Mitigating Confirmation Bias in Semi-supervised Learning via Efficient Bayesian Model Averaging

Anonymous authors
Paper under double-blind review

Abstract

State-of-the-art (SOTA) semi-supervised learning (SSL) methods have been highly successful in leveraging a mix of labeled and unlabeled data, often via self-training or pseudo-labeling. During pseudo-labeling, the model's predictions on unlabeled data are used for training and may result in confirmation bias where the model reinforces its own mistakes. In this work, we show that SOTA SSL methods often suffer from confirmation bias and demonstrate that this is often a result of using a poorly calibrated classifier for pseudo labeling. We introduce BaM-SSL, an efficient Bayesian Model averaging technique that improves uncertainty quantification in SSL methods with limited computational or memory overhead. We demonstrate that BaM-SSL mitigates confirmation bias in SOTA SSL methods across standard vision benchmarks of CIFAR-10, CIFAR-100 and ImageNet, giving up to 16% improvement in test accuracy on the CIFAR-100 with 400 labels benchmark. Furthermore, we also demonstrate their effectiveness in additional realistic and challenging problems, such as class-imbalanced datasets and in photonics science.

1 Introduction

While deep learning has achieved unprecedented success in recent years, its reliance on vast amounts of labeled data remains a long standing challenge. Semi-supervised learning (SSL) aims to mitigate this by leveraging unlabeled samples in combination with a limited set of annotated data. In computer vision, two powerful techniques that have emerged are consistency regularization (Bachman et al., 2014; Sajjadi et al., 2016) and pseudo-labeling (also known as self-training) (Rosenberg et al., 2005; Xie et al., 2019b). Broadly, consistency regularization enforces that random perturbations of the unlabeled inputs produce similar predictions, while pseudo-labeling assigns artificial labels to unlabeled samples, which are then used to train the model. These two techniques are typically combined by minimizing the cross-entropy between pseudo-labels and predictions that are derived from differently augmented inputs, and have led to strong performances on vision benchmarks (Sohn et al., 2020; Assran et al., 2021).

In many SOTA SSL methods, a selection metric (Lee, 2013; Sohn et al., 2020) based on the model's confidence is often used in conjunction with pseudo-labeling, where only confident pseudo-labels are selected to update the model. As such, there is a need for proper confidence estimates; in other words, the calibration of the model should be of paramount importance. Model calibration (Guo et al., 2017) can be understood as a measure of how a model's output truthfully quantifies its predictive uncertainty, i.e. it denotes the alignment between its prediction confidence and its ground-truth accuracy. Apart from the importance of calibration arising from the selection metric, the use of cross-entropy minimization objectives common in SSL implies that models will naturally be driven to output high-confidence predictions (Grandvalet & Bengio, 2004). Having high-confidence predictions is highly desirable in SSL since we want the decision boundary to lie in low-density regions of the data manifold, i.e. away from labeled data points (Murphy, 2022). However, without proper calibration, a model would easily become over-confident. This is highly detrimental as the model would be encouraged to reinforce its mistakes, resulting in the phenomenon commonly known as *confirmation bias* (Arazo et al., 2019).

In this work, we propose to mitigate confirmation bias in semi-supervised learning by incorporating approximate Bayesian techniques, which have been widely known to improve uncertainty estimates (Wilson & Izmailov,

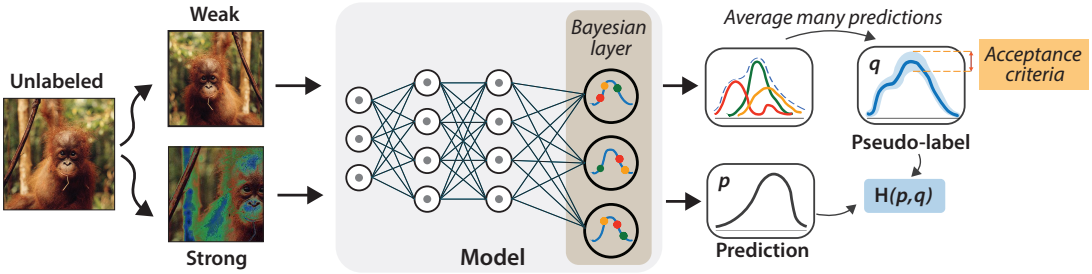

Figure 1: **Illustration of our Bayesian Model averaging (BaM) approach.** The last layer of the model has weights that are represented by probability distributions rather than the typical single fixed value. During pseudo-labeling of each unlabeled data point, multiple weights are sampled and averaged to derive the prediction. The selection criteria is based upon the variance over predictions.

Table 1: **Comparison of techniques used in BaM- with techniques used in prior art of SSL.**

SSL method	Augmentation	Pseudo-label	Selection metric
Temporal ensemble (Laine & Aila, 2016)	Weak	Model from earlier step	-
Mean teacher (Tarvainen & Valpola, 2017)	Weak	EMA	-
UDA (Xie et al., 2019a)	Weak & strong	Sharpen	Logit thresholding
MixMatch (Berthelot et al., 2019)	Weak	Averaging aug + sharpen	-
FixMatch (Sohn et al., 2020)	Weak & strong	Hard labels	Logit thresholding
FlexMatch (Zhang et al., 2021)	Weak & strong	Hard labels	Class-wise logit threshold
Our main method BaM-	Weak & strong	**Posterior sampling** + sharpen	**Variance thresholding**

2020). Our main approach, BaM-, performs Bayesian Model averaging during pseudo-labeling by incorporating a Bayesian last layer to the model and is illustrated in Fig. 1. Broadly, BaM- can be described as follows; instead of single-fixed-value weights, the last layer are parameterized as Gaussian random variables and two main modifications are made during pseudo-labeling: 1) multiple weight samples are drawn from the layer and averaged to derive the predictions and 2) the selection criterion is based on their posterior variances (further details follow in Section 4.1). We further contextualize the novelty of BaM- against prior art in the SSL literature in Table 1. In contrast to prior methods, BaM- is designed to specifically target improving model calibration in order to mitigate confirmation bias.

Our contributions are summarized as follows:

1. We introduce BaM-, which is designed to mitigate confirmation bias via Bayesian model averaging in SOTA SSL methods based upon a selection metric. We empirically demonstrate that BaM- effectively improves model calibration, resulting in better performances on standard benchmarks like CIFAR-10 and CIFAR-100, notably giving up to 16% gains in test accuracy.

2. We also demonstrate the efficacy of BaM- in more challenging and realistic scenarios, such as class-imbalanced datasets and a real-world application in photonic science.

3. Finally, we explore a different family of SSL methods that do not rely on a selection metric and show that approximate bayesian techniques can also be effective in improving calibration and test performance in these methods. We evaluate these methods on CIFAR-10, CIFAR-100 and ImageNet.

2 Related Work

Semi-supervised learning (SSL) and confirmation bias. A fundamental problem in SSL methods based on pseudo-labeling (Rosenberg et al., 2005) is that of confirmation bias (Tarvainen & Valpola, 2017; Murphy, 2022), i.e. the phenomenon where a model overfits to incorrect pseudo-labels. Several strategies have

emerged to tackle this problem; Guo et al. (2020) and Ren et al. (2020) looked into weighting unlabeled samples, Thulasidasan et al. (2019) and Arazo et al. (2019) proposes to use augmentation strategies like MixUp (Zhang et al., 2017), while Cascante-Bonilla et al. (2020) proposes to re-initialize the model before every iteration to overcome confirmation bias. Another popular technique is to impose a selection metric (Yarowsky, 1995) to retain only the highest quality pseudo-labels, commonly realized via a fixed threshold on the maximum class probability (Xie et al., 2019a; Sohn et al., 2020). Recent works have further extended such selection metrics to be based on dynamic thresholds, either in time (Xu et al., 2021) or class-wise (Zou et al., 2018; Zhang et al., 2021). Different from the above approaches, our work proposes to overcome confirmation bias in SSL by directly improving the calibration of the model through approximate Bayesian techniques.

Model calibration and uncertainty quantification. Proper estimation of a network's prediction uncertainty is of practical importance (Amodei et al., 2016) and has been widely studied. A common approach to improve uncertainty estimates is via Bayesian marginalization (Wilson & Izmailov, 2020), i.e. by weighting solutions by their posterior probabilities. Since exact Bayesian inference is computationally intractable for neural networks, a series of approximate Bayesian methods have emerged, such as variational methods (Graves, 2011; Blundell et al., 2015; Kingma et al., 2015), Hamiltonian methods (Springenberg et al., 2016) and Langevin diffusion methods (Welling & Teh, 2011). Other methods to achieve Bayesian marginalization also exist, such as deep ensembles (Lakshminarayanan et al., 2016) and efficient versions of them (Wen et al., 2020; Gal & Ghahramani, 2015), which have been empirically shown to improve uncertainty quantification. The concept of uncertainty and calibration are inherently related, where calibration is commonly interpreted as the frequentist notion of uncertainty. It is known that a well specified Bayesian model (i.e. one where the prior captures the model uncertainty) has a well-calibrated posterior (Gelman et al., 2021). Motivated by this, in our work we adopt some approximate bayesian techniques specifically for the context of semi-supervised learning in order to improve model calibration during pseudo-labeling and empirically validate their effectiveness. While other methods for improving model calibration exists (Platt, 1999; Zadrozny & Elkan, 2002; Guo et al., 2017), these are most commonly achieved in a post-hoc manner using a held-out validation set; instead, we seek to improve calibration during training and with a scarce set of labels. Finally, in the intersection of SSL and calibration, Rizve et al. (2021) proposes to leverage uncertainty to select a better calibrated subset of pseudo-labels. Our work builds on a similar motivation, however, in addition to improving the selection metric with uncertainty estimates, we show that directly incorporating approximate Bayesian techniques into SSL methods can indeed improve calibration via its better-calibrated approximate posterior.

3 Notation and Background

Given a small amount of labeled data $\mathcal{L} = \{(x_l, y_l)\}_{l=1}^{N_l}$ (here, $y_l \in \{0,1\}^K$, are one-hot labels) and a large amount of unlabeled data $\mathcal{U} = \{x_u\}_{u=1}^{N_u}$, i.e. $N_u \gg N_l$, in SSL, we seek to perform a K-class classification task. Let $f(\cdot, \theta_f)$ be a backbone encoder (e.g. ResNet or WideResNet) with trainable parameters θ_f, likewise let $h(\cdot, \theta_h)$ be a linear classification head, and H denote the standard cross-entropy loss.

SSL methods based on a selection metric. Many SSL methods such as Pseudo-Labels (Lee, 2013), UDA (Xie et al., 2019a) and FixMatch (Sohn et al., 2020) use a selection metric in conjunction with pseudo-labeling to achieve SOTA performance. These methods minimizes a cross-entropy loss on augmented copies of unlabeled samples whose confidence exceeds a pre-defined threshold. Let α_1 and α_2 denote two augmentation transformations and their corresponding network predictions for sample x to be $q_1 = h \circ f(\alpha_1(x))$ and $q_2 = h \circ f(\alpha_2(x))$, the total loss on a batch of unlabeled data has the following form:

$$L_u = \frac{1}{\mu B} \sum_{u=1}^{\mu B} \mathbb{1}(\max(q_{1,u}) \geq \tau) H(\rho_t(q_{1,u}), q_{2,u}) \tag{1}$$

where B denotes the batch-size of labeled examples, μ a scaling hyperparameter for the unlabeled batch-size, $\tau \in [0,1]$ is a threshold parameter often set close to 1 and ρ_t is either a sharpening operation on the pseudo-labels, i.e. $[\rho_t(q)]_k := [q]_k^{1/t} / \sum_{c=1}^{K} [q]_c^{1/t}$ or an argmax operation (i.e. $t \to 0$). ρ_t also implicitly includes a "stop-gradient" operation, i.e. gradients are not back-propagated from predictions of pseudo-labels.

L_u is combined with the expected cross-entropy loss on labeled examples, $L_l = \frac{1}{B}\sum_{l=1}^{B} H(y_l, q_{1,l})$ to form the combined loss $L_l + \lambda L_u$, with a scaling hyperparameter λ. Differences between Pseudo-Labels, UDA and FixMatch are detailed in Appendix D.1.

Calibration metrics. A popular empirical metric to measure a model's calibration is via the *Expected Calibration Error (*ECE*)*. Following (Guo et al., 2017; Minderer et al., 2021), we focus on a slightly weaker condition and consider only the model's most likely class-prediction, which can be computed as follows. Let $\rho_0(q)$ denote the model's prediction (where ρ_0 is the argmax operation) as defined before, the model predictions on a batch of N samples are grouped into M equal-interval bins, i.e. $\mathcal{B}_m$ contains the set of samples with $\rho_0(q) \in (\frac{m-1}{M}, \frac{m}{M}]$. ECE is then computed as the expected difference between the accuracy and confidence of each bin over all N samples:

$$\text{ECE} = \sum_{m=1}^{M} \frac{|\mathcal{B}_m|}{N} |\text{acc}(\mathcal{B}_m) - \text{conf}(\mathcal{B}_m)| \tag{2}$$

where $\text{acc}(\mathcal{B}_m) = (1/|\mathcal{B}_m|)\sum_{i\in\mathcal{B}_m} \mathbb{1}(\rho_0(q_i) = y_i)$ and $\text{conf}(\mathcal{B}_m) = (1/|\mathcal{B}_m|)\sum_{i\in\mathcal{B}_m} \max q_i$ with y_i the true label of sample i. In this work, we estimate ECE using $M = 10$ bins. We also caveat here that while ECE is not free from biases (Minderer et al., 2021), we chose ECE over alternatives (Brier, 1950; DeGroot & Fienberg, 1983) due to its simplicity and widespread adoption.

4 Mitigating Confirmation Bias in Semi-supervised learning

As we see from Eq. (1), the model's confidence (the maximum softmaxed logit value) is used to determine if the pseudo-label of a particular unlabeled data point is used to update the model; as such it is important for the model to have proper confidence estimates, i.e. to be well-calibrated. More recently, temperature scaling (Guo et al., 2017) or other similar methods (Platt, 1999; Zadrozny & Elkan, 2002) have been highly effective towards model calibration. However, such methods pose several challenges in the semi-supervised setting; 1) they operate post-hoc, i.e. *after* training is completed, while in SSL the model needs to be calibrated constantly *during* training to reduce confirmation bias of pseudo-labels; 2) these methods use a held-out labeled set (typically around 10% of the total dataset size (Guo et al., 2017)) to perform calibration, while common benchmarks of SSL typically have label percentages less than that (up to as little as <1%), making it challenging to create a held-out set with so little data to begin with.

4.1 Mitigating Confirmation Bias with Bayesian Model Averaging

Given the above limitations of post-hoc calibration methods, in this work we propose to incorporate approxmate Bayesian methods such as Bayesian Neural Networks (BNN) into existing SSL methods. Bayesian models, when well-specified (i.e. where the prior captures the model's uncertainty), are known to produce well-calibrated posteriors (Gelman et al., 2021) and approximate Bayesian techniques have been widely empirically shown to produce well-calibrated uncertainty estimates in deep neural networks (Wilson & Izmailov, 2020; Lakshminarayanan et al., 2016; Blundell et al., 2015). To minimize the additional computational overhead, we propose to only replace the **final layer** of the network, i.e. the linear classification head h, with a BNN layer. For notation convenience, we denote the input embedding to the BNN layer to be v in this section.

Following a Bayesian approach, we first assume a prior distribution on weights $p(\theta_h)$. Given some training data $\mathcal{D}_\mathcal{X} := (X, Y)$, we seek to calculate the posterior distribution of weights, $p(\theta_h|\mathcal{D}_\mathcal{X})$, which can then be used to derive the posterior predictive $p(y|v, \mathcal{D}_\mathcal{X}) = \int p(y|v, \theta_h)p(\theta_h|\mathcal{D}_\mathcal{X})d\theta_h$. This process is also known as "Bayesian model averaging" or "Bayesian marginalization" (Wilson & Izmailov, 2020). Since exact Bayesian inference is computationally intractable for neural networks, we adopt a variational approach following Blundell et al. (2015), where we learn a Gaussian variational approximation to the posterior $q_\phi(\theta_h|\phi)$, parameterized by ϕ, by maximizing the evidence lower-bound (ELBO) (see Appendix C.1 for details). The ELBO $= \mathbb{E}_q \log p(Y|X;\theta) - KL(q(\theta|\phi)\|p(\theta))$ consists of a log-likelihood (data-dependent) term and a KL (prior-dependent) term. We provide some preliminary theoretical connections to generalization

bounds via Corollary 1 in Appendix A. Corollary 1 shows that the generalization error is upper bounded by the negative ELBO, i.e. by maximizing the ELBO we may improve generalization.

Pseudo-labeling via BaM-. As depicted in Fig. 1, pseudo-labeling in BaM- proceeds in two-stages: 1) M weights from the BNN layer are sampled and predictions are derived from the Monte Carlo estimated posterior predictive, i.e. $\hat{q} = (1/M)\sum_m^M h(v, \theta_h^{(m)})$, and 2) the selection criteria is based upon their variance, $\sigma_c^2 = (1/M)\sum_m^M (h(v, \theta_h^{(m)}) - \hat{q})^2$, at the predicted class $c = \text{argmax}_{c'} [\hat{q}]_{c'}$. This constitutes a **more accurate measure of model uncertainty** compared to the maximum logit value commonly used in prior SSL methods which **does not have an uncertainty interpretation**. The variance is also highly intuitive — if the model's prediction has a large variance, it is highly uncertain and the pseudo-label should not be accepted. We later show that better uncertainty estimates from BaM- effectively mitigates confirmation bias. In practice, as σ_c^2 decreases across training, we use a simple quantile Q over the batch to define the threshold where pseudo-labels of samples with $\sigma_c^2 < Q$ are accepted, with Q as a hyperparameter. Algorithm 1 shows a snippet of pseudo-code to highlight the main modifications introduced by BaM- during pseudo-labeling (a more complete version of the pseudo-code can be found in Appendix C.1).

We explore the effectiveness of BaM- by modifying upon SOTA SSL methods and denote them with the BaM- suffix, i.e. "BaM-X" incorporates approximate Bayesian Model averaging (BaM) during pseudo-labeling for SSL method X.

5 Experimental Setup

In all our experiments, we begin with and modify upon the original implementations of the baseline SSL methods. The backbone encoder f is a Wide ResNet-28-2 and Wide ResNet-28-8 for the CIFAR-10 and CIFAR-100 benchmarks respectively. We use the default hyperparameters and dataset-specific settings (learning rates, batch size, optimizers and schedulers) recommended by the original authors for both the baselines and in BaM-. We set the weight priors in BaM- as unit Gaussians and use a separate Adam optimizer for the BNN layer with learning rate 0.01, no weight decay and impose the same cosine learning rate scheduler as the backbone. We set $Q = 0.75$ for the CIFAR-100 benchmark and $Q = 0.95$ for the CIFAR-10 benchmark; which are both linearly warmed-up from 0.1 in the first 10 epochs. As Q is computed across batches, we improve stability by using a moving average of the last 50 quantiles.

ECE and test accuracy evaluation. In our experiments, we found that the test accuracy exhibits a considerable amount of noise across training, especially in label-scarce settings. Sohn et al. (2020) proposes to take the median accuracy of the last 20 checkpoints, while Zhang et al. (2021) argues that this fixed training budget approach is not suitable when convergence speeds of the algorithms are different (as the faster converging algorithm would over-fit more severely at the end) and thus report also the overall best accuracy. In our experiments, we adopt a balance between the two aforementioned approaches: we consider the median of 20

Algorithm 1 Snippet of PyTorch-style pseudocode showing pseudo-labeling in BaM-UDA.

```
# Q: quantile parameter
# num_samples: number of weight samples

q_list = []
for x_weak, x_strong in unlabeled_loader:
    z_weak, z_strong = encoder(x_weak), encoder(x_strong) # get representations
    mean_weak, std_weak = bayes_predict(bayes_classifier, z_weak) # get mean and std of predictions
    q_list.pop(0) if len(q_list) > 50 # keep 50 most recent quantiles
    q_list.append(quantile(std_weak, Q))
    accept_mask = std_weak.le(q_list.mean()) # determine acceptance for samples with small std

    # compute unlabeled loss using soft pseudo-labels on accepted samples
    loss_unlab = cross_entropy_loss(bayes_classifier(z_strong), sharpen(mean_weak)) * accept_mask
    loss_kl = KL_loss(bayes_classifier) # prior-dependent (data-independent) loss
    loss = loss_unlab + loss_kl

def bayes_predict(h, z):
    outputs = stack([h(z).softmax(-1) for _ in range(num_samples)] # sample weights
    return outputs.mean(), outputs.std() # mean and std of predictions
```

Table 2: **BaM- in SSL** showing "Test accuracy (%) / ECE". BaM- improves calibration and result in better test accuracies, consistently across all benchmarks and across two SSL methods, FixMatch (FM) (Sohn et al., 2020) and UDA (Xie et al., 2019a). For each benchmark, results are averaged over 3 random dataset splits.

	CIFAR-10		CIFAR-100		
	250 labels	2500 labels	400 labels	4000 labels	10000 labels
FM (repro)	95.0±0.19 / 0.046±0.002	95.7±0.03 / 0.039±0.0	56.4±1.6 / 0.366±0.017	74.2±0.2 / 0.183±0.003	78.1±0.2 / 0.147±0.001
BaM-FM (ours)	95.1±0.07 / 0.044±0.000	95.7±0.1 / 0.039±0.0	59.0±1.4 (↑2.6) / 0.331±0.015	74.8±0.09 (↑0.6) / 0.171±0.002	78.1±0.2 / 0.139±0.002
UDA (repro)	94.1±0.6 / 0.053±0.006	95.7±0.05 / 0.039	44.1±0.7 / 0.473±0.013	72.9±0.01 / 0.189±0.003	77.2±0.3 / 0.154±0.002
BaM-UDA (ours)	**95.2**±0.04 (↑1.1) / **0.042**±0.00	**95.9**±0.08 (↑0.2) / **0.038**	**60.3**±0.6 (↑16.2) / **0.314**±0.005	**75.2**±0.1 (↑2.3) / **0.165**±0.002	**78.3**±0.2 (↑1.1) / **0.138**±0.003

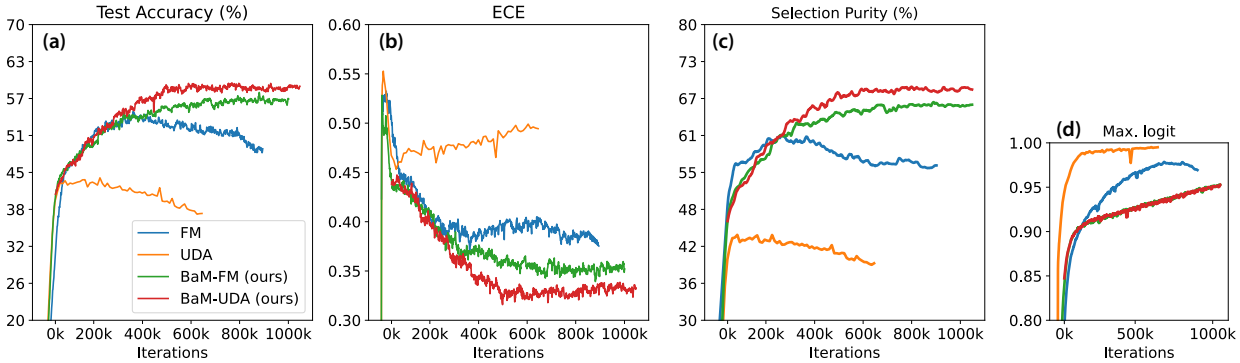

Figure 2: **FixMatch, UDA, BaM-FM and BaM-UDA across training** on CIFAR-100 with 400 labels (a) Test accuracies and (b) ECE as a function of training time. (c) Selection purity shows the accuracy of unlabeled samples that are accepted by the selection metric and (d) Max. logit shows the model's confidence, i.e. the average maximum class probability of all unlabeled samples.

checkpoints around the best accuracy checkpoint as the *convergence criteria*, and report this value as the test accuracy. The ECE is reported when the model reaches this convergence criteria (one could also also aggregate the ECEs up till convergence — we found this gave similar trends and thus report the simpler metric).

6 Results

6.1 BaM- improves model calibration and consistently leads to better SSL performances

Results on the CIFAR-10 and CIFAR-100 benchmarks for various number of labels are depicted in Table 2. Table 2 demonstrates that BaM- successfully reduces the ECE over the baselines across all the benchmarks and as a result, we also attained significant improvements in test accuracies, notably up to 16.2% (for UDA on CIFAR-100-400 labels). Interestingly, while the baseline FixMatch outperforms UDA across all the benchmarks, improving calibration in UDA (i.e. BaM-UDA) allows it to outperfom both FixMatch and BaM-FixMatch. A key difference between FixMatch and UDA is the use of hard pseudo-labels in FixMatch (i.e. $t \to 0$ in ρ_t defined in Section 3) versus soft pseudo-labels in UDA (with $t = 0.4$); this suggests that a Bayesian classifier is more effective in conjunction with soft pseudo-labels. Leveraging on this insight, we set $t = 0.9$ in BaM-UDA (also see Appendix G.2 for ablations on t), resulting in consistently better performances over the two SSL baselines across all CIFAR benchmarks.

Overall, we found that the improvements in both calibration and test accuracy are more significant for label-scarce settings, as expected, since the problem of confirmation bias is more acute there and BaM- can provide greater benefits by mitigating this. The additional computation overhead incurred from BaM- is minimal, adding approximately only 2-5% in wall-clock time (see Appendix H). While previous works (Sohn et al., 2020) also include extreme low label settings such as CIFAR-10-40 labels, we found this benchmark to be highly sensitive to the random initialization and different splits of the data, giving up to 2% variance with the exact same SSL method and thus we exclude them in our study.

Table 3: **Long-tailed CIFAR-10 & CIFAR-100** showing "Test accuracy (%) / ECE". We use 10% of labels from each class. For better interpretation, we also show the supervised (100% labels) accuracy reported in Cao et al. (2019), which use a different architecture, ResNet-32, and an algorithm targeted for long-tailed problems.

	CIFAR-10-LT		CIFAR-100-LT	
	$\alpha = 10$	$\alpha = 100$	$\alpha = 10$	$\alpha = 100$
FixMatch	91.3 / 0.073	70.0 / 0.26	48.8 / 0.38	28.6 / 0.55
BaM-UDA (ours)	**91.6** (↑0.3) / **0.067**	**71.2** (↑1.2) / **0.24**	**53.6** (↑4.8) / **0.32**	**31.9** (↑3.3) / **0.50**
Supervised (reported)	88.2	77.0	58.7	42.0

6.2 BaM- improves performances by reducing confirmation bias

To further understand how BaM- mitigates confirmation bias, we track the test accuracy, ECE, model confidence and ground-truth accuracy of *accepted* pseudo-labels (i.e. the "selection purity") over the course of training for the baselines and BaM-, as shown in Fig. 2. While the baselines learn effectively for the initial stages of training, learning is eventually hindered. Due to the entropy minization objective, the model is encouraged to output increasingly confident predictions (as evident from the growing model confidence in Fig. 2d). Thus, in the absence of explicit calibration, the baselines quickly become *over-confident*, resulting in confirmation bias where the model reinforces its mistakes. Confirmation bias in the baselines is particularly evident from the selection purity (i.e. the ground truth accuracy of accepted pseudo-labels) in Fig. 2c — after a short amount of training, the selection purity starts to drop suggesting that the model begins to accept pseudo-labels that it makes mistakes on. In contrast, BaM- successfully mitigates confirmation bias as evident from the constantly improving selection purity, thus promoting learning for longer periods to result in better final performances. Further ablation studies are discussed in Appendix G.2.

6.3 BaM- is more effective in class-imbalanced settings

Long-tailed image datasets. Datasets in the real world are often long-tailed or class-imbalanced, where some classes are more commonly observed while others are rare. We curate long-tailed versions from the CIFAR datasets following Cao et al. (2019), where α indicates the imbalance ratio (i.e. ratio between the sample sizes of the most frequent and least frequent classes). We randomly select 10% of the samples in each class to form the labeled set; see further details in Appendix E. We use the best performing baseline method from Table 2, i.e. FixMatch (FM), as our baseline and compare against BaM-UDA. Results are shown in Table 3, where BaM-UDA achieves consistent improvements over FM in both calibration and accuracy across all benchmarks. Notably, gains from BaM- are more significant than those in the class-balanced settings (for e.g. BaM-UDA improves upon FM by a smaller margin of 1.5% on the CIFAR-100-4000 labels benchmark which is also approximately 10%). Further analysis are provided in Appendix E.2, where test samples were separated into three groups depending on the number of samples per class and test accuracies are plotted for each group.

Photonics science. A practical example of a real-world domain where long-tailed datasets are prevalent is that of science – samples with the desired properties are often much rarer than trivial samples. Further, SSL is highly important in scientific domains since labeled data is particularly scarce (owing to the high resource cost needed for data collection). To demonstrate the effectiveness of BaM-, we adopt a problem in photonics (Loh et al., 2022), where the task is a 5-way classification of photonic crystals (PhCs) based on their

Table 4: **Photonic crystals (PhC) band gap prediction** showing "Test accuracy (%) / ECE". The fully-supervised (100% labels) accuracy is 88.5%.

	PhC-10%	PhC-1%
FixMatch	78.8 / 0.098	55.0 / 0.385
BaM-UDA	**81.0** (↑2.2) / **0.052**	**56.9** (↑1.9) / **0.356**

band gap sizes. A brief summary and visualization of this dataset are available in Appendix F. We explored an approach similar to FixMatch for the baseline and similar to BaM-UDA for ours (some modifications were needed in the augmentation strategies to respect the correct physics of this problem; see Appendix F). Results are shown in Table 4, where we demonstrate the consistency of BaM-UDA's effectiveness in improving calibration and accuracies in this real-world problem.

7 Exploring SSL methods without a selection metric

So far, we have looked at two popular SSL methods that are based on a selection metric. More recently, there is also a family of SSL methods based upon representation learning (Assran et al., 2021; Chen et al., 2020; Caron et al., 2021), where the model is first trained to produce useful representations and subsequently fine-tuned on limited labeled data. This give rise to some interesting questions: Is confirmation bias still an issue in SSL methods without a selection metric? And if so, can BaM- or other approximate Bayesian techniques be used to mitigate confirmation bias? We explore these questions in this section, using PAWS (Assran et al., 2021) as a canonical example for the family of SSL methods based on representation learning approach. Details about the training loss used in PAWS are explained in Appendix B. The main difference PAWS has in contrast to the methods studied above is in its prediction generation technique — instead of a classification head, PAWS uses a non-parametric nearest neighbour classification method over its representations.

7.1 Exploring other approximate Bayesian techniques

Since BaM- replaces the classification head and PAWS does not have one, we incorporate BaM- into PAWS by using a Bayesian last layer in the encoder and "Bayesian marginalizing" over the representations (also see Appendix B.1 for detailed formulation). Nonetheless, apart from Bayesian marginalization, BaM- is originally desirable towards improving uncertainty estimates in the selection metric. Since PAWS does not use a selection metric, instead of Bayesian marginalizing or aggregating over just one layer, one could seek to aggregate over the entire network. To perform this aggregation computational efficiently, we explored other approximate Bayesian approaches based upon weight averaging, such as Stochastic Weight Averaging (Izmailov et al., 2018) (SWA) and Exponential Moving Averaging (EMA) (Tarvainen & Valpola, 2017; He et al., 2020; Grill et al., 2020) and studied their role in mitigating confirmation bias during pseudo-labeling.

Weight averaging during pseudo-labeling We maintain a separate set of non-trainable weights θ_g containing the aggregated weight average of θ_h which is used to produce pseudo-labels throughout training. In SWA, we update them via $\theta_g \leftarrow (n_a\theta_g + \theta_f)/(n_a + 1)$ at every iteration, where n_a represents the total number of models in the aggregate and θ_f are the trainable parameters of our backbone encoder as before. In practice, we switch on SWA only after some amount of training. In EMA, this update is $\theta_g \leftarrow \gamma\theta_g + (1-\gamma)\theta_f$, where γ is a momentum hyperparameter controlling how much memory θ_g should retain at each iteration (see Appendix C.2 for pseudocode). While EMA has been previously explored in the context of SSL (Tarvainen & Valpola, 2017), to the best of our knowledge SWA has not been explored in SSL and more importantly, their link to mitigating confirmation bias in pseudo-labeling has not been explicitly shown.

Table 5: "Test accuracy (%) / ECE" for PAWS (Assran et al., 2021) and our methods. 'BaM-" refers to Bayesian Model averaging via a BNN final layer, while +SWA and +EMA are using weight averaging techniques. For each benchmark, results are averaged over 3 random dataset splits.

	CIFAR-10	CIFAR-100	
	2500 labels	4000 labels	10000 labels
PAWS (repro)	95.3±0.1 / 0.046±0.001	71.5±0.3 / 0.232±0.005	75.6±0.1 / 0.198±0.002
BaM-PAWS	95.4±0.1 (↑0.1) / 0.046±0.001	72.5±0.3 (↑1.0) / 0.228±0.003	76.5±0.2 (↑0.9) / 0.195±0.002
PAWS+SWA	95.6±0.1 (↑0.3) / 0.046±0.001	**74.5**±0.3 (↑3.0) / **0.193**±0.002	**78.4**±0.4 (↑2.8) / **0.164**±0.003
PAWS+EMA	**95.8**±0.0 (↑0.5) / **0.042**±0.000	73.5±0.2 (↑2.0) / **0.193**±0.005	77.2±0.1 (↑1.6) / 0.168±0.003

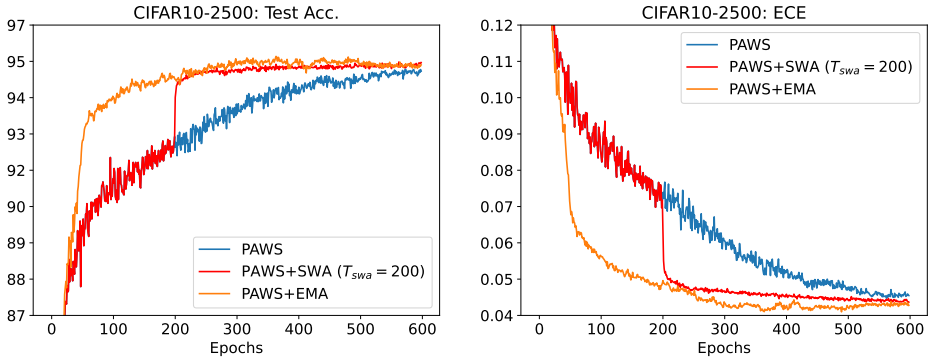

Figure 3: **Weight averaging techniques on PAWS for CIFAR-10 with 2500 labels** displays highly consistent relationships between increasing test accuracy and decreasing ECE.

7.2 BaM- is useful, but weight averaging is more effective when a selection metric is absent

Results are shown in Table 5; while BaM- gave some improvements over the baselines across all benchmarks, the improvements from weight averaging approaches are consistently larger. This is unsurprising, since BaM- relies on stochasticity only in the last layer while weight averaging aggregates over the entire model. Since PAWS does not use a selection metric and simply "accepts all unlabeled samples", the "selection purity" is absent and thus we investigate the issue of confirmation bias by tracking the accuracy and model calibration on a held-out set as shown in Fig. 3. Incorporating weight averaging results in better accuracies on the held-out set compared to the baseline at every stage of training. This suggests that even in the absence of a selection metric, proper uncertainty estimates (or improved model calibration) can lead to improved SSL performance. The effectiveness of improved uncertainty estimates is particularly evident through PAWS+SWA—when SWA was switched on (at $T_{\text{swa}} = 200$ epochs), model calibration was immediately improved (i.e. the ECE quickly dips) and the accuracy also correspondingly spikes. Furthermore, we also observe a significant improvement in convergence rates resulting from the weight averaging techniques, i.e. better accuracies can be attained in just a third of the number of training iterations of the baseline. Ablations on T_{swa} and momentum schedules are presented in Appendix G.3.

Effective also in large-scale datasets like ImageNet. A strength of representation learning SSL methods is in large-scale datasets like ImageNet, where PAWS outperforms selection metric based SSL methods (i.e. 73.9% vs 71.5% on the best performing selection metric based method, FixMatch for the ImageNet-10% benchmark). In this work, we use the 10% labels setting of Assran et al. (2021) with slight modifications to the default setting in order to fit on our system (see Appendix D.2 for implementation details) and explored weight averaging techniques SWA and EMA, given their effectiveness over BaM- in PAWS. Results are shown in Table 6, where our methods provide consistent improvements in calibration and test accuracy over the baseline.

Table 6: **ImageNet-10%** showing "Top-1 test accuracy (%) / ECE" for PAWS and our methods, trained for 200 epochs and fine-tuned with a linear head.

	IN-10%
PAWS	74.4 / 0.186
+SWA (ours)	74.5 / 0.186
+EMA (ours)	**75.1** / **0.183**

8 Conclusion and Broader Impact

Since confirmation bias is a fundamental problem in SSL, in this work we showed that it is imperative for the model to have proper uncertainty estimates, or be well-calibrated, to mitigate this problem. In particular, we empirically demonstrated that approximate Bayesian techniques such as a last Bayesian layer or weight averaging approaches can be used to improve a model's uncertainty estimates which can result in better model performance across a variety of SSL methods. We further underscore their importance in more challenging real-world datasets. We hope that our findings can motivate future research directions to incorporate techniques targeted for optimizing calibration during the development of new SSL methods. Furthermore, while the primary goal of improving calibration is to mitigate confirmation bias during pseudo-labeling, an

auxiliary benefit brought about by our approach is a better calibrated network, i.e. one that can better quantify its uncertainty, which is highly important for real-world applications. A potential limitation in our work lies in the use of ECE as a metric to measure calibration which, while commonly used across literature, are not free from flaws (Nixon et al., 2019). However, in our work, we empirically demonstrate that despite their flaws, the ECE metric still provides good correlations to measuring confirmation bias and test accuracy. We provide further discussion of the societal impact and ethical considerations of our work in Appendix I.

References

Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. Concrete problems in ai safety, 2016. URL https://arxiv.org/abs/1606.06565.

Eric Arazo, Diego Ortego, Paul Albert, Noel E. O'Connor, and Kevin McGuinness. Pseudo-labeling and confirmation bias in deep semi-supervised learning, 2019. URL https://arxiv.org/abs/1908.02983.

Mahmoud Assran, Mathilde Caron, Ishan Misra, Piotr Bojanowski, Armand Joulin, Nicolas Ballas, and Michael G. Rabbat. Semi-supervised learning of visual features by non-parametrically predicting view assignments with support samples. *CoRR*, abs/2104.13963, 2021. URL https://arxiv.org/abs/2104.13963.

Philip Bachman, Ouais Alsharif, and Doina Precup. Learning with pseudo-ensembles. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K.Q. Weinberger (eds.), *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014. URL https://proceedings.neurips.cc/paper/2014/file/66be31e4c40d676991f2405aaecc6934-Paper.pdf.

David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin A Raffel. Mixmatch: A holistic approach to semi-supervised learning. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/file/1cd138d0499a68f4bb72bee04bbec2d7-Paper.pdf.

Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural networks, 2015. URL https://arxiv.org/abs/1505.05424.

Glenn W. Brier. Verification of forecasts expressed in terms of probability. *Monthly Weather Review*, 78(1):1 – 3, 1950. doi: 10.1175/1520-0493(1950)078<0001:VOFEIT>2.0.CO;2. URL https://journals.ametsoc.org/view/journals/mwre/78/1/1520-0493_1950_078_0001_vofeit_2_0_co_2.xml.

Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Arechiga, and Tengyu Ma. Learning imbalanced datasets with label-distribution-aware margin loss, 2019. URL https://arxiv.org/abs/1906.07413.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. *CoRR*, abs/2104.14294, 2021. URL https://arxiv.org/abs/2104.14294.

Paola Cascante-Bonilla, Fuwen Tan, Yanjun Qi, and Vicente Ordonez. Curriculum labeling: Self-paced pseudo-labeling for semi-supervised learning. *CoRR*, abs/2001.06001, 2020. URL https://arxiv.org/abs/2001.06001.

Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey Hinton. Big self-supervised models are strong semi-supervised learners, 2020. URL https://arxiv.org/abs/2006.10029.

Morris H. DeGroot and Stephen E. Fienberg. The comparison and evaluation of forecasters. *Journal of the Royal Statistical Society. Series D (The Statistician)*, 32(1/2):12–22, 1983. ISSN 00390526, 14679884. URL http://www.jstor.org/stable/2987588.

Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning, 2015. URL https://arxiv.org/abs/1506.02142.

Andrew Gelman, John Carlin, Hal Stern, David Dunson, Aki Vehtari, and Donald Rubin. *Bayesian Data Analysis, Third Edition*. CRC press, 2021.

Yves Grandvalet and Yoshua Bengio. Semi-supervised learning by entropy minimization. In L. Saul, Y. Weiss, and L. Bottou (eds.), *Advances in Neural Information Processing Systems*, volume 17. MIT Press, 2004. URL https://proceedings.neurips.cc/paper/2004/file/96f2b50b5d3613adf9c27049b2a888c7-Paper.pdf.

Alex Graves. Practical variational inference for neural networks. In J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K.Q. Weinberger (eds.), *Advances in Neural Information Processing Systems*, volume 24. Curran Associates, Inc., 2011. URL https://proceedings.neurips.cc/paper/2011/file/7eb3c8be3d411e8ebfab08eba5f49632-Paper.pdf.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in Neural Information Processing Systems*, 33:21271–21284, 2020.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks, 2017. URL https://arxiv.org/abs/1706.04599.

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 Hal Daumé III and Aarti Singh (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 3897–3906. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/guo20i.html.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9729–9738, 2020.

Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry Vetrov, and Andrew Gordon Wilson. Averaging weights leads to wider optima and better generalization, 2018. URL https://arxiv.org/abs/1803.05407.

Diederik P. Kingma, Tim Salimans, and Max Welling. Variational dropout and the local reparameterization trick, 2015. URL https://arxiv.org/abs/1506.02557.

Samuli Laine and Timo Aila. Temporal ensembling for semi-supervised learning, 2016. URL https://arxiv.org/abs/1610.02242.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles, 2016. URL https://arxiv.org/abs/1612.01474.

Dong-hyun Lee. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks, 2013.

Charlotte Loh, Thomas Christensen, Rumen Dangovski, Samuel Kim, and Marin Soljacic. Surrogate- and invariance-boosted contrastive learning for data-scarce applications in science. *Nat Commun*, 13, 2022. URL https://doi.org/10.1038/s41467-022-31915-y.

Matthias Minderer, Josip Djolonga, Rob Romijnders, Frances Hubis, Xiaohua Zhai, Neil Houlsby, Dustin Tran, and Mario Lucic. Revisiting the calibration of modern neural networks. *CoRR*, abs/2106.07998, 2021. URL https://arxiv.org/abs/2106.07998.

Kevin P. Murphy. *Probabilistic Machine Learning: An introduction*. MIT Press, 2022. URL probml.ai.

Jeremy Nixon, Mike Dusenberry, Ghassen Jerfel, Timothy Nguyen, Jeremiah Liu, Linchuan Zhang, and Dustin Tran. Measuring calibration in deep learning, 2019. URL https://arxiv.org/abs/1904.01685.

John C. Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In *ADVANCES IN LARGE MARGIN CLASSIFIERS*, pp. 61–74. MIT Press, 1999.

Zhongzheng Ren, Raymond A. Yeh, and Alexander G. Schwing. Not all unlabeled data are equal: Learning to weight data in semi-supervised learning, 2020. URL https://arxiv.org/abs/2007.01293.

Mamshad Nayeem Rizve, Kevin Duarte, Yogesh S Rawat, and Mubarak Shah. In defense of pseudo-labeling: An uncertainty-aware pseudo-label selection framework for semi-supervised learning. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=-ODN6SbiUU.

Chuck Rosenberg, Martial Hebert, and Henry Schneiderman. Semi-supervised self-training of object detection models. In *2005 Seventh IEEE Workshops on Applications of Computer Vision (WACV/MOTION'05) - Volume 1*, volume 1, pp. 29–36, 2005. doi: 10.1109/ACVMOT.2005.107.

Mehdi Sajjadi, Mehran Javanmardi, and Tolga Tasdizen. Regularization with stochastic transformations and perturbations for deep semi-supervised learning. *CoRR*, abs/1606.04586, 2016. URL http://arxiv.org/abs/1606.04586.

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, 2020. URL https://arxiv.org/abs/2001.07685.

Jost Tobias Springenberg, Aaron Klein, Stefan Falkner, and Frank Hutter. Bayesian optimization with robust bayesian neural networks. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016. URL https://proceedings.neurips.cc/paper/2016/file/a96d3afec184766bfeca7a9f989fc7e7-Paper.pdf.

Antti Tarvainen and Harri Valpola. Weight-averaged consistency targets improve semi-supervised deep learning results. *CoRR*, abs/1703.01780, 2017. URL http://arxiv.org/abs/1703.01780.

Sunil Thulasidasan, Gopinath Chennupati, Jeff Bilmes, Tanmoy Bhattacharya, and Sarah Michalak. On mixup training: Improved calibration and predictive uncertainty for deep neural networks, 2019. URL https://arxiv.org/abs/1905.11001.

Max Welling and Yee Whye Teh. Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, ICML'11, pp. 681–688, Madison, WI, USA, 2011. Omnipress. ISBN 9781450306195.

Yeming Wen, Dustin Tran, and Jimmy Ba. Batchensemble: An alternative approach to efficient ensemble and lifelong learning. 2020. doi: 10.48550/ARXIV.2002.06715. URL https://arxiv.org/abs/2002.06715.

Andrew Gordon Wilson and Pavel Izmailov. Bayesian deep learning and a probabilistic perspective of generalization. *CoRR*, abs/2002.08791, 2020. URL https://arxiv.org/abs/2002.08791.

Qizhe Xie, Zihang Dai, Eduard H. Hovy, Minh-Thang Luong, and Quoc V. Le. Unsupervised data augmentation. *CoRR*, abs/1904.12848, 2019a. URL http://arxiv.org/abs/1904.12848.

Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V. Le. Self-training with noisy student improves imagenet classification, 2019b. URL https://arxiv.org/abs/1911.04252.

Yi Xu, Lei Shang, Jinxing Ye, Qi Qian, Yu-Feng Li, Baigui Sun, Hao Li, and Rong Jin. Dash: Semi-supervised learning with dynamic thresholding, 2021. URL https://arxiv.org/abs/2109.00650.

David Yarowsky. Unsupervised word sense disambiguation rivaling supervised methods. In *33rd Annual Meeting of the Association for Computational Linguistics*, pp. 189–196, Cambridge, Massachusetts, USA, June 1995. Association for Computational Linguistics. doi: 10.3115/981658.981684. URL https://aclanthology.org/P95-1026.

Bianca Zadrozny and Charles Elkan. Transforming classifier scores into accurate multiclass probability estimates. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '02, pp. 694–699, New York, NY, USA, 2002. Association for Computing Machinery. ISBN 158113567X. doi: 10.1145/775047.775151. URL https://doi.org/10.1145/775047.775151.

Bowen Zhang, Yidong Wang, Wenxin Hou, Hao Wu, Jindong Wang, Manabu Okumura, and Takahiro Shinozaki. Flexmatch: Boosting semi-supervised learning with curriculum pseudo labeling, 2021. URL https://arxiv.org/abs/2110.08263.

Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization, 2017. URL https://arxiv.org/abs/1710.09412.

Yang Zou, Zhiding Yu, BVK Kumar, and Jinsong Wang. Unsupervised domain adaptation for semantic segmentation via class-balanced self-training. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 289–305, 2018.