposed ***Weighted SplitBatch Sampler and Dataset-Adaptive Techniques for Model Calibration and Entropy-Based Pseudo-Label Selection*** *provided performance gains over existing design choices across multiple datasets. Finally, we also study the extendability of our enhanced approach to Open Set unlabeled data (containing classes not seen in labeled data). The source code can be licensed for use via email.*

1. Introduction

In today's data-driven world, deep learning techniques have become the predominant approach for computer vision tasks (such as image classification and object detection). Most state-of-the-art (SOTA) deep learning models use large-scale labeled datasets (e.g., ImageNet [7], JFT-3B [38], Instagram-3.5B [19]), a few of which are proprietary and cannot be leveraged by the public. It is challenging in practice to curate and annotate large labeled real-world datasets across different data domains and learning tasks. However, it is much easier to collect large quantities of *unlabeled data* in real-world domains (e.g., remote sensing imagery [20, 25], medical imagery [12, 32]). Semi-supervised learning (SSL) techniques are designed to jointly leverage small *labeled* datasets along with large *unlabeled* datasets to improve model performance.

Self-training (ST) [26–28, 36] is an iterative SSL method where a "teacher" model trained on the labeled data annotates the unlabeled data with pseudo-labels. The subsequent learning of the "student" model uses both the labeled and pseudo-labeled data. This process is iterated, as shown in Fig. 1. The major caveat of pseudo-labeling is the introduction of noisy pseudo-labels from incorrect predictions by the teacher. These noisy pseudo-labels accumulate over time, resulting in the model developing a bias toward incorrectly predicted pseudo-labels. This issue is known as the "confirmation bias" problem [1].

SSL techniques that learn from limited labeled data employ consistency regularization techniques [2, 3, 16, 29, 33] to reduce confirmation bias. Another popular method for reducing confirmation bias when enough labeled data is available is the NoisyStudent (NS) [34] pseudo-labeling approach that uses softmax confidence thresholding to filter out under-confident pseudo-label predictions. This approach also found that training a student model larger than the initial teacher made the student more robust to handle noisy pseudo-labels. In this paper, we focus on the NS iterative learning pipeline and explore multiple design choices and variations for NS to reduce confirmation bias.

We analyze multiple SSL design choices and study novel ways of integrating them into the NS pipeline. Our **proposed Weighted SplitBatch Sampling**, **Teacher Model Calibration**, and **Entropy-Based Pseudo-Label Selection** can be integrated into the data-loading stage of training to adaptively determine optimal hyperparameters settings for our design choices before training, unlike previous works [2, 3, 29] that require costly hyperparameter tuning training steps. The proposed design choices are modular and can be easily applied to enhance any pseudo-labeling-based SSL methods. We demonstrate using the proposed design choices to enhance NS across multiple benchmark datasets. Lastly, we present a practical scenario using real-world Open Set unlabeled data that contains data belonging to the target training classes and data from additional/unwanted classes. We demonstrate our enhanced ST technique with an Open Set Filtering module to improve performance even when trained with challenging Open Set data.

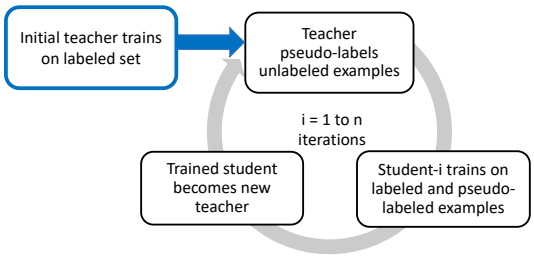

Figure 1. Basic iterative self-training pipeline.

2. Related Work

SSL is an active field of research in deep learning [23, 24, 40]. Consistency regularization and pseudo-labeling are some of the most commonly used SSL methods to learn from large sets of unlabeled data over recent years.

Consistency regularization approaches such as FixMatch [29], MixMatch [3], and ReMixMatch [2] follow the data manifold assumption that perturbations applied to the inputs, such as data augmentation, should not increase the likelihood of the predicted labels switching classes. These methods minimize the difference in predictions between an unlabeled sample and its perturbed counterpart. In [34], they discussed that consistency regularization methods work better in scarcely labeled data scenarios as they simultaneously learn to generate target predictions while maintaining the consistency requirements described above. However, teacher-student pseudo-labeling methods are preferred when the given labeled data is sufficient to train a supervised model for generating quality pseudo-labels (unlike consistency regularization methods). In our work, we focus on such teacher-student pseudo-labeling SSL methods.

The ST pseudo-labeling approach is one of the oldest and most widely used SSL approaches. In the NS [34] approach, the initial teacher model is trained on labeled data and then used to generate pseudo-labels for the unlabeled data. The student model is trained on both datasets (labeled + pseudo-labeled). They also found that injecting model noise (Dropout [30]) and data noise (RandAugment [5]) into the student training made it more robust to noisy pseudo-labels. NS also employed a student model larger than the original teacher to improve generalization in their experiments. They iterated the above steps using the student model from the previous iteration as the new teacher model, as shown in Fig. 1. We adapt this popular iterative teacher-student ST pseudo-labeling pipeline in our work.

There have been various approaches introduced to help reduce confirmation bias [1, 31] in SSL methods. In [9], a regularization term was added to the loss function to encourage the model to make confident low-entropy pseudo-label predictions. In [34], they used user-specified softmax thresholds on pseudo-labels to filter out noisy low-confidence predictions that can increase confirmation bias. In [16], a Gaussian mixture model was applied to divide the training data into clean and noisy sets using a per-sample loss distribution. They trained two networks where each network used the other network's divided set to reduce confirmation bias. In [35], confidence thresholding was used to separate the unlabeled data into clean in-distribution and noisy out-of-distribution data. They applied a class-aware clustering module for the in-distribution pseudo-labeled data along with a contrastive learning module to mitigate the noise in the out-of-distribution pseudo-labeled data. The modularity of our enhanced approach could enable the use of it as the drop-in pseudo-labeling module in [35] to reduce confirmation bias further. Our work proposes dataset-adaptive thresholding methods to filter out noisy pseudo-labels instead of manually fine-tuning hyperparameters for each dataset.

In [1], substantial data augmentation and regularization policies such as RandAugment [5], Mixup [39], and Dropout [30] were shown to minimize the effect of confirmation bias. Following [1], we apply an improved learnable version of Mixup called SAMix [17] in this work to help minimize confirmation bias. SAMix saliently mixes images by learning the mixing hyperparameters during training, eliminating the tuning stage for Mixup hyperparameters. Most of the above methods require complex modifications to the training architecture and optimization strategies to mitigate the problem of confirmation bias. Overall, we propose modular enhancements to fundamental training components that can adapt to existing ST pipelines to improve SSL performance.

3. Design Choices within Self-Training

We aim to enhance the baseline NS approach by modifying different stages of the pipeline to generate and select better pseudo-labels (having higher pseudo-label accuracy) to help reduce confirmation bias. Let the training data D be composed of the labeled subset with pairs $D_l = \{(x_i, y_i)\}_{i=1}^{N_l}$ where x_i denotes a labeled sample (e.g., image) and y_i denotes its corresponding ground truth label. The unlabeled subset contains data $D_u = \{(\tilde{x}_i)\}_{i=1}^{N_u}$ with no labels. A pseudo-label predicted for an unlabeled sample $\tilde{x}_i$ will be denoted as $\tilde{y}_i$. Let f_T and f_S denote the teacher and student models, respectively. We now propose the following comparisons of possible design choices for the fundamental ST components to study how the best components can be integrated to improve the NS baseline.

Hard vs. Soft Loss

Supervised deep learning models trained with one-hot ground truth labels generally use a *categorical* cross-entropy loss known as a **hard loss** (using a known, single ground truth label for each example and associated softmax

prediction). ST techniques must handle both clean ground truth labels and noisy pseudo-labels generated from the softmax prediction vectors of the teacher model. Previous work [1, 34] has shown that using a **soft loss** (used in NS) with the *entire* softmax vector of pseudo-label predictions as targets in a full cross-entropy loss works better than a hard loss with a one-hot categorical distribution over all training classes, helping to reduce confirmation bias.

Student Initialization

ST techniques train a new student model for every iteration. Two main methods exist for initializing the student model for each iteration. **Fresh-training** (used in NS) initiates every student model from scratch (randomly initialized model weights). Conversely, **fine-tuning** uses the weights of the model from *any* previous iterations with the best accuracy (on validation data) to initialize the student model and then fine-tune the weights during training.

Labeled/Pseudo-labeled Mini-Batch

Student models in ST learn from the labeled ground truth subset and the unlabeled subset with noisy pseudo-labels. ST commonly uses a **randomly collected (uniform) mini-batch** (used in NS) that tends to overfit due to their bias towards selecting a larger number of noisy pseudo-labeled than labeled data during training, as the unlabeled subset sizes are usually much larger than the labeled subset. We propose a **custom SplitBatch Sampler** that collects a user-specified split of labeled and pseudo-labeled examples for every mini-batch. Our approach uses bootstrapping to over-sample clean labeled examples that provide an additional regularization effect by reducing overfitting on the noisy pseudo-labels. The user controls the hyperparameter that sets the ratio of labeled to pseudo-labeled examples in a mini-batch, making the approach adaptive to differently sized labeled/pseudo-labeled subsets. For example, a larger ratio of labeled to pseudo-labeled examples can be used for datasets having a large number of labeled samples.

With a split batch of labeled and pseudo-label examples, the loss function should similarly engage a split loss. In this work, we combine a labeled loss L_{lab} and a pseudo-labeled loss L_{pslab} (average losses across respective mini-batches) with *equal* contributions into a custom MixedLoss function

$$L_{mix} = \lambda_b L_{lab} + (1 - \lambda_b) L_{pslab} \tag{1}$$

where λ_b is set to 0.5 to balance the loss between the labeled and pseudo-labeled examples in all experiments.

Sampling Techniques

ST methods can easily generate class-unbalanced and low-confidence pseudo-labeled subsets that can also increase confirmation bias. NS uses a **naïve softmax-thresholded class balancing** technique that first uses the uncalibrated softmax scores of pseudo-label predictions to threshold high-confidence predictions (softmax scores for the argmax class > 0.3). NS then samples a user-specified number of thresholded pseudo-labeled examples that have the highest softmax confidence across every class, oversampling examples from classes not having enough pseudo-labeled examples (less than the user-specified count per-class). This method requires manually specifying a softmax threshold and per-class sampling count for every dataset.

Our extended **Weighted SplitBatch sampler** adaptively re-weights and samples pseudo-labeled examples for each dataset using two different sample weightings. The first weighting uses inverted per-class counts ($\frac{1}{N_c}$ where N_c is the number of pseudo-labeled examples belonging to class c). *This method assigns larger weights to classes with a lower number of pseudo-labels, which thus will be oversampled during training.* The second set of sample weightings uses the per-class normalized softmax confidence scores

$$\text{normalizedSoftmax} = \frac{\max(\tilde{y})}{\max(S_c)} \tag{2}$$

where $\tilde{y}$ is the complete pseudo-label softmax vector prediction by the teacher model having argmax predicted class c for a given unlabeled sample $\tilde{x}$, and $S_c = \{\max(\tilde{y_1}), ..., \max(\tilde{y}_{N_c})\}$ is the set of max softmax scores for all pseudo-label predictions belonging to class c that scale the weights per-class to avoid oversampling only from pseudo-labeled predictions with higher softmax confidence that may lead to underfitting on examples from harder-to-classify classes which will not be sampled often. We average the two weights (class-counts and normalized softmax-confidence-based weights) to **adaptively obtain the final sampling weights assigned to all training samples based on pseudo-label counts and confidences** without needing any expensive hyperparameter re-tuning when changing datasets.

Pseudo-Label Selection

NS uses the **naïve softmax thresholding** approach (described above), employing softmax scores as a metric to determine pseudo-label confidence. However, modern deep neural networks are known to be poorly calibrated [10], implying that the softmax prediction probabilities do not accurately represent the true likelihood of the predictions. Hence, the uncalibrated softmax score is a poor confidence metric for rejecting noisy samples and thus can increase confirmation bias.

Alternatively, we propose adding a temperature-scaling calibration [10] step in the ST pipeline to the current teacher model for generating calibrated pseudo-label softmax predictions. We use a grid search over 400 linearly spaced temperature values between 0.05 and 20 and choose the optimal value, denoted by τ, with the lowest Expected Calibra-

tion Error [21] on the validation data. We then apply τ to soften/sharpen the softmax pseudo-label predictions of the teacher model to get a full softmax vector of pseudo-labels

$$\tilde{y} = \text{softmax}\left(\frac{f_T(\tilde{x})}{\tau}\right) \quad (3)$$

where $f_T(\tilde{x})$ denotes the output logits of the teacher model for a given unlabeled sample $\tilde{x}$ and $\tilde{y}$ is the calibrated pseudo-label softmax vector.

We next propose using **entropy thresholding** of calibrated softmax pseudo-label vectors rather than simply thresholding the softmax score for the argmax class to determine if the pseudo-label is acceptable. We calculate the **normalized entropy** (dividing by $\log(C)$ for a dataset containing C classes) of the calibrated pseudo-labels of validation data and then grid-search over 500 thresholds between 0 and 1. We then calculate the true-positive rate (TPR) and false-positive rate (FPR) for each entropy threshold on the validation data. We perform ROC analysis [8] by plotting the TPR against the FPR at the various thresholds and selecting the optimal threshold with the lowest Euclidean distance to the top left corner (optimal/perfect classification) of the ROC curve. This method for **pseudo-label selection can adapt to different datasets**, unlike the naïve approach of using hyperparameter-tuning to select softmax thresholds for each dataset.

Teacher Size

Lastly, the NS approach uses a **smaller-sized initial teacher model** trained on clean labels and a larger student model (and thus a larger subsequent teacher) trained jointly on labeled and pseudo-labeled examples. As previously mentioned, the NS model incorporates model noise (Dropout) and data noise (strong data augmentation techniques) to reduce confirmation bias. A natural alternative to their approach is a **same-sized teacher-student model**, where the teacher and student have the same model size but use *stronger* data augmentation techniques (SAMix+RandAugment) to similarly reduce confirmation bias. We compare both the size settings described above.

Given the above-listed design choice alternatives, we next compare them in a sequential greedy experimental setting to pick the best design to enhance the basic ST pipeline.

4. Experiments and Analysis

We aim to create an improved ST model by exploring the previously described design choices using the sequential strategy used by [18], where a linear series of experiments are employed to modernize a baseline model by augmenting the model with the best component obtained after each design choice comparison. Similarly, we start from the basic ST iterative learning pipeline, follow the roadmap described in Table 1, and choose the best design choices sequentially using a majority voting selection across multiple benchmark datasets to create an enhanced ST approach (rather than evaluating all possible combinations of design choices). We designed the order of experiments, starting from fundamental components (such as loss functions) and moving toward finer settings (such as sampling techniques and model sizes). We analyze in detail and discuss the insights gained from each design choice component at the end of every individual experimental comparison section to study the effects of each component and how they interact with the previously selected design choices. Finally, we evaluate the generalizability of the enhanced approach and compare it to the existing NS approach.

4.1. Datasets

We created custom labeled/unlabeled subsets from various benchmark datasets (SVHN [22], CIFAR-10 [14], and CIFAR-100 [14]) following the standard subset splits from previous SSL work [1], as shown at the top of Table 2. For each dataset, we also created a validation subset with ground-truth labels for hyperparameter tuning and evaluating model performance during training and a corresponding test subset for evaluating model inference. We first evaluated the experiments described in Table 1 on the three datasets and constructed the enhanced approach using the best component choices. We further evaluated the generalization performance of the resulting enhanced approach with *different labeled/unlabeled dataset splits* and *model sizes* on additional *larger datasets* (CINIC-10 [6], Tiny-ImageNet [15]) as shown at the bottom of Table 2. Note that every dataset except SVHN is class-balanced. Finally, we extended the enhanced approach with a basic Open Set detection technique to help filter out (suppress) additional/unwanted classes in a custom-built Open Set version of CIFAR-10/100 with 110 separate classes.

4.2. Comparison Roadmap

We trained a supervised baseline model for each dataset on only the labeled subset for three different randomly initialized runs for each experiment (Exp. 1 to 6). We reported the mean and standard deviation of the best test set score obtained across three student iterations (unless otherwise mentioned for experiments below). We used a ResNet(RN)-18 [13] for SVHN and a WideResNet(WRN) 28-2 [37] + SAMix for the CIFAR datasets unless otherwise mentioned. SAMix was not used on SVHN as certain mixing data augmentation policies were expected not to be appropriate for digit classification datasets (e.g., crops, flips). We also applied RandAugment with the hyperparameter settings for each dataset given in the original work [5]. RandomCrop and RandomHorizontalFlip were included in the data augmentation policy for the CIFAR datasets only.

Experiment	Description
Exp 1. Hard vs. Soft Loss	One-Hot Categorical Cross-Entropy Loss vs. Soft Cross-Entropy Loss
Exp 2. Student Initialization	Training from Scratch vs. Fine-tuning Student Iterations
Exp 3. Lab./Pseudo-lab. Mini-Batch	Random Mini-batch (Mixed) vs. SplitBatch (Labeled + Pseudo-labeled)
Exp 4. Sampling Techniques	Naïve Softmax-Thresholded Class Balancing vs. Weighted SplitBatch Sampling
Exp 5. Pseudo-Label Selection	Naïve Softmax Thresholding vs. Calibrated Entropy Thresholding
Exp 6. Teacher Size	Smaller vs. SameSized Teacher

Table 1. Experimental comparison roadmap.

Dataset (Classes)	Lab	UL	Val	Test
SVHN (10)	1K	70K	1K	26K
CIFAR-10 (10)	4K	42K	4K	10K
CIFAR-100 (100)	10K	30K	10K	10K
CINIC-10 (10)	20K	150K	10K	90K
TinyImageNet (200)	20K	60K	20K	10K

Table 2. Dataset sizes. (Lab: Labeled, UL: Unlabeled, Val: Validation, Test: Test set sizes)

Datasets	Mean Teacher Acc.
SVHN 1K	$75.69_{\pm 0.51}$
CIFAR-10 4K	$83.66_{\pm 0.11}$
CIFAR-100 10K	$63.9_{\pm 0.29}$

Table 3. Labeled subset supervised baseline results.

Datasets	Mean Student Acc.	
	Hard Loss	Soft Loss
SVHN 1K	$80.96_{\pm 0.42}$	**81.22**$_{\pm 1.12}$*
CIFAR-10 4K	$85.15_{\pm 0.07}$	**86.85**$_{\pm 0.23}$*
CIFAR-100 10K	$59.24_{\pm 0.35}$	**62.24**$_{\pm 0.36}$

Table 4. Hard vs. Soft loss results. (**Bold**: Best result in table, *: Current best result for each dataset. Applies to Tables 4-11)

Datasets	Mean Student Acc.	
	Fresh-Train	Fine-Tune
SVHN 1K	$81.22_{\pm 1.12}$	**81.55**$_{\pm 0.12}$*
CIFAR-10 4K	$86.85_{\pm 0.23}$	**87.45**$_{\pm 0.09}$*
CIFAR-100 10K	$62.24_{\pm 0.36}$	**65.86**$_{\pm 0.33}$*

Table 5. Student initialization comparison results.

We trained the initial teacher model for 400 epochs on the labeled subsets of all datasets following the suggested training and hyperparameter settings for SAMix [16]. Each student iteration was trained for 100 epochs (epoch for student models corresponds to one pass through labeled and pseudo-labeled examples). We used a batch size of 100 for all experiments. Table 3 shows the mean initial teacher accuracy trained only using the labeled subset (these scores are expected to be lower than fully-supervised SOTA benchmarks that use the complete datasets).

Exp 1. Hard vs. Soft Loss

Table 4 shows the comparison results between ST models using **soft** loss vs. **hard** loss. We can see that soft loss employed by the NS approach performs better on SVHN and CIFAR-10. Both losses degrade the performance on CIFAR-100 from the supervised baseline because the basic ST pipeline can easily overfit to the noisier CIFAR-100 pseudo-labels (CIFAR-100 has the worst initial teacher in Table 3, which would generate the noisiest pseudo-labels). *By 2-1 majority vote, we apply the **soft loss** (cross-entropy loss with soft targets) henceforth in our experiments that reduce confirmation bias by softening the noisy targets.*

Exp 2. Student Initialization

We next evaluated **fresh-training** vs. **fine-tuning** of ST models across training iterations. Table 5 shows that fine-tuning improved ST performance across all datasets compared to the fresh-training approach used in NS. *Hence we use **fine-tuning** for the remaining set of experiments as carrying over the learned weights from the initial cleanly trained teacher model during ST is beneficial.*

Exp 3. Labeled/Pseudo-labeled Mini-Batch

Table 6 shows the results of using the default **random mini-batch** approach used in NS against our **proposed SplitBatch** approach and the associated MixedLoss function L_{mix}. For SVHN and CIFAR-10, which have small amounts of labeled examples, we used a 20/80% labeled/pseudo-labeled batch split, whereas, for CIFAR-100, we used a 40/60% split as it has a larger number of labeled examples. *We use our better-performing **proposed SplitBatch approach** along with the above split percentages going forward, which reduces confirmation bias by oversampling (sampling with replacement) clean labeled examples in every mini-batch providing additional regularization that reduces overfitting on noisy pseudo-labels.*

Exp 4. Sampling Techniques

We compared the **naïve softmax-thresholded class balancing** technique employed by the NS approach with our **proposed weighted SplitBatch sampler** (employs class-

Datasets	Mean Student Acc.	
	Random	SplitBatch
SVHN 1K	$81.55_{\pm 0.12}$	**81.68**$_{\pm 0.93}$*
CIFAR-10 4K	$87.45_{\pm 0.09}$	**87.61**$_{\pm 0.12}$*
CIFAR-100 10K	$65.86_{\pm 0.33}$	**65.90**$_{\pm 0.15}$*

Table 6. Labeled/Pseudo-labeled mini-batch comparison results.

Datasets	Mean Student Acc.	
	Naïve	WeightedSplitBatch
SVHN 1K	**83.30**$_{\pm 0.84}$*	$81.07_{\pm 0.66}$
CIFAR-10 4K	$86.56_{\pm 0.33}$	**87.95**$_{\pm 0.09}$*
CIFAR-100 10K	$68.47_{\pm 0.29}$	**69.53**$_{\pm 0.15}$*

Table 7. Sampling techniques comparison results.

length balancing and confidence weighting using the same splits from the previous experiment). The student iterations henceforth are trained for 150 epochs (instead of 100). These methods need more epochs to converge as they work on thresholded/oversampled pseudo-labeled subsets (previous experiments used the complete set of pseudo-labeled data). Unlike NS, which used a softmax threshold of 0.3 on 1000-class ImageNet, we used a larger threshold of 0.5 to threshold noisy pseudo-labeled data as the maximum number of classes is only 100 in our datasets, resulting in higher softmax values for the argmax classes (0.5 is a natural decision boundary between low and high confidence). In this experiment, we employed our weighted SplitBatch sampling without thresholding pseudo-labeled data. Table 7 shows that our weighted SplitBatch sampler performed better on the CIFAR datasets. In contrast, the naïve method, which uses confidence-sorted sampling, is better on the easier SVHN dataset that had more highly confident pseudo-labels. However, the naïve method samples more incorrect high-confidence pseudo-labels on CIFAR datasets than SVHN, leading to performance degradation on CIFAR data. *By 2-1 majority vote, we employ our* ***novel Weighted SplitBatch sampler*** *in the ST pipeline henceforth as it generates dataset-adaptive weights for sampling class-balanced and confidence-calibrated pseudo-labels without needing any hyperparameter-tuning training steps.*

Exp 5. Pseudo-Label Selection

We compared the **naïve softmax-thresholding** approach used by NS from the previous experiment with our **proposed Dataset-Adaptive Calibrated Entropy Thresholding** for pseudo-label selection based on optimal temperature and entropy thresholds returned by our dataset-adaptive grid search methods on validation data, combined with our weighted SplitBatch sampling. Table 8 shows that our enhanced approach performed better than naïve softmax-thresholding on all datasets. We also improved upon the SVHN naïve sampling scores from the

Datasets	Mean Student Acc.	
	Softmax-Thresh.	Entropy-Thresh.
SVHN 1K	$83.30_{\pm 0.84}$	**84.28**$_{\pm 0.73}$*
CIFAR-10 4K	$86.56_{\pm 0.33}$	**88.52**$_{\pm 0.25}$*
CIFAR-100 10K	$68.47_{\pm 0.29}$	**69.84**$_{\pm 0.10}$*

Table 8. Pseudo-label selection results. (Thresh: Thresholding)

previous experiment, demonstrating the efficacy of using our weighted SplitBatch sampler and calibrated entropy thresholding in tandem. *Hereafter, we apply our* ***proposed calibrated entropy thresholding*** *method that reduces confirmation bias by employing the calibrated entropy thresholds to filter out noisy pseudo-labels.*

Exp 6. Teacher Size

Table 9 shows the results of employing **differently sized teacher models** in the ST pipeline. We found that the NS approach of using a smaller teacher (RN18 for SVHN and WRN28-2 for CIFAR-10 and CIFAR-100) and a larger student (RN34 for SVHN and WRN40-2 for CIFAR-10 and CIFAR-100) with model noise (Dropout) is unnecessary once we add our selected design choices to reduce confirmation bias. The results in Table 9 compare NS with the SmallerSameSized (SSS: using the *smaller* teacher model size as the student model size) and LargerSameSized (LSS: using the *larger* teacher model size as the student model size) approaches. We can see that SSS models performed on par on SVHN and slightly better on CIFAR than the NS approach, whereas the LSS approach improved accuracy across all datasets. *Hence we use* ***larger same-sized teacher-student models*** *that reduce confirmation bias by employing stronger data augmentation, which provides additional regularization to subdue noisy pseudo-labels.*

Final Model

We aggregated the best components, selected sequentially by majority voting on the three datasets, which resulted in selecting (1) Soft Loss, (2) Fine-tuning, (3) Proposed Weighted SplitBatch Sampler (w/ associated MixedLoss Function), (5) Proposed Calibrated Entropy Thresholding, and (6) Larger Same-Sized Teacher-Student models. The resulting enhanced approach is shown in Fig. 2. We refer to this final model with optimal ST design choices as the **enhanced self-train (EST)** approach.

4.3. Generalizability of EST

We evaluated how the selected best components in our EST model interact and work together by studying the model's generalizability to different labeled/unlabeled subset sizes and larger model architectures and datasets. We reported results from a single run for each experiment.

Datasets	Models	Mean Student Acc.		
		NS	SSS	LSS
SVHN 1K	NS: RN18+RN34, SSS: RN18, LSS: RN34	$84.28_{\pm 0.76}$	$84.28_{\pm 0.73}$	**86.71**$_{\pm 0.57}$*
C-10 4K	NS: WRN28-2+WRN40-2, SSS: WRN28-2, LSS: WRN40-2	$88.19_{\pm 0.08}$	$88.52_{\pm 0.25}$	**89.07**$_{\pm 0.11}$*
C-100 10K	NS: WRN28-2+WRN40-2, SSS: WRN28-2, LSS: WRN40-2	$69.23_{\pm 0.32}$	$69.84_{\pm 0.10}$	**71.06**$_{\pm 0.41}$*

Table 9. NoisyStudent comparison results. (C: CIFAR, NS: NoisyStudent, SSS: SmallerSameSized, LSS: LargerSameSized)

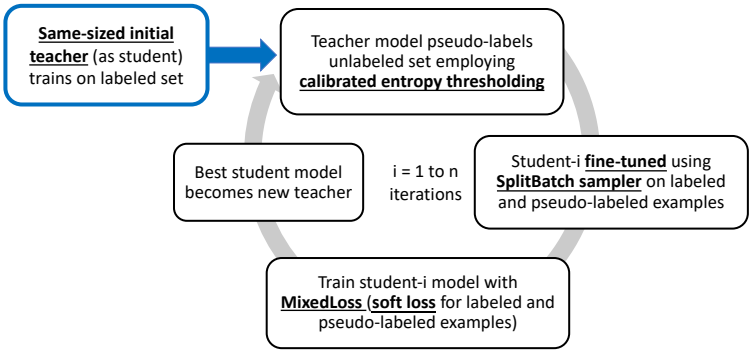

Figure 2. Enhanced self-training (EST) pipeline.

Different Labeled/Unlabeled Subset Sizes

We evaluated our EST approach on different labeled/unlabeled subset sizes of CINIC-10 with its large number of samples (270K) split into different-sized labeled/unlabeled subsets. First, we examined a small labeled data scenario with 1K labeled examples (100 examples per-class) across increases in unlabeled data (20K-150K). Next, we evaluated a large labeled data scenario with 20K labeled examples (2K examples per-class) across the various unlabeled sets. We applied a 20/80% mini-batch split in the small label scenario to avoid overfitting when oversampling the smaller labeled set. We used a 50/50% mini-batch split for the large labeled scenario having many labeled examples to oversample. A balanced loss was used in both scenarios (as in the earlier experiments). We trained all models using a WRN28-8 backbone for the same number of training optimization steps.

As expected, the results in Table 10 show that performance increases as we add more unlabeled examples during training in both scenarios, highlighting the importance of building large unlabeled datasets for semi-supervised learning methods. Our proposed EST approach provided significant improvements in the small labeled scenario while providing slight improvements to the large labeled scenario (as the teacher model already has enough labeled examples for better learning performance), showing adaptability to different-sized SSL datasets.

Larger Model Architectures and Datasets

We next evaluated the generalizability of our proposed EST approach using *larger models* with *longer epochs* and *larger datasets*. We evaluated a RN34 model on SVHN (previously used RN18) with a 20/80% mini-batch split, a WRN28-8 model on CIFAR-10/100 (previously used WRN28-2) with a 20/80% and 40/60% mini-batch split, respectively, a WRN28-8 model on the CINIC-10 dataset (20K Lab + 150K UnLab) with a 20/80% mini-batch split (previously used 50/50% split), and additionally included a RN34 model on TinyImageNet (which comprises more challenging, downsized ImageNet samples) with a 50/50% mini-batch split. We trained the teacher for 400 epochs and all student iterations for 200 epochs (previously used 150 epochs) with the SAMix+RandAugmet data augmentation policy for all models/datasets. Interestingly, we found that applying SAMix on SVHN helped improve digit classification performance (unlike previous expectations). We compared against previous related work, the NS approach reimplemented as described in [34] using smaller teacher models as suggested (RN18 for SVHN and TinyImageNet, WRN28-2 for CIFAR and CINIC datasets) but the same larger student model sizes as ours. Exp 6 (Table 9) already demonstrated that our proposed EST

CINIC-10 Split	Best Acc.	
	1K Lab	20K Lab
Init. Teacher	53.97	82.55
20K Unlab	65.82	82.64
50K Unlab	67.03	82.74
100K Unlab	67.42	83.24
150K Unlab	**67.63***	**83.59***

Table 10. Lab/Unlab subset sizes comparison results.

Datasets	Best Acc.	
	NS	EST
SVHN	91.65	**93.00***
CIFAR-10	89.15	**94.21***
CIFAR-100	70.53	**76.42***
CINIC-10	83.47	**88.59***
TinyImageNet	49.32	**52.23***

Table 11. EST vs. NS [34] best student top-1 accuracy results.

Experiment Description	Acc.
NS Teacher Model on Labeled Closed Set	71.82
EST Teacher Model on Labeled Closed Set	85.83
NS Best Student Model on Open Set	87.8
EST Best Student Model on Open Set	**92.62**
NS Best Student Model on Filtered Open Set	88.14
EST Best Student Model on Filtered Open Set	**93.12**

Table 12. Open Set ST results. (**Bold**: Best result in subsection)

approach performed better than NS when using either smaller or larger teacher models as the student model. The larger teacher model further augmented performance, thus validating its selection for comparison with NS. The results in Table 11 show that our proposed EST approach outperformed the previous related work, the NS [34] approach on all the evaluated datasets and also extended well to a more extensive set of target classes (from 10 to 200 classes).

4.4. Open Set Data

Real-world unlabeled data can be from an Open Set that contains data belonging to the target classes (classes from labeled training data) and data from additional non-target classes. Including non-target class examples in the unlabeled set can degrade SSL performance [11]. We propose a basic Open Set recognition technique using contrastive learning to build a feature space for all target classes, where the non-target classes should hopefully be farther away from the target classes. We used SimCLR [4] to learn a contrastive feature space from the labeled target data and the unlabeled data (contains target and non-target class examples). We used a validation set to find a mean prototype vector for each known target class and fit a Beta distribution per-class of the distances from labeled target examples to their class prototype. We then pre-filtered training examples expected to be from any non-target class by using a per-class Beta cumulative distribution function (CDF) and a global CDF threshold (learned from validation), where examples having CDF values above the threshold for all classes were considered to be from non-target classes.

We evaluated this method on our custom Open Set version of CIFAR-10/100 with labeled and unlabeled subsets. The labeled subset is made up of a Closed Set with 10 target classes (CIFAR-10 subset consisting of 4K images), and the unlabeled subset contains 110 total classes with 10 target classes (CIFAR-10 subset consisting of different 42K images) and 100 non-target classes (CIFAR-100 subset consisting of 42K images). We compared the performance of the NS approach (reimplemented as suggested in [34] with a smaller WRN28-2 initial teacher and the same larger WRN28-8 student as EST, thus resulting in a performance decrease) to our EST approach (same-sized WRN28-8 teacher-student). After grid-searching through multiple values, we found that pre-filtered training sets using CDF thresholds of 0.85 and 0.9 led to the best Closed Set validation accuracy for NS and EST training, respectively. Table 12 shows that our EST approach performed better than the NS approach showing that our enhancements for handling noisy pseudo-labels extend to Open Set data as well by providing some basic filtering of pseudo-labels belonging to non-target classes. We also found that both the NS and EST approaches had further improvements upon training with our filtered Open Set data, with our EST approach on filtered Open Set data performing the best.

5. Conclusion

We proposed multiple modular novel enhancements to the existing NS pipeline to reduce confirmation bias commonly seen in pseudo-labeling SSL methods. We demonstrated that integrating our proposed *Weighted SplitBatch Sampling, Adaptive Confidence Calibration, and Entropy-Based Pseudo-Label Selection* modules into the ST pipeline reduced overfitting on noisy pseudo-labels, thereby reducing confirmation bias. We built these enhancements into a **custom PyTorch DataLoader** that can adaptively work with any multi-class SSL dataset, enabling it to be applied to any pseudo-labeling-based ST methods. Our dataset-adaptive strategies, however, depend on having a small number of validation examples to select optimal hyperparameters. In cases where enough validation data is unavailable, we can still use our EST model with user-provided hyperparameter settings based on the dataset. We also demonstrated a basic Open Set Filtering technique that augmented ST performance compared to previous related work [34] on unlabeled training data with novel unseen class distributions. In future work, we plan to directly integrate Open Set recognition capabilities into ST models and leverage contrastive learning to assist in learning better feature representations to separate known and unknown classes.

6. Acknowledgments

This work was partly supported by the U.S AFRL under contract #GRT00054740 (Release #AFRL-2022-5050).

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. In *International Joint Conference on Neural Networks*, 2020. 1, 2, 3, 4

[2] David Berthelot, Nicholas Carlini, Ekin Dogus Cubuk, Alexey Kurakin, Kihyuk Sohn, Han Zhang, and Colin Raffel. ReMixMatch: Semi-Supervised Learning with Distribution Matching and Augmentation Anchoring. In *International Conference on Learning Representations*, 2020. 1, 2

[3] David Berthelot, Nicholas Carlini, Ian Goodfellow, Avital Oliver, Nicolas Papernot, and Colin Raffel. MixMatch: A Holistic Approach to Semi-Supervised Learning. In *Advances in Neural Information Processing Systems*, 2019. 1, 2

[4] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A Simple Framework for Contrastive Learning of Visual Representations. In *International Conference on Machine Learning*, 2020. 8

[5] Ekin D. Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V. Le. Randaugment: Practical Automated Data Augmentation with a Reduced Search Space. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops: Efficient Deep Learning for Computer Vision*, 2020. 2, 4

[6] Luke Darlow, Elliot Crowley, Antreas Antoniou, and Amos Storkey. CINIC-10 is not ImageNet or CIFAR-10. *arXiv preprint arXiv:1810.03505*, 2018. 4

[7] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2009. 1

[8] Tom Fawcett. An Introduction to ROC Analysis. *Pattern Recognition Letters*, 2006. 4

[9] Yves Grandvalet and Yoshua Bengio. Semi-Supervised Learning by Entropy Minimization. In *Advances in Neural Information Processing Systems*, 2004. 2

[10] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On Calibration of Modern Neural Networks. In *International Conference on Machine Learning*, 2017. 3

[11] Lan-Zhe Guo, Zhen-Yu Zhang, Yuan Jiang, Yu-Feng Li, and Zhi-Hua Zhou. Safe Deep Semi-Supervised Learning for Unseen-Class Unlabeled Data. In *International Conference on Machine Learning*, 2020. 8

[12] Ayaan Haque, Abdullah-Al-Zubaer Imran, Adam Wang, and Demetri Terzopoulos. Generalized Multi-Task Learning from Substantially Unlabeled Multi-Source Medical Image Data. *Machine Learning for Biomedical Imaging*, 2021. 1

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2016. 4

[14] Alex Krizhevsky. Learning Multiple Layers of Features from Tiny Images. 2009. 4

[15] Ya Le and Xuan Yang. Tiny ImageNet Visual Recognition Challenge. 2015. 4

[16] Junnan Li, Richard Socher, and Steven C.H. Hoi. DivideMix: Learning with Noisy Labels as Semi-supervised Learning. In *International Conference on Learning Representations*, 2020. 1, 2, 5

[17] Siyuan Li, Zicheng Liu, Di Wu, Zihan Liu, and Stan Z. Li. Boosting Discriminative Visual Representation Learning with Scenario-Agnostic Mixup. *arXiv preprint arXiv:2111.15454*, 2021. 2

[18] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A ConvNet for the 2020s. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. 4

[19] 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 *European Conference on Computer Vision*, 2018. 1

[20] Nicholas Kashani Motlagh, Aswathnarayan Radhakrishnan, Jim Davis, and Roman Ilin. A Framework for Semi-Automatic Collection of Temporal Satellite Imagery for Analysis of Dynamic Regions. In *IEEE/CVF International Conference on Computer Vision Workshops: Learning to Understand Aerial Images*, 2021. 1

[21] Mahdi Pakdaman Naeini, Gregory F. Cooper, and Milos Hauskrecht. Obtaining Well Calibrated Probabilities using Bayesian Binning. In *AAAI Conference on Artificial Intelligence*, 2015. 4

[22] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Ng. Reading Digits in Natural Images with Unsupervised Feature Learning. In *Advances in Neural Information Processing Systems*, 01 2011. 4

[23] Chapelle Olivier, Scholkopf Bernhard, and Zien Alexander. Semi-Supervised Learning. *IEEE Transactions on Neural Networks*, 2009. 2

[24] Yassine Ouali, Céline Hudelot, and Myriam Tami. An Overview of Deep Semi-Supervised Learning. *arXiv preprint arXiv:2006.05278*, 2020. 2

[25] Aswathnarayan Radhakrishnan, Jamie Cunningham, Jim Davis, and Roman Ilin. A Framework for Collecting and Classifying Objects in Satellite Imagery. In *Advances in Visual Computing*. Springer International Publishing, 2019. 1

[26] Ellen Riloff. Automatically Generating Extraction Patterns from Untagged Text. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence - Volume 2*. AAAI Press, 1996. 1

[27] Ellen Riloff and Janyce Wiebe. Learning Extraction Patterns for Subjective Expressions. In *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2003. 1

[28] H. Scudder. Probability of error of some adaptive pattern-recognition machines. *IEEE Transactions on Information Theory*, 1965. 1

[29] Kihyuk Sohn, David Berthelot, Chun-Liang Li, Zizhao Zhang, Nicholas Carlini, Ekin D. Cubuk, Alex Kurakin, Han Zhang, and Colin Raffel. FixMatch: Simplifying Semi-Supervised Learning with Consistency and Confidence. In *Advances in Neural Information Processing Systems*, 2020. 1, 2

[30] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A Simple

Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*, 2014. 2

[31] Jong-Chyi Su, Zezhou Cheng, and Subhransu Maji. A Realistic Evaluation of Semi-Supervised Learning for Fine-Grained Classification. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021. 2

[32] Nima Tajbakhsh, Laura Jeyaseelan, Qian Li, Jeffrey N. Chiang, Zhihao Wu, and Xiaowei Ding. Embracing Imperfect Datasets: A Review of Deep Learning Solutions for Medical Image Segmentation. *Medical Image Analysis*, 2020. 1

[33] Antti Tarvainen and Harri Valpola. Mean Teachers Are Better Role Models: Weight-Averaged Consistency Targets Improve Semi-Supervised Deep Learning Results. In *Advances in Neural Information Processing Systems*, 2017. 1

[34] Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V. Le. Self-Training with Noisy Student improves ImageNet Classification. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020. 1, 2, 3, 7, 8

[35] Fan Yang, Kai Wu, Shuyi Zhang, Guannan Jiang, Yong Liu, Feng Zheng, Wei Zhang, Chengjie Wang, and Long Zeng. Class-Aware Contrastive Semi-Supervised Learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. 2

[36] David Yarowsky. Unsupervised Word Sense Disambiguation Rivaling Supervised Methods. In *Proceedings of the 33rd Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics, 1995. 1

[37] Sergey Zagoruyko and Nikos Komodakis. Wide Residual Networks. *arXiv preprint arXiv:1605.07146*, 2016. 4

[38] Xiaohua Zhai, Alexander Kolesnikov, Neil Houlsby, and Lucas Beyer. Scaling Vision Transformers. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. 1

[39] Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond Empirical Risk Minimization. In *International Conference on Learning Representations*, 2018. 2

[40] Xiaojin Zhu. Semi-Supervised Learning Literature Survey. Technical report, Computer Sciences, University of Wisconsin-Madison, 2005. 2