Pseudo-Labeling and Confirmation Bias in Deep Semi-Supervised Learning

Eric Arazo, Diego Ortego, Paul Albert, Noel E. O'Connor & Kevin McGuinness
Insight Centre for Data Analytics, Dublin City University (DCU)
{eric.arazo, diego.ortego}@insight-centre.org

***Abstract*—Semi-supervised learning, i.e. jointly learning from labeled and unlabeled samples, is an active research topic due to its key role on relaxing human supervision. In the context of image classification, recent advances to learn from unlabeled samples are mainly focused on consistency regularization methods that encourage invariant predictions for different perturbations of unlabeled samples. We, conversely, propose to learn from unlabeled data by generating soft pseudo-labels using the network predictions. We show that a naive pseudo-labeling overfits to incorrect pseudo-labels due to the so-called confirmation bias and demonstrate that mixup augmentation and setting a minimum number of labeled samples per mini-batch are effective regularization techniques for reducing it. The proposed approach achieves state-of-the-art results in CIFAR-10/100, SVHN, and Mini-ImageNet despite being much simpler than other methods. These results demonstrate that pseudo-labeling alone can outperform consistency regularization methods, while the opposite was supposed in previous work. Source code is available at https://git.io/fjQsC.**

I. Introduction

Convolutional neural networks (CNNs) have become the dominant approach in computer vision [1–4]. To best exploit them, vast amounts of labeled data are required. Obtaining such labels, however, is not trivial, and the research community is exploring alternatives to alleviate this [5–7].

Knowledge transfer via deep domain adaptation [8] is a popular alternative that seeks to learn transferable representations from source to target domains by embedding domain adaptation in the learning pipeline. Other approaches focus exclusively on learning useful representations from scratch in a target domain when annotation constraints are relaxed [6, 9, 10]. Semi-supervised learning (SSL) [6] focuses on scenarios with sparsely labeled data and extensive amounts of unlabeled data; learning with label noise [9] seeks robust learning when labels are obtained automatically and may not represent the image content; and self-supervised learning [10] uses data supervision to learn from unlabeled data in a supervised manner. This paper focuses on SSL for image classification, a recently very active research area [11].

SSL is a transversal task for different domains including images [6], audio [12], time series [13], and text [14]. Recent approaches in image classification primarily focus on exploiting the consistency in the predictions for the same sample under different perturbations (consistency regularization) [11, 15], while other approaches directly generate labels for the unlabeled data to guide the learning process (pseudo-labeling) [16, 17]. These two alternatives differ importantly in the mechanism they use to exploit unlabeled samples. Consistency regularization and pseudo-labeling approaches apply different strategies such as a warm-up phase using labeled data [17, 18], uncertainty weighting [11, 19], adversarial attacks [20, 21], or graph-consistency [17, 22]. These strategies deal with confirmation bias [11, 18], also known as noise accumulation [12]. This bias stems from using incorrect predictions on unlabeled data for training in subsequent epochs and, thereby increasing confidence in incorrect predictions and producing a model that will tend to resist new changes.

This paper explores pseudo-labeling for semi-supervised deep learning from the network predictions and shows that, contrary to previous attempts on pseudo-labeling [6, 17, 19], simple modifications to prevent confirmation bias lead to state-of-the-art performance without adding consistency regularization strategies. We adapt the approach proposed by Tanaka et al. [23] in the context of label noise and apply it exclusively on unlabeled samples. Experiments show that this naive pseudo-labeling is limited by confirmation bias as prediction errors are fit by the network. To deal with this issue, we propose to use mixup augmentation [24] as an effective regularization that helps calibrate deep neural networks [25] and, therefore, alleviates confirmation bias. We find that mixup alone does not guarantee robustness against confirmation bias when reducing the amount of labeled samples or using certain network architectures (see Subsection IV-D), and show that, when properly introduced, dropout regularization [26] and data augmentation mitigates this issue. Our purely pseudo-labeling approach achieves state-of-the-art results (see Subsection IV-E) without requiring multiple networks [11, 18, 21, 27], nor does it require over a thousand epochs of training to achieve peak performance in every dataset [28, 29], nor needs many (ten) forward passes for each sample [11]. Compared to other pseudo-labeling approaches, the proposed approach is simpler in that it does not require graph construction and diffusion [17] or combination with consistency regularization methods [19], but still achieves state-of-the-art results.

II. Related work

This section reviews closely related SSL methods, i.e. those using deep learning with mini-batch optimization over large image collections. Previous work on deep SSL differ in whether they use consistency regularization or pseudo-labeling to learn

This work was supported by Science Foundation Ireland (SFI) under grant numbers SFI/15/SIRG/3283 and SFI/12/RC/2289_P2.

from the unlabeled set [17], while they all share the use of a cross-entropy loss (or similar) on labeled data.

*a) **Consistency regularization**:* Imposes that the same sample under different perturbations must produce the same output. This idea was used in [15] where they apply randomized data augmentation, dropout, and random max-pooling while forcing softmax predictions to be similar. A similar idea is applied in [30], which also extends the perturbation to different epochs, i.e. the current prediction for a sample has to be similar to an ensemble of predictions of the same sample in the past. Here the different perturbations come from networks at different states, dropout, and data augmentation. In [18], the temporal ensembling method is interpreted as a teacher-student problem where the network is both a teacher that produces targets for the unlabeled data as a temporal ensemble, and a student that learns the generated targets by imposing the consistency regularization. [18] naturally re-defines the problem to deal with confirmation bias by separating the teacher and the student. The teacher is defined as a different network with similar architecture whose parameters are updated as an exponential moving average of the student network weights. This method is extended in [11], where they apply an uncertainty weight over the unlabeled samples to learn from the unlabeled samples with low uncertainty (i.e. entropy of the predictions for each sample under random perturbations). Additionally, Miyato et al. [20] use virtual adversarial training to carefully introduce perturbations to data samples as adversarial noise and later impose consistency regularization on the predictions. More recently, Luo et al. [22] propose to use a contrastive loss on the predictions as a regularization that forces predictions to be similar (different) when they are from the same (different) class. This method extends the consistency regularization previously considered only in-between the same data samples to in-between different samples. Their method can naturally be combined with [18] or [20] to boost their performance. Similarly, Verma et al. [27] propose interpolation consistency training, a method inspired by [24] that encourage predictions at interpolated unlabeled samples to be consistent with the interpolated predictions of individual samples. Also, authors in [29] apply consistency regularization by guessing low-entropy labels, generating data-augmented unlabeled examples and mixing labeled and unlabeled examples using mixup [24]. Both [27] and [29] adopt [18] to estimate the targets used in the consistency regularization.

Co-training [21] uses two (or more) networks trained simultaneously to agree on their predictions (consistency regularization) and disagree on their errors. Errors are defined as different predictions when exposed to adversarial attacks, thus forcing different networks to learn complementary representations for the same samples. Recently, Chen et al. [31] measure the consistency between the current prediction and an additional prediction for the same sample given by an external memory module that keeps track of previous representations. They additionally introduce an uncertainty weighting of the consistency term to reduce the contribution of uncertain predictions. Consistency regularization methods such as [18, 20, 30] have all been shown to benefit from stochastic weight averaging method [28], that averages network parameters at different training epochs to move the SGD solution on borders of flat loss regions to their center and improve generalization.

*b) **Pseudo-labeling**:* Seeks the generation of labels or pseudo-labels for unlabeled samples to guide the learning process. An early attempt at pseudo-labeling proposed in [16] uses the network predictions as labels. However, they constrain the pseudo-labeling to a fine-tuning stage, i.e. there is a pre-training or warm-up to initialize the network. A recent pseudo-labeling approach proposed in [19] uses the network class prediction as hard labels for the unlabeled samples. They also introduce an uncertainty weight for each sample loss, it being higher for samples that have distant k-nearest neighbors in the feature space. They further include a loss term to encourage intra-class compactness and inter-class separation, and a consistency term between samples with different perturbations. Improved results are reported in combination with [18]. Finally, a recently published work [17] implements pseudo-labeling through graph-based label propagation. The method alternates between two steps: training from labeled and pseudo-labeled data and using the representations of the network to build a nearest neighbor graph where label propagation is applied to refine hard pseudo-labels. They further add an uncertainty score for every sample (softmax prediction entropy based) and class (class population based) to deal, respectively, with the unequal confidence in network predictions and class-imbalance.

III. Pseudo-labeling

We formulate SSL as learning a model $h_\theta(x)$ from a set of N training samples $\mathcal{D}$. These samples are split into the unlabeled set $\mathcal{D}_u = \{x_i\}_{i=1}^{N_u}$ and the labeled set $\mathcal{D}_l = \{(x_i, y_i)\}_{i=1}^{N_l}$, being $y_i \in \{0,1\}^C$ the one-hot encoding label for C classes corresponding to x_i and $N = N_l + N_u$. In our case, h_θ is a CNN and θ represents the model parameters (weights and biases). As we seek to perform pseudo-labeling, we assume that a pseudo-label $\tilde{y}$ is available for the N_u unlabeled samples. We can then reformulate SSL as training using $\tilde{\mathcal{D}} = \{(x_i, \tilde{y}_i)\}_{i=1}^{N}$, being $\tilde{y} = y$ for the N_l labeled samples.

The CNN parameters θ can be optimized using categorical cross-entropy:

$$\ell^*(\theta) = -\sum_{i=1}^{N} \tilde{y}_i^T \log\left(h_\theta(x_i)\right), \tag{1}$$

where $h_\theta(x)$ are the softmax probabilities produced by the model and $\log(\cdot)$ is applied element-wise. A key decision is how to generate the pseudo-labels $\tilde{y}$ for the N_u unlabeled samples. Previous approaches have used hard pseudo-labels (i.e. one-hot vectors) directly using the network output class [16, 19] or the class estimated using label propagation on a nearest neighbor graph [17]. We adopt the former approach, but use soft pseudo-labels, as we have seen this outperforms hard labels, confirming the observations noted in [23] in the context of relabeling when learning with label noise. In particular, we store the softmax predictions $h_\theta(x_i)$ of the network in

every mini-batch of an epoch and use them to modify the soft pseudo-label $\tilde{y}$ for the N_u unlabeled samples at the end of every epoch. We proceed as described from the second to the last training epoch, while in the first epoch we use the softmax predictions for the unlabeled samples from a model trained in a 10 epochs warm-up phase using the labeled data subset $\mathcal{D}_u$.

We use the two regularizations applied in [23] to improve convergence. The first regularization deals with the difficulty of converging at early training stages when the network's predictions are mostly incorrect and the CNN tends to predict the same class to minimize the loss. Assignment of all samples to a single class is discouraged by adding:

$$R_A = \sum_{c=1}^{C} p_c \log\left(\frac{p_c}{\overline{h}_c}\right), \tag{2}$$

where p_c is the prior probability distribution for class c and $\overline{h}_c$ denotes the mean softmax probability of the model for class c across all samples in the dataset. As in [23], we assume a uniform distribution $p_c = 1/C$ for the prior probabilities (R_A stands for all classes regularization) and approximate $\overline{h}_c$ using mini-batches. The second regularization is needed to concentrate the probability distribution of each soft pseudo-label on a single class, thus avoiding the local optima in which the network might get stuck due to a weak guidance:

$$R_H = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C} h_\theta^c(x_i)\log\left(h_\theta^c(x_i)\right), \tag{3}$$

where $h_\theta^c(x_i)$ denotes the c class value of the softmax output $h_\theta(x_i)$ and again using mini-batches (i.e. N is replaced by the mini-batch size) to approximate this term. This second regularization is the average per-sample entropy (R_H stands for entropy regularization), a well-known regularization in SSL [32]. Finally, the total semi-supervised loss is:

$$\ell = \ell^* + \lambda_A R_A + \lambda_H R_H, \tag{4}$$

where λ_A and λ_H control the contribution of each regularization term (see Subsection IV-C for a study of these hyperparameters). We stress that this pseudo-labeling approach adapted from [23] is far from the state-of-the-art for SSL (see Subsection IV-B), and are the mechanisms proposed in Subsection III-A which make pseudo-labeling a suitable alternative.

A. Confirmation bias

Network predictions are, of course, sometimes incorrect. This situation is reinforced when incorrect predictions are used as labels for unlabeled samples, as it is the case in pseudo-labeling. Overfitting to incorrect pseudo-labels predicted by the network is known as confirmation bias. It is natural to think that reducing the confidence of the network on its predictions might alleviate this problem and improve generalization. Recently, mixup data augmentation [24] introduced a strong regularization technique that combines data augmentation with label smoothing, which makes it potentially useful to deal with this bias. Mixup trains on convex combinations of sample pairs (x_p and x_q) and corresponding labels (y_p and y_q):

$$x = \delta x_p + (1-\delta)x_q, \tag{5}$$

$$y = \delta y_p + (1-\delta)y_q, \tag{6}$$

where $\delta \in \{0,1\}$ is randomly sampled from a beta distribution $\mathcal{B}e\left(\alpha,\beta\right)$, with $\alpha = \beta$ (e.g. $\alpha = 1$ uniformly selects δ). This combination regularizes the network to favor linear behavior in-between training samples, reducing oscillations in regions far from them. Additionally, Eq. 6 can be re-interpreted in the loss as $\ell^* = \delta\ell_p^* + (1-\delta)\ell_q^*$, thus re-defining the loss ℓ^* used in Eq. 4 as:

$$\ell^* = -\sum_{i=1}^{N} \delta\left[\tilde{y}_{i,p}^T \log\left(h_\theta(x_i)\right)\right] + (1-\delta)\left[\tilde{y}_{i,q}^T \log\left(h_\theta(x_i)\right)\right]. \tag{7}$$

As shown in [25], overconfidence in deep neural networks is a consequence of training on hard labels and it is the label smoothing effect from randomly combining y_p and y_q during mixup training that reduces prediction confidence and improves model calibration. In the semi-supervised context with pseudo-labeling, using soft-labels and mixup reduces overfitting to model predictions, which is especially important for unlabeled samples whose predictions are used as soft-labels. Note that training with mixup generates softmax outputs $h_\theta(x)$ for mixed inputs x, thus requiring a second forward pass with the original images to compute unmixed predictions.

Mixup data augmentation alone may be insufficient to deal with confirmation bias when few labeled examples are provided. For example, when training with 500 labeled samples in CIFAR-10 and mini-batch size of 100, just 1 clean sample per batch is seen, which is especially problematic at early stages of training where little correct guidance is provided. Oversampling the labelled examples by setting a minimum number of labeled samples per mini-batch k (as done in other works [17, 18, 29, 31]) provides a constant reinforcement with correct labels during training, reducing confirmation bias and helping to produce better pseudo-labels.

The effect of this oversampling can be understood by splitting the total loss (Eq. 1) into two terms, the first depending on the labeled examples and the second on the unlabelled:

$$\ell^* = N_l\overline{\ell}_l + N_u\overline{\ell}_u, \tag{8}$$

where N_l and N_u are the number of labelled and unlabelled samples, and the $\overline{\ell}_l = \frac{1}{N_l}\sum_{i=1}^{N_l}\ell_l^{(i)}$ is the average loss for labeled samples and similarly $\overline{\ell}_u$ for the unlabeled samples. The first term is a data loss on the labeled samples and the second can be interpreted as a regularization term that encourages the network to fit the pseudo-labels of the unlabeled samples. When few labeled samples are available, $N_l << N_u$, the regularization term dominates the loss, i.e. fitting the pseudo-labels is weighted far higher than fitting the labelled samples. This can be overcome either by upweighting the the first term or by oversampling labeled samples. We use the latter strategy as it results in more frequent parameter updates to satisfy the first

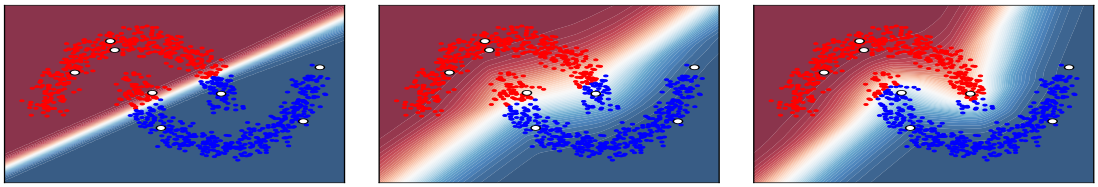

Fig. 1. Pseudo-labeling in the "two moons" data (4 labels/class) for 1000 samples. From left to right: no mixup, mixup, and mixup with a minimum number of labeled samples per mini-batch. We use an NN classifier with one hidden layer with 50 hidden units as in [20].

term, rather than larger magnitude updates. Subsections IV-B and IV-D experimentally show that mixup, a minimum number of samples per mini-batch, and other techniques (dropout and data augmentation) reduce confirmation bias and make pseudo-labeling an effective alternative to consistency regularization.

IV. Experimental work

A. Datasets and training

We use four image classification datasets, CIFAR-10/100 [33], SVHN [34] and Mini-ImageNet [35], to validate our approach. Part of the training images are labeled and the remaining are unlabeled. Following [6], we use a validation set of 5K samples for CIFAR-10/100 for studying hyperparameters in Subsections IV-B and IV-D. However, as done in [28], we add the 5K samples back to the training set for comparisons in Subsection IV-E, where we report test results (model from the best epoch).

*a) **CIFAR-10, CIFAR-100, and SVHN**:* These datasets contain 10, 100, and 10 classes respectivelly, with 50K color images for training and 10K for testing in CIFAR-10/100 and 73257 images for training and 26032 for testing in SVHN. The three datasets have resolution 32×32. We perform experiments with a number of labeled images N_l = 0.25K, 0.5K, and 1K for SVHN and N_l = 0.25K, 0.5K, 1K, and 4K (4K and 10K) for CIFAR-10 (CIFAR-100). We use the well-known "13-CNN" architecture [28] for CIFAR-10/100 and SVHN. We also experiment with a Wide ResNet-28-2 (WR-28) [6] and a PreAct ResNet-18 (PR-18) [24] in Subsection IV-D to study the generalization to different architectures.

*b) **Mini-ImageNet**:* We emulate the semi-supervised learning setup Mini-ImageNet [35] (a subset of the well-known ImageNet [36] dataset) used in [17]. Train and test sets of 100 classes and 600 color images per class with resolution 84 × 84 are selected from ImageNet, as in [37]. 500 (100) images per-class are kept for train (test) splits. The train and test sets therefore contain 50k and 10k images. As with CIFAR-100, we experiment with a number of labeled images N_l = 4K and 10K. Following [17], we use a ResNet-18 (RN-18) architecture [38].

*c) **Hyperparameters**:* We use the typical configuration for CIFAR-10/100 and SVHN [30], and the same for Mini-ImageNet. Image normalization using dataset mean and standard deviation and subsequent data augmentation [30] by random horizontal flips and 2 (6) pixel translations for CIFAR and SVHN (Mini-ImageNet). Additionally, color jitter is applied as in [39] in Subsections IV-D and IV-E for higher robustness against confirmation bias. We train using SGD with momentum of 0.9, weight decay of 10^{-4}, and batch size of 100. Training always starts with a high learning rate (0.1 in CIFAR and SVHN, and 0.2 in Mini-ImageNet), dividing it by ten twice during training. We train for CIFAR and Mini-ImageNet 400 epochs (reducing learning rate in epochs 250 and 350) and use 10 epoch warm-up with labeled data, while for SVHN we train 150 epochs (reducing learning rate in epochs 50 and 100) and use a longer warm-up of 150 epochs to start the pseudo-labeling with good predictions and leading to reliable convergence (experiments in CIFAR-10 with longer warm-up provided results in the same error range already reported). We do not attempt careful tuning of the regularization weights λ_A and λ_H and just set them to 0.8 and 0.4 as done in [23] (see Subsection IV-C for an ablation study of these parameters). When using dropout, it is introduced between consecutive convolutional layers of ResNet blocks in WR-28, PR-18, and RN-18, while for 13-CNN we introduce it as in [30]. Following [28][1], we use weight normalization [40] in all networks.

TABLE I

Confirmation bias alleviation using mixup and a minimum number of k labeled samples per mini-batch. Top: Validation error for naive pseudo-labeling without mixup (C), mixup (M), and alternatives with minimum k. Bottom: Study of the effect of k on the validation error.

	CIFAR-10		CIFAR-100
Labeled images	500	4000	4000
C	52.44	11.40	48.54
C* ($k = 16$)	35.08	10.90	46.60
M	32.10	7.16	41.80
M *($k = 16$)	**13.68**	**6.90**	**38.78**

	CIFAR-10		CIFAR-100
Labeled images	500	4000	4000
$k = 8$	**13.14**	7.18	42.32
$k = 16$	13.68	**6.90**	**38.78**
$k = 32$	14.58	7.06	39.62
$k = 64$	19.40	8.20	46.28

B. Effect of mixup on confirmation bias

This section demonstrates that carefully regularized pseudo-labeling is a suitable alternative for SSL. Figure 1 illustrates our approach on the "two moons" toy data. Figure 1 (left) shows the limitations of a naive pseudo-labeling adapted from [23], which fails to adapt to the structure in the unlabelled examples

[1] https://github.com/benathi/fastswa-semi-sup

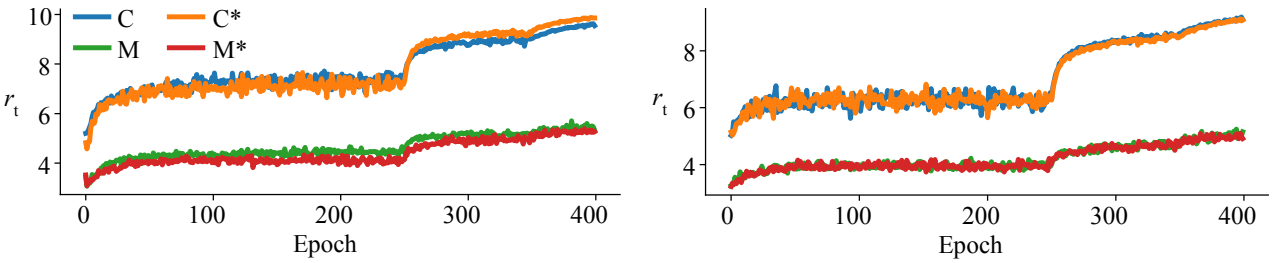

Fig. 2. Example of certainty of incorrect predictions r_t during training when using 500 (left) and 4000 (right) labeled images in CIFAR-10. Moving from cross-entropy (C) to mixup (M) reduces r_t, whereas adding a minimum number of samples per mini-batch (*) also helps in 500 labels, where M* (with slightly lower r_t than M) is the only configuration that converges, as shown in Table I (top).

TABLE II
VALIDATION ERROR FOR DIFFERENT VALUES OF THE α PARAMETER FROM MIXUP, λ_A, AND λ_H. BOLD INDICATES LOWEST ERROR. UNDERLINED VALUES INDICATE THE RESULTS OF THE CONFIGURATION USED.

Labeled images:	500				4000			
α	0.1	1	4	8	0.1	1	4	8
	23.18	<u>13.68</u>	**10.60**	11.04	8.58	<u>6.90</u>	**6.56**	6.68
λ_A/λ_H	0.1	0.4	0.8	2	0.1	0.4	0.8	2
0.1	22.94	29.64	60.76	83.96	7.22	**6.88**	7.74	33.98
0.4	20.92	**12.88**	17.62	38.40	7.18	6.96	7.18	8.82
0.8	23.50	<u>13.68</u>	14.72	25.92	7.24	<u>6.90</u>	7.18	8.78
2	31.30	14.80	14.62	23.40	8.16	7.28	7.40	8.64

and results in a linear decision boundary. Figure 1 (middle) shows the effect of mixup, which alleviates confirmation bias to better model the structure and gives a smoother boundary. Figure 1 (right) shows that combining mixup with a minimum number of labeled samples k per mini-batch improves the semi-supervised decision boundary.

Naive pseudo-labeling leads to overfitting the network predictions and high training accuracy in CIFAR-10/100. Table I (top) reports mixup effect in terms of validation error. Naive pseudo-labeling leads to an error of 11.40/48.54 for CIFAR-10/100 when training with cross-entropy (C) loss for 4000 labels. This error can be greatly reduced when using mixup (M) to 7.16/41.80. However, when further reducing the number of labels to 500 in CIFAR-10, M is insufficient to ensure low-error (32.10). We propose to set a minimum number of samples k per mini-batch to tackle the problem. Table I (bottom) studies this parameter k when combined with mixup, showing that 16 samples per mini-batch works well for both CIFAR-10 and CIFAR-100, dramatically reducing error in all cases (e.g. in CIFAR-10 for 500 labels error is reduced from 32.10 to 13.68). Confirmation bias causes a dramatic increase in the certainty of incorrect predictions during training. To demonstrate this behavior we compute the average cross-entropy of the softmax output with a uniform $\mathcal{U}$ across the classes in every epoch t for all incorrectly predicted samples $\{x_{m_t}\}_{m_t=1}^{M_t}$ as: $r_t = -\frac{1}{M_t}\sum_{m_t=1}^{M_t}\mathcal{U}^T \log\left(h_\theta(x_{m_t})\right)$. Figure 2 shows that mixup and minimum k are effective regularizers for reducing r_t, i.e. confirmation bias is reduced. We also experimented with using label noise regularizations [41], but setting a minimum k proved more effective.

C. Extended hyperparameters study

This subsection studies the effect of α, λ_A, and λ_H hyperparameters of our pseudo-labeling approach. Table II reports the validation error in CIFAR-10 using 500 and 4000 labels for, respectively, α and λ_A and λ_H . Note that we keep the same configuration used in Subsection IV-B with $k = 16$, i.e. no dropout or additional data augmentation is used. Table II results suggest that $\alpha = 4$ and $\alpha = 8$ values might further improve the reported results using $\alpha = 1$. However, we experimented on CIFAR-10 with 500 labels using the final configuration (adding dropout and additional data augmentation) and observed marginal differences (8.54 with $\alpha = 4$, which is within the error range of the 8.80 ± 0.45 obtained with $\alpha = 1$) shown in Table IV, thus suggesting that stronger mixup regularization might not be additive to dropout and extra data augmentation in our case. Table II shows that our configuration ($\lambda_A = 0.8$ and $\lambda_H = 0.4$) adopted from [23] is very close to the best performance in this experiment where marginal improvements are achieved. More careful hyperparameter tuning might slightly improve the results here, but the default configuration is already good and generalizes well across datasets.

D. Generalization to different architectures

There are examples in the recent literature [42] where moving from one architecture to another changes which methods appear to have a higher potential. Kolesnikov et al. [42] show that skip-connections in ResNet architectures play a key role on the quality of learned representations, while most approaches in previous literature were systematically evaluated using AlexNet

TABLE III
VALIDATION ERROR ACROSS ARCHITECTURES IS STABILIZED USING DROPOUT p AND DATA AUGMENTATION (A).

	M*		M* ($p = 0.1$)		M* ($p = 0.3$)		M* ($p = 0.1$, A)	
Labeled images	500	4000	500	4000	500	4000	500	4000
13-layer	13.68	6.90	12.62	6.58	11.94	6.66	**9.16**	**6.22**
WR-28	29.50	**6.40**	14.14	7.06	30.56	11.44	**10.94**	6.74
PR-18	**13.90**	**5.94**	14.78	5.90	14.78	6.62	14.96	6.32

TABLE IV
TEST ERROR IN CIFAR-10/100 FOR THE PROPOSED APPROACH USING THE 13-CNN NETWORK. (*) DENOTES THAT WE HAVE RUN THE ALGORITHM. BOLD INDICATES LOWEST ERROR. WE REPORT AVERAGE AND STANDARD DEVIATION OF 3 RUNS WITH DIFFERENT LABELED/UNLABELED SPLITS.

	CIFAR-10			CIFAR-100	
Labeled images	500	1000	4000	4000	10000
Supervised (C)*	43.64 ± 1.21	34.83 ± 1.15	19.26 ± 0.26	54.49 ± 0.53	41.14 ± 0.26
Supervised (M)*	37.60 ± 0.65	28.59 ± 1.21	15.94 ± 0.26	52.70 ± 0.28	39.42 ± 0.37
Consistency regularization methods					
Π model	-	-	12.36 ± 0.31	-	39.19 ± 0.36
TE	-	-	12.16 ± 0.24	-	38.65 ± 0.51
MT	27.45 ± 2.64	19.04 ± 0.51	11.41 ± 0.25	45.36 ± 0.49	36.08 ± 0.51
Π model-SN	-	21.23 ± 1.27	11.00 ± 0.13	-	37.97 ± 0.29
MA-DNN	-	-	11.91 ± 0.22	-	34.51 ± 0.61
Deep-Co	-	-	9.03 ± 0.18	-	38.77 ± 0.28
MT-TSSDL	-	18.41 ± 0.92	9.30 ± 0.55	-	-
MT-LP	24.02 ± 2.44	16.93 ± 0.70	10.61 ± 0.28	43.73 ± 0.20	35.92 ± 0.47
MT-CCL	-	16.99 ± 0.71	10.63 ± 0.22	-	34.81 ± 0.52
MT-fast-SWA	-	15.58 ± 0.12	9.05 ± 0.21	-	34.10 ± 0.31
ICT	-	15.48 ± 0.78	7.29 ± 0.02	-	-
Pseudo-labeling methods					
TSSDL	-	21.13 ± 1.17	10.90 ± 0.23	-	-
LP	32.40 ± 1.80	22.02 ± 0.88	12.69 ± 0.29	46.20 ± 0.76	38.43 ± 1.88
Ours*	**8.80 ± 0.45**	**6.85 ± 0.15**	**5.97 ± 0.15**	**37.55 ± 1.09**	**32.15 ± 0.50**

[43]. Ulyanov et al. [44] showed that different architectures lead different and useful image priors, highlighting the importance of exploring different networks. We, therefore, test our method with two more architectures: a Wide ResNet-28-2 (WR-28) [45] typically used in SSL [6] (1.5M parameters) and a PreAct ResNet-18 (PR-18) [46] used in the context of label noise [24] (11M parameters). Table III presents the results for the 13-CNN (AlexNet-type) and these network architectures (ResNet-type). Our pseudo-labeling with mixup and $k = 16$ (M*) works well for 4000 and 500 labels across architectures, except for 500 labels for WR-28 where there is large error increase (29.50). This is due to a stronger confirmation bias in which labeled samples are not properly learned, while incorrect pseudo-labels are fit. Interestingly, PR-18 (11M) is more robust to confirmation bias than WR-28 (1.5M), while the 13-layer network (3M) has fewer parameters than PR-18 and achieves better performance. This suggests that the network architecture plays an important role, being a relevant prior for SSL with few labels.

We found that dropout [26] and data augmentation help to achieve good performance across all architectures. Table III shows that dropout $p = 0.1, 0.3$ helps in achieving better convergence in CIFAR-10, whereas adding color jitter as additional data augmentation (details in Subsection IV-A) further contributes to error reduction. Note that the quality of pseudo-labels is key, so it is essential to disable dropout to prevent corruption when computing these in the second forward pass. We similarly disable data augmentation in the second forward pass, which consistently improves performance. This configuration is used for comparison with the state-of-the-art in Subsection IV-E.

E. Comparison with the state-of-the-art

We compare our pseudo-labeling approach against related work that makes use of the 13-CNN [18] in CIFAR-10/100: Π model [30], TE [30], MT [18], Π model-SN [22], MA-DNN [31], Deep-Co [21], TSSDL [19], LP [17], CCL [11], fast-SWA [28] and ICT [27]. Tables IV and V divide methods into those based on consistency regularization and pseudo-labeling. Note that we include pseudo-labeling approaches combined with consistency regularization ones (e.g. MT) in the consistency regularization set. The proposed approach clearly outperforms consistency regularization methods, as well as other purely pseudo-labeling approaches and their combination with consistency regularization methods in CIFAR-10/100 and SVHN. These results demonstrate the generalization of the proposed approach compared to other methods that fail when decreasing the number of labels. Furthermore, Table VI (left)

TABLE V
TEST ERROR IN SVHN FOR THE PROPOSED APPROACH USING THE 13-CNN NETWORK. (*) DENOTES THAT WE HAVE RUN THE ALGORITHM. BOLD INDICATES LOWEST ERROR. WE REPORT AVERAGE AND STANDARD DEVIATION OF 3 RUNS WITH DIFFERENT LABELED/UNLABELED SPLITS.

Labeled images	250	500	1000
Supervised (C)*	43.60±3.35	22.67±2.80	13.32±0.89
Supervised (M)*	53.15±6.54	20.74±0.80	11.66±0.17
Consistency regularization methods			
Π model	9.69 ± 0.92	6.83 ± 0.66	4.95 ± 0.26
TE	-	5.12 ± 0.13	4.42 ± 0.16
MT	4.35 ± 0.50	4.18 ± 0.27	3.95 ± 0.19
Π model-SN	5.07 ± 0.25	4.52 ± 0.30	3.82 ± 0.25
MA-DNN	-	-	4.21 ± 0.12
Deep-Co	-	-	3.61 ± 0.15
MT-TSSDL	4.09 ± 0.42	3.90 ± 0.27	**3.35 ± 0.27**
ICT	4.78 ± 0.68	4.23 ± 0.15	3.89 ± 0.04
Pseudo-labeling methods			
TSSDL	5.02 ± 0.26	4.32 ± 0.30	3.80 ± 0.27
Ours*	**3.66 ± 0.12**	**3.64 ± 0.04**	3.55 ± 0.08

demonstrates that the proposed approach successfully scales to higher resolution images, obtaining an over 10 point margin on the best related work in Mini-ImageNet. Note that all supervised baselines are reported using the same data augmentation and dropout as in the proposed pseudo-labeling.

Table VI (right) compares our pseudo-labeling approach against recent consistency regularization approaches that use mixup. We achieve better performance than ICT [27], while being competitive with MM [29] for 500 and 4000 labels using WR-28. Regarding PR-18, we converge to reasonable performance for 4000 and 500 labels, whereas for 250 we do not. Finally, the 13-CNN robustly converges even for 250 labels where we obtain 9.37 test error. Therefore, these results suggest that it is worth exploring the relationship between number of labels, dataset complexity and architecture type. As shown in Subsection IV-D, dropout and additional data augmentation help with 500 labels/class across architectures, but are insufficient for 250 labels. Better data augmentation [47] or self-supervised pre-training [48] might overcome this challenge. However, it is already interesting that a straightforward modification of pseudo-labeling, designed to tackle confirmation bias, gives a competitive semi-supervised learning approach, without any consistency regularization, and future work should take this into account.

V. CONCLUSIONS

This paper presented a semi-supervised learning approach for image classification based on pseudo-labeling. We proposed to directly use the network predictions as soft pseudo-labels for unlabeled data together with mixup augmentation, a minimum number of labeled samples per mini-batch, dropout and data augmentation to alleviate confirmation bias. This conceptually simple approach outperforms related work in four datasets, demonstrating that pseudo-labeling is a suitable alternative to the dominant approach in recent literature: consistency-regularization. The proposed approach is, to the best of our knowledge, both simpler and more accurate than most recent approaches. Future work should explore SSL in class-unbalanced and large-scale datasets and synergies of pseudo-labelling and consistency regularization.

REFERENCES

[1] T. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal Loss for Dense Object Detection," in *IEEE International Conference on Computer Vision (ICCV)*, 2017.

[2] W. Liu, R. Lin, Z. Liu, L. Liu, Z. Yu, B. Dai, and L. Song, "Learning towards Minimum Hyperspherical Energy," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

[3] C. Kim, F. Li, and J. Rehg, "Multi-object Tracking with Neural Gating Using Bilinear LSTM," in *European Conference on Computer Vision (ECCV)*, 2018.

[4] S. Xie, C. Sun, J. Huang, Z. Tu, and K. Murphy, "Rethinking Spatiotemporal Feature Learning: Speed-Accuracy Trade-offs in Video Classification," in *European Conference on Computer Vision (ECCV)*, September 2018.

[5] W. Li, L. Wang, W. Li, E. Agustsson, and L. Van Gool, "WebVision Database: Visual Learning and Understanding from Web Data," *arXiv: 1708.02862*, 2017.

[6] A. Oliver, A. Odena, C. Raffel, E. Cubuk, and I. Goodfellow, "Realistic Evaluation of Deep Semi-Supervised Learning Algorithms," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

[7] X. Liu, J. Van De Weijer, and A. D. Bagdanov, "Exploiting Unlabeled Data in CNNs by Self-supervised Learning to Rank," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

[8] M. Wang and W. Deng, "Deep visual domain adaptation: A survey," *Neurocomputing*, vol. 312, pp. 135–153, 2018.

[9] E. Arazo, D. Ortego, P. Albert, N. O'Connor, and K. McGuinness, "Unsupervised Label Noise Modeling and Loss Correction," in *International Conference on Machine Learning (ICML)*, 2019.

[10] S. Gidaris, P. Singh, and N. Komodakis, "Unsupervised Representation Learning by Predicting Image Rotations," in *International Conference on Learning Representations (ICLR)*, 2018.

[11] Y. Li, L. Liu, and R. Tan, "Decoupled Certainty-Driven Consistency Loss for Semi-supervised Learning," *arXiv: 1901.05657*, 2019.

[12] Z. Zhang, F. Ringeval, B. Dong, E. Coutinho, E. Marchi, and B. Schüller, "Enhanced semi-supervised learning for multimodal emotion recognition," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016.

[13] M. González, C. Bergmeir, I. Triguero, Y. Rodríguez, and J. Benítez, "Self-labeling techniques for semi-supervised time series classification: an empirical study," *Knowledge and Information Systems*, vol. 55, no. 2, pp. 493–528, 2018.

[14] T. Miyato, A. Dai, and I. Goodfellow, "Adversarial Training Methods for Semi-Supervised Text Classification," *arXiv: 1605.07725*, 2016.

[15] M. Sajjadi, M. Javanmardi, and T. Tasdizen, "Regularization With Stochastic Transformations and Perturbations for Deep Semi-Supervised Learning," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2016.

[16] D. Lee, "Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks," in *International Conference on Machine Learning Workshops (ICMLW)*, 2013.

[17] A. Iscen, G. Tolias, Y. Avrithis, and O. Chum, "Label Propagation for Deep Semi-supervised Learning," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[18] A. Tarvainen and H. Valpola, "Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[19] W. Shi, Y. Gong, C. Ding, Z. Ma, T. Xiaoyu, and N. Zheng, "Transductive Semi-Supervised Deep Learning using Min-Max Features," in *European Conference on Computer Vision (ECCV)*, 2018.

[20] T. Miyato, S. Maeda, S. Ishii, and M. Koyama, "Virtual Adversarial Training: A Regularization Method for Supervised and Semi-Supervised Learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018.

[21] S. Qiao, W. Shen, Z. Zhang, B. Wang, and A. Yuille, "Deep Co-Training for Semi-Supervised Image Recognition," in *European Conference on Computer Vision (ECCV)*, 2018.

TABLE VI

TEST ERROR IN MINI-IMAGENET (LEFT) AND CIFAR-10 WITH FEW LABELED SAMPLES (RIGHT). (*) DENOTES THAT WE HAVE RUN THE ALGORITHM. BOLD INDICATES LOWEST ERROR. WE REPORT AVERAGE AND STANDARD DEVIATION OF 3 RUNS WITH DIFFERENT LABELED/UNLABELED SPLITS.

Labeled images	4000	10000
Supervised (C)*	75.69 ± 0.24	63.24 ± 0.33
Supervised (M)*	72.03 ± 0.21	59.96 ± 0.40
Consistency regularization methods		
MT	72.51 ± 0.22	57.55 ± 1.11
MT-LP	72.78 ± 0.15	57.35 ± 1.66
Pseudo-labeling methods		
LP	70.29 ± 0.81	57.58 ± 1.47
Ours*	**56.49 ± 0.51**	**46.08 ± 0.11**

Labeled images	250	500	4000
MM (WR-28)	**11.08 ± 0.87**	**9.65 ± 0.94**	**6.24 ± 0.06**
ICT* (WR-28)	52.19 ± 1.54	42.33 ± 0.08	7.26 ± 0.04
Ours* (WR-28)	24.81 ± 5.35	14.25 ± 0.86	6.28 ± 0.3
Ours* (13-CNN)	**9.37 ± 0.12**	**8.80 ± 0.45**	5.97 ± 0.15
Ours* (PR-18)	23.86 ± 4.82	12.16 ± 1.06	**5.86 ± 0.17**

[22] Y. Luo, J. Zhu, M. Li, Y. Ren, and B. Zhang, "Smooth Neighbors on Teacher Graphs for Semi-Supervised Learning," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[23] D. Tanaka, D. Ikami, T. Yamasaki, and K. Aizawa, "Joint Optimization Framework for Learning with Noisy Labels," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[24] H. Zhang, M. Cisse, Y. Dauphin, and D. Lopez-Paz, "mixup: Beyond Empirical Risk Minimization," in *International Conference on Learning Representations (ICLR)*, 2018.

[25] S. Thulasidasan, G. Chennupati, J. Bilmes, T. Bhattacharya, and S. Michalak, "On Mixup Training: Improved Calibration and Predictive Uncertainty for Deep Neural Networks," *arXiv: 1905.11001*, 2019.

[26] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting," *Journal of Machine Learning Research*, vol. 15, pp. 1929–1958, 2014.

[27] V. Verma, A. Lamb, J. Kannala, Y. Bengio, and D. Lopez-Paz, "Interpolation Consistency Training for Semi-Supervised Learning," in *International Joint Conference on Artificial Intelligence (IJCAI)*, 2019.

[28] B. Athiwaratkun, M. Finzi, P. Izmailov, and A. Wilson, "There Are Many Consistent Explanations of Unlabeled Data: Why You Should Average," in *International Conference on Learning Representations (ICLR)*, 2019.

[29] D. Berthelot, N. Carlini, I. Goodfellow, N. Papernot, A. Oliver, and C. Raffel, "MixMatch: A Holistic Approach to Semi-Supervised Learning," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[30] S. Laine and T. Aila, "Temporal Ensembling for Semi-Supervised Learning," in *International Conference on Learning Representations (ICLR)*, 2017.

[31] Y. Chen, X. Zhu, and S. Gong, "Semi-Supervised Deep Learning with Memory," in *European Conference on Computer Vision (ECCV)*, 2018.

[32] Y. Grandvalet and Y. Bengio, "Semi-supervised Learning by Entropy Minimization," in *International Conference on Neural Information Processing Systems (NIPS)*, 2004.

[33] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," University of Toronto, Tech. Rep., 2009.

[34] Y. Netzer, T. Wang, A. Coates, A. Bissacco, . Wu, B, and A. Ng, "Reading digits in natural images with unsupervised feature learning," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2011.

[35] O. Vinyals, C. Blundell, T. Lillicrap, K. Kavukcuoglu, and D. Wierstra, "Matching Networks for One Shot Learning," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2016.

[36] J. Deng, W. Dong, R. Socher, L. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[37] S. Ravi and H. Larochelle, "Optimization as a model for few-shot learning," in *International Conference on Learning Representations (ICLR)*, 2017.

[38] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[39] Y. Asano, C. Rupprecht, and A. Vedaldi, "Surprising Effectiveness of Few-Image Unsupervised Feature Learning," in *IEEE International Conference on Computer Vision (ICCV)*, 2019.

[40] T. Salimans and D. Kingma, "Weight Normalization: A Simple Reparameterization to Accelerate Training of Deep Neural Networks," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2016.

[41] L. Xie, J. Wang, Z. Wei, M. Wang, and Q. Tian, "DisturbLabel: Regularizing CNN on the Loss Layer," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[42] A. Kolesnikov, X. Zhai, and L. Beyer, "Revisiting Self-Supervised Visual Representation Learning," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[43] I. S. G. H. A. Krizhevsky, "ImageNet Classification with Deep Convolutional Neural Networks," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2012.

[44] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Deep Image Prior," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[45] N. K. S. Zagoruyko, "Wide Residual Networks," in *British Machine Vision Conference (BMVC)*, 2016.

[46] K. He, X. Zhang, S. Ren, and J. Sun, "Identity Mappings in Deep Residual Networks," in *European Conference on Computer Vision (ECCV)*, 2016.

[47] D. Ho, E. Liang, X. Chen, I. Stoica, and P. Abbeel, "Population Based Augmentation: Efficient Learning of Augmentation Policy Schedules," in *International Conference on Machine Learning (ICML)*, 2019.

[48] S.-A. Rebuffi, S. Ehrhardt, K. Han, A. Vedaldi, and A. Zisserman, "Semi-Supervised Learning with Scarce Annotations," in *IEEE International Conference on Computer Vision (ICCV)*, 2019.