SEMI-SUPERVISED LEARNING UNDER SELF-TRAINING VIA f-DIVERGENCE

Gholamali Aminian
Alan Turing Institute
gaminian@turing.ac.uk

Amirhossien Bagheri
Sharif University of Technology
amir.bagheri@sharif.edu

Radmehr Karimian
Alan Turing Institute
radmehr.karimian@sharif.edu

Mahyar JafariNodeh
Massachusetts Institute of Technology
mahyarjn@mit.edu

Mohammad-Hossein Yassaee
Sharif University of Technology
yassaee@sharif.edu

ABSTRACT

This paper investigates a range of empirical risk functions and regularization methods suitable for self-training methods in semi-supervised learning. These approaches draw inspiration from f-divergences. In the pseudo-labeling and entropy minimization techniques as self-training methods for effective semi-supervised learning, the self-training process has some inherent mismatch between the true label and pseudo-label (noisy pseudo-labels) and our empirical risk functions are robust with respect to noisy pseudo-labels. Under some conditions, our empirical risk functions demonstrate better performance when compared to traditional self-training methods.

1 INTRODUCTION AND PROBLEM FORMULATION

Many applications of machine learning, such as in finance, natural language processing and computer vision, are rich in data but lack labeling. This poses a challenge for traditional supervised learning methods. Via semi-supervised learning (SSL), labeled and unlabeled data samples are leveraged to have better performance with respect to supervised learning scenarios. One such SSL technique is self-training algorithms. These algorithms employ confident predictions from a supervised model to assign labels to unlabeled data. The two primary approaches to self-training-based SSL are entropy minimization and pseudo-labeling.

In this work, we propose new empirical risk functions and regularizers based on divergence between the empirical distribution data samples and conditional discrete distribution over the label set. Then, we apply these empirical risk functions to self-training approaches, i.e., pseudo-labeling and entropy minimization, in SSL applications. Our empirical risk functions are more robust to noisy pseudo-labels (the pseudo-label is different from the true label) of unlabeled data samples, which are generated by self-training approaches. Related works are provided in Appendix B.

Problem Formulation: We denote the space of labels and features by $\mathcal{Y}$ and $\mathcal{X}$, respectively. The set of labeled and unlabeled data samples[1] are defined with $\mathbf{X}_n^l := \{X_i^l\}_{i=1}^n$ and $\mathbf{X}_m^u := \{X_j^u\}_{j=1}^m$, where the X_i^l and X_j^u are the labeled and unlabeled data samples drawn of distribution P_X. The set of all labeled and unlabeled data samples is defined by $\mathbf{X}^{l,u} := \mathbf{X}_n^l \cup \mathbf{X}_m^u$. The labeled dataset is denoted by $\mathbf{Z}_n^l$, which contains n samples, $\mathbf{Z}_n^l = \{(X_i^l, Y_i^l)\}_{i=1}^n$, where $X_i^l \in \mathcal{X}^l \subset \mathcal{X}$ and $Y_i^l \in \mathcal{Y}$ are labeled features and the corresponding labels, respectively. For classification problems with k classes, we consider $|\mathcal{Y}| = k$. We define the uniform distribution over $\mathcal{Y}$ with $\mathrm{Unif}(k)$. Let $\hat{P}(\mathbf{Y}|X_i)$ denote the distribution over labels given the feature X_i. Our model is able to predict the underlying conditional distributions of labels given features, i.e., $P_\theta(\mathbf{Y}|X_i) := \{P_\theta(Y = y_i|X_i)\}_{i=1}^k$, where

[1]We use features and data samples terms interchangeably.

$\theta \in \Theta$ is the parameter of our model. This means that our model can estimate the probability of each possible label for each given feature vector. For example, the output of the Softmax layer in neural networks can be considered as an estimation of the conditional distribution of labels given the feature.
f-divergence: The f-divergence Polyanskiy and Wu (2022) between two discrete distributions, $P = \{p_i\}_{i=1}^k$, and $Q = \{q_i\}_{i=1}^k$, is defined as, $D_f(P\|Q) := \sum_{i=1}^k q_i f\left(\frac{p_i}{q_i}\right)$ where $f : (0, \infty) \to \mathbb{R}$ is a convex generator function with $f(1) = 0$. Note that $D_f(P\|Q) = 0$, if $P = Q$. For ease of notation, we define the general divergence and D-entropy as $D_f(P\|Q)$. Different f-divergences are introduce in Polyanskiy and Wu (2022).

2 FDP-SSL ALGORITHM

In SSL applications, we are focused on self-training approaches, which include methods such as pseudo-labeling and entropy minimization (Appendix D).
Pseudo-labeling: In this scenario, we assign a pseudo-label to each unlabeled feature through a pseudo-labeling process. We define the pseudo-labeled dataset as $\hat{\mathbf{Z}} := \{\hat{Y}^j, X_j^u\}_{j=1}^m$, where $\hat{Y}^j$ is the pseudo-labeled assigned to unlabeled data sample. Therefore, we define $\hat{P}(\widehat{\mathbf{Y}}^u|X_j^u)$ and $\hat{P}(\mathbf{Y}^l|X_i^l)$ as the empirical true label distribution and empirical distribution[2] over unlabeled dataset inspired by pseudo-label generation process for unlabeled feature X_j^u, respectively. To apply our divergence based ERM (FD-ERM) approach in this setup, we define a convex combination of the empirical distribution over label set for all labeled and unlabeled datasets by

$$\widehat{P}(\mathbf{Y}^l, \widehat{\mathbf{Y}}^u|\mathbf{X}^{l,u}) := \left\{\left\{\frac{\beta}{n}\hat{P}(\mathbf{Y}^l|X_i^l)\right\}_{i=1}^n, \left\{\frac{(1-\beta)}{m}\hat{P}(\widehat{\mathbf{Y}}^u|X_j^u)\right\}_{j=1}^m\right\},$$

where $\beta \in [0, 1]$. Similarly, the estimated conditional distribution as a joint distribution over the set $\mathbf{Y}^l \times \mathbf{X}^{l,u}$

$$P_\theta(\mathbf{Y}|\mathbf{X}^{l,u}) := \left\{\left\{\frac{\beta}{n}P_\theta(\mathbf{Y}|X_i^l)\right\}_{i=1}^n, \left\{\frac{(1-\beta)}{m}P_\theta(\mathbf{Y}|X_j^u)\right\}_{j=1}^m\right\}.$$

Note that both $P_\theta(\mathbf{Y}|\mathbf{X}^{l,u})$ and $\widehat{P}(\mathbf{Y}^l, \widehat{\mathbf{Y}}^u|\mathbf{X}^{l,u})$ are joint probability distributions over $\mathcal{Y} \times \mathbf{X}^{l,u}$. We can define the FD-ERM for SSL application based on f-divergence,

$$\hat{R}_{D_f}(\theta, \mathbf{Z}^l, \hat{\mathbf{Z}}) = D_f\Big(\widehat{P}(\mathbf{Y}^l, \widehat{\mathbf{Y}}^u|\mathbf{X}^{l,u})\|P_\theta(\mathbf{Y}|\mathbf{X}^{l,u})\Big).$$

FDP-SSL Algorithm: We propose a f-divergence-based pseudo-labeling SSL (FDP-SSL) algorithm (Algorithm 1 in Appendix C). In this algorithm, we first generate pseudo-labels for unlabeled data samples based on a process in an iterative manner. Let us define $Q(j) := \max_{i\in[k]} P_\theta(y_i|X_j^u)$ where $q := \arg\max_{i\in[k]} P_\theta(y_i|X_j^u)$, then we have, $\hat{\mathrm{Y}}_q^j := \mathbb{1}\big[Q(j) \geq \tau_p\big]$. If $\hat{\mathrm{Y}}_q^j = 0$, then the unlabeled sample would be neglected to reduce the confirmation bias incurred by pseudo-labeling. Otherwise, we select the pseudo-label for the q-th class.

3 EXPERIMENTS

We conduct the experiments for KL-ERM, JS-ERM, P-ERM, and χ^2-ERM. More Experiments and details are provided in Appendix E.
Results: In Table 1, we conducted experiments involving the FDP-SSL algorithm. In the case of the FDP-SSL algorithm, we set $\tau_p = 0.3$. For the CIFAR-100 dataset, the JS-ERM achieves the highest accuracy at 72.43 ± 1.06, outperforming other FD-ERMs. Among the FD-ERMs, JS-ERM achieved the highest accuracy in the LETTER dataset. We consider KL-ERM (Pseudo-labeling based on cross-entropy) as baseline.

[2]The empirical pseudo-label distribution can be either empirical hard pseudo-label or empirical soft pseudo-label distributions.

Table 1: Accuracy of FDP-SSL. We consider $\tau_p = 0.3$.

FD-ERM	**LETTER**	**CIFAR-100**
KL(Baseline)	58.87 ± 2.13	67.80 ± 0.75
χ^2	56.52 ± 0.67	68.02 ± 1.06
Pow, ($p = 1.2$)	58.55 ± 1.04	67.20 ± 0.34
JS	$\mathbf{61.67 \pm 0.94}$	$\mathbf{72.43 \pm 1.06}$

4 CONCLUSION AND FUTURE WORKS

We provide novel empirical risk functions and regularizers inspired by f-divergence for self-training algorithms in semi-supervised learning scenarios. Our algorithms can be applied to both pseudo-labeling and entropy-minimization. As future works, our framework can be combined with other methods for semi-supervised learning, e.g., Fixmatch Sohn et al. (2020), MixMatch Berthelot et al. (2019), and Meta pseudo-label Pham et al. (2020).

URM STATEMENT

The authors acknowledge that at least one key author of this work meets the URM criteria of the ICLR 2024 Tiny Papers Track.

BIBLIOGRAPHY

Gholamali Aminian, Mahed Abroshan, Mohammad Mahdi Khalili, Laura Toni, and Miguel Rodrigues. An information-theoretical approach to semi-supervised learning under covariate-shift. In *International Conference on Artificial Intelligence and Statistics*, pages 7433–7449. PMLR, 2022.

Eric Arazo, Diego Ortego, Paul Albert, Noel E. O'Connor, and Kevin McGuinness. Pseudo-labeling and confirmation bias in deep semi-supervised learning, 2020.

Ben Athiwaratkun, Marc Finzi, Pavel Izmailov, and Andrew Gordon Wilson. Improving consistency-based semi-supervised learning with weight averaging. *CoRR*, abs/1806.05594, 2018. URL http://arxiv.org/abs/1806.05594.

David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin Raffel. Mixmatch: A holistic approach to semi-supervised learning, 2019.

David Berthelot, Nicholas Carlini, Ekin D. Cubuk, Alex Kurakin, Kihyuk Sohn, Han Zhang, and Colin Raffel. Remixmatch: Semi-supervised learning with distribution alignment and augmentation anchoring, 2020.

Aleksandar Botev, Guy Lever, and David Barber. Nesterov's accelerated gradient and momentum as approximations to regularised update descent, 2016.

Chih-Chung Chang and Chih-Jen Lin. Libsvm: A library for support vector machines. *ACM transactions on intelligent systems and technology (TIST)*, 2(3):1–27, 2011.

Olivier Chapelle, Jason Weston, and Bernhard Scholkopf. Cluster kernels for semi-supervised learning. *Advances in neural information processing systems*, pages 601–608, 2003.

Yves Grandvalet, Yoshua Bengio, et al. Semi-supervised learning by entropy minimization. *CAP*, 367:281–296, 2005.

Ahmet Iscen, Giorgos Tolias, Yannis Avrithis, and Ondrej Chum. Label propagation for deep semi-supervised learning, 2019.

Zhanghan Ke, Daoye Wang, Qiong Yan, Jimmy Ren, and Rynson W. H. Lau. Dual student: Breaking the limits of the teacher in semi-supervised learning, 2019.

Yiwen Kou, Zixiang Chen, Yuan Cao, and Quanquan Gu. How does semi-supervised learning with pseudo-labelers work? a case study. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=Dzmd-Cc8OI.

Alex Krizhevsky. Learning multiple layers of features from tiny images. 2009. URL https://api.semanticscholar.org/CorpusID:18268744.

Dong-Hyun Lee. Pseudo-label : The simple and efficient semi-supervised learning method for deep neural networks. *ICML 2013 Workshop : Challenges in Representation Learning (WREPL)*, 07 2013.

Dong-Hyun Lee et al. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on Challenges in Representation Learning, ICML*, 2013.

Junnan Li, Richard Socher, and Steven C. H. Hoi. Dividemix: Learning with noisy labels as semi-supervised learning, 2020.

Samet Oymak and Talha Cihad Gulcu. Statistical and algorithmic insights for semi-supervised learning with self-training, 2020.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library, 2019.

Hieu Pham, Qizhe Xie, Zihang Dai, and Quoc V. Le. Meta pseudo labels. *CoRR*, abs/2003.10580, 2020. URL https://arxiv.org/abs/2003.10580.

Yury Polyanskiy and Yihong Wu. Information theory: From coding to learning, 2022.

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.

Weiwei Shi, Yihong Gong, C. Ding, Zhiheng Ma, Xiaoyu Tao, and Nanning Zheng. Transductive semi-supervised deep learning using min-max features. In *European Conference on Computer Vision*, 2018. URL https://api.semanticscholar.org/CorpusID:52958532.

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.

Daiki Tanaka, Daiki Ikami, Toshihiko Yamasaki, and Kiyoharu Aizawa. Joint optimization framework for learning with noisy labels. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5552–5560, 2018.

Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results, 2018.

Vikas Verma, Kenji Kawaguchi, Alex Lamb, Juho Kannala, Arno Solin, Yoshua Bengio, and David Lopez-Paz. Interpolation consistency training for semi-supervised learning. *Neural Networks*, 145: 90–106, jan 2022. doi: 10.1016/j.neunet.2021.10.008. URL https://doi.org/10.1016%2Fj.neunet.2021.10.008.

Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization, 2018.

Dixian Zhu and Tianbao Yang. A unified DRO view of multi-class loss functions with top-n consistency. *CoRR*, abs/2112.14869, 2021. URL https://arxiv.org/abs/2112.14869.

A NOTATIONS

Throughout the paper, upper-case letters denote random variables (e.g., Z), lower-case letters denote the realizations of random variables (e.g., z), and calligraphic letters denote sets (e.g., $\mathcal{Z}$). All the logarithms are natural ones, and all the information measure units are nats. We denote the set of integers from 1 to N by $[N] \triangleq \{1, \ldots, N\}$.

B RELATED WORKS

We provide an overview of relevant works concerning self-training techniques in SSL, as well as other SSL methodologies.

Self-training and SSL: Entropy minimization methods incorporate an entropy function as a regularization term, aiming to penalize uncertainty in label predictions for unlabeled data Grandvalet et al. (2005). The underlying assumption behind entropy minimization algorithms can be attributed to either the manifold assumption Iscen et al. (2019), which assumes that labeled and unlabeled data samples are drawn from a common data manifold, or the cluster assumption Chapelle et al. (2003), which suggests that similar data features tend to share the same label. Pseudo-labelling, introduced in Lee et al. (2013), involves training a model using labeled data and subsequently assigning pseudo-labels to the unlabeled data based on the model's predictions. These pseudo-labels are then used to construct another model, which is trained in a supervised manner using both labeled and pseudo-labeled data. Network predictions may exhibit inaccuracies, as is commonly observed in neural networks. This issue is further exacerbated when these erroneous predictions are employed as labels for unlabeled samples, a characteristic inherent in the practice of pseudo-labeling. The phenomenon of overfitting to incorrect pseudo-labels generated by the network is widely recognized as confirmation bias Arazo et al. (2020). Under different experiments, it is shown that the pseudo-labeling is effective, Arazo et al. (2020) and Rizve et al. (2021). Kou et al. (2023) shows that semi-supervised learning with pseudo-labelling can achieve near-zero test loss under some conditions. The study by Pham et al. (2020) introduced meta pseudo labelling. This method enhanced the accuracy of pseudo-labels by incorporating feedback from the student model. Rizve et al. (2021) proposed confidential-based pseudo-label generation for training networks with unlabeled data. Arazo et al. (2020) suggests soft-labeling with the MixUp method to reduce over-fitting to model predictions and confirmation bias. Oymak and Gulcu (2020) and Lee (2013) analyzed both theoretical and algorithmic side of self-training. In this work, we propose a more general framework as a combination of self-training methods which outperforms previous self-training algorithms.

Other SSL methods: Some methods use a combination of consistency regularization and pseudo-labeling. MixMatch Berthelot et al. (2019) computes k augmentations for each unlabeled sample, and one for labeled sample in the batch, then sharpens the average output probability of the model for k augmented data and applies the Mix-Up approach Zhang et al. (2018). Continuing the idea of MixMatch, Berthelot et al. (2020) introduced ReMixMatch; this method adds distributional alignment between unlabeled and labeled data, moreover, augmentation anchoring and utilizing the output of weakly-augmented data as labels for k strongly-augmented unlabeled data. Li et al. (2020) established DivideMix proposed a new method for learning with noise based on the Gaussian Mixture Model (GMM) and MixMatch method. Sohn et al. (2020) presents FixMatch, which uses weakly-augmented input model prediction pseudo-label as a label for strongly-augmented input model prediction. This line of research differs from ours as our focus is self-training algorithms despite consistency regularization methods.

C FDP-SSL

The FDP-SSL algorithm is presented in Algorithm 1. Note that, after each pseudo-labeling iteration, we balance the pseudo-labeled data samples.

Algorithm 1: FDP-SSL Algorithm

Data: $\mathbf{Z}^l = \{(X_i^l, Y_i^l)\}_{i=1}^n$ sampled from P_{XY}, $\mathbf{X}_m^u = \{X_j^u\}_{j=1}^m$ sampled from P_X, hyper-parameters β, τ_p, $\hat{R}_{D_f}(\theta, \mathbf{Z}^l)$, and $\hat{R}_{D_f}(\theta,\ \mathbf{Z}^l \cup \hat{\mathbf{Z}})$, the P_θ model based on a divergence, Iteration index by t_g and max Iterations I

Result: A trained neural network with parameter θ and output of softmax P_θ which minimizes the FD-ERM

$t_g \leftarrow 1$

Train model (Warm-Up) P_θ with SGD based on $\hat{R}_{D_f}(\theta,\ \mathbf{Z}^l)$

while $t_g \leq I$ **do**

1. Select pseudo-labels based on all unlabeled data samples $\mathbf{X}_m^u$ based on
$$\hat{\mathrm{Y}}_q^j = \mathbb{1}\big[Q(j) \geq \tau_p\big],$$
2. $\forall j \in [m]$, if $\hat{\mathrm{Y}}_q^j > 0$, then $\hat{\mathbf{Z}} \leftarrow \{(\widehat{X}_j^u, \hat{Y}_q^j) \cup \hat{\mathbf{Z}}\}$
3. Initial your model P_θ
4. $\hat{\mathbf{Z}} \leftarrow \text{Balance}(\hat{\mathbf{Z}})$
5. Train your model P_θ with SGD based on $\hat{R}_{D_f}(\theta,\ \mathbf{Z}^l \cup \hat{\mathbf{Z}})$
6. $t_g \leftarrow t_g + 1$

end

D D-ENTROPY

We can also define the f-entropy, for discrete distribution P as,

$$H_f(P) = -D_f(P \| \mathrm{Unif}(k)), \tag{1}$$

where $f(\cdot)$ is the same generator function for f-divergence and $\mathrm{Unif}(k)$ is the uniform distribution over set with size k. For example, for $f(t) = t\log(t)$, we have KL-divergence and the entropy is equal to the summation of traditional entropy and a constant term,

$$H_{\mathrm{KL}}(P) = h_{\mathrm{KL}}(P) - \log(k), \tag{2}$$

where $h_{\mathrm{KL}}(P) = -\sum_{i=1}^k P_i \log(P_i)$. Different FD-ERMs and the corresponding entropy are introduced in Table 2.

Table 2: FD-ERM and D-Entropy for KL divergence, Power divergence, JS divergence, Le Cam, and Total variation distance. We have $P_i := P_\theta(y_i^l | X_i^l)$.

Name/ Generator $f(t)$	FD-ERM	D-Entropy
KL, $t\log(t)$	$\frac{-1}{n}\sum_{i=1}^n \log(P_i)$	$-\log k - \sum_{i=1}^k P_i \log P_i$
χ^2, $(1-t)^2$	$\frac{1}{n}\Big(\sum_{i=1}^n (P_i^{-1} - 1)\Big)$	$-\frac{1}{k}\sum_{i=1}^k (1 - kP_i)^2$
Power, $t^p - 1$	$\frac{1}{n}\Big(\sum_{i=1}^n (P_i^{-p+1} - 1)\Big)$	$1 - k^{p-1}\sum_{i=1}^k P_i^p$
Jensen-Shannon, $t\log\left(\frac{2t}{1+t}\right) + \log\left(\frac{2}{1+t}\right)$	$\frac{1}{n}\Big(\sum_{i=1}^n P_i \log(P_i) - (P_i+1)\log(P_i+1)\Big)$ $+2\log(2)$	$-\sum_{i=1}^k P_i \log\left(1 + \frac{1}{kP_i}\right)$ $+\sum_{i=1}^k \frac{1}{k}\log(1+kP_i)$ $-2\log(2)$

D.1 SOFT-LABEL AND HARD-LABEL

Our study uses two distinct label types: hard-label and soft-label. In the case of hard-label, the distribution over the label set is such that $\hat{P}(Y = y_i | X_i) = 1$, indicating a certainty that the label is y_i, while $\hat{P}(Y = y_j | X_i) = 0$ for all $y_j \in \mathcal{Y}$ not equal to y_i. Conversely, in the soft-label scenario, we have $\hat{P}(Y = y_j | X_i) \geq 0$ for all labels y_j, and $\sum_{j=1}^k \hat{P}(Y = y_j | X_i) = 1$. It is worth noting that

for labeled datasets, we employ hard-labels. However, for unlabeled datasets, we have the flexibility to adopt either hard-label or soft-label.

D.2 ENTROPY MINIMIZATION

Building upon the ideas presented in Grandvalet et al. (2005), we delve into the concept of D-entropy, denoted as H_f as defined in equations equation 1. In this approach, we compute D-entropy as a regularization term over the distribution of predicted labels, denoted as $P_\theta(\mathbf{Y}|\mathbf{X}_m^u)$, for the unlabeled dataset. It's worth noting that the minimization of D-entropy can be interpreted as the maximization of $D_f(P_\theta(\mathbf{Y}|\mathbf{X}_m^u)|\mathrm{Unif}(k))$. Essentially, this means we are actively seeking predicted labels for each unlabeled feature with the maximum dissimilarity with the uniform distribution in terms of f-divergence. However, the minimization of D–entropy can cause the system to predict the same class for each data sample.

To avoid the prediction of one class for each unlabeled feature, Tanaka et al. (2018) and Arazo et al. (2020) proposed to use a KL divergence between the mean distribution of Softmax outputs for all unlabeled data samples, i.e., $\bar{P}_\theta(\mathbf{Y}^l|\mathbf{X}_m^u) := \frac{1}{m}\sum_{j=1}^m P_\theta(\mathbf{Y}|X_j^u)$, and the uniform distribution as another regularization. In a similar approach, we propose to minimize the divergence, i.e., f-divergence, between $\bar{P}_\theta(\mathbf{Y}^l|\mathbf{X}_m^u)$ and uniform distribution. Minimizing this divergence would help the system predict uniform distribution over all classes. Note that, if we have the balance assumption for all classes, then we expect that $\bar{P}_\theta(\mathbf{Y}^l)$ would be uniform. Therefore, this regularization can also help in the case when we have an imbalanced number of data samples from classes during pseudo-labeling process. In particular, after pseudo-labeling (with soft-label or hard-label), we can expect an imbalance pseudo-labeled dataset.

Our final regularized risk minimization for entropy minimization would be,

$$\begin{aligned}\widehat{R}_{\tilde{D}}(\theta, \mathbf{Z}^l, \mathbf{X}_m^u, \lambda) &:= \hat{R}_{\tilde{D}}(\theta, \mathbf{Z}^l) + \lambda_h H_{\tilde{D}}(P_\theta(\mathbf{Y}|\mathbf{X}_m^u)) \\ &+ \lambda_u D_f(\bar{P}_\theta(\mathbf{Y}^l|\mathbf{X}_m^u)\|\mathrm{Unif}(k)),\end{aligned} \tag{3}$$

where $\tilde{D} \in \{\mathrm{KL}, \mathrm{JS}, \mathrm{P}, \chi^2\}$ and $D \in \{D_{\mathrm{KL}}, \mathrm{JS}, D_{\mathrm{P}}, \chi^2\}$.

D.3 FDEM-SSL

Motivated by the concept of entropy minimization, we introduce a novel approach, Divergence-based entropy minimization Semi-Supervised Learning (FDEM-SSL), in this paper. In developing this algorithm, we build upon the techniques presented in Rizve et al. (2021), incorporating D-entropy minimization. In each iteration of the algorithm, we adopt the previous predictions of unlabeled data samples as soft-labels for these unlabeled data samples. Our objective is to minimize the FD-ERM with respect to the true labels for labeled features and the soft-labels assigned to unlabeled data samples. As discussed before, we introduce the minimization of D-entropy and the divergence term $D_f(\bar{P}_\theta(\mathbf{Y}^l|\mathbf{X}_m^u)|\mathrm{Unif}(k))$ as regularization terms. The utilization of soft-labels for unlabeled data samples serves to reduce confirmation bias, enhancing the effectiveness of our approach.

E EXPERIMENT DETAILS AND DISCUSSION

Anonymized code is provided at https://anonymous.4open.science/r/Robust_DEM_SSLv_1/README.md.

τ_p: The selection of τ_p would help us to select the most certain predictions for unlabeled data samples. In addition, increasing the τ_p would reduce the number of unlabeled samples that can be utilized in the training process. It is worth mentioning that unlabeled data are not included in the first iteration. Therefore, the model derived in the first iteration (Warm-up), is utilized to generate a pseudo-label based on $\hat{\mathrm{Y}}_q^j := \mathbb{1}\left[Q(j) \geq \tau_p\right]$ in the next iteration. After each iteration of the pseudo-labeling process, we balance the set of pseudo-labeled dataset. For this purpose, we under-sample the pseudo-labeled dataset, based on the data samples from the minority class.

Datasets: We ran different experiments to validate our proposed algorithms, FDEM-SSL and FDP-SSL on two datasets: CIFAR-100 Krizhevsky (2009) and the Letter Chang and Lin (2011) datasets. For the SSL scenario, we have allocated $n = 104$ labeled data samples and $m = 17896$ unlabeled

data samples for the Letter dataset and $n = 400$ labeled data samples and $m = 49600$ unlabeled data samples for CIFAR-100. We utilized the CNN-13 network architecture for CIFAR-100 (Iscen et al. (2019), Shi et al. (2018), Tarvainen and Valpola (2018), Verma et al. (2022), Ke et al. (2019),Berthelot et al. (2020), Athiwaratkun et al. (2018)) and 2-layer Feedforward neural network inspired by Zhu and Yang (2021) for letter.

Hyper-parameters: We use a combination of manual and automatic hyper-parameter tuning for the learning rate values and regularization coefficients. For parameter β, we select $\beta = \frac{n}{n+m}$. For FDP-SSL, we have one hyper-parameters, i.e., τ_p. We set $\tau_p = 0.3$ in $\hat{\mathrm{Y}}_q^j := \mathbb{1}\left[Q(j) \geq \tau_p\right]$.

We used 20%/80% of CIFAR-100 and 10%/90% of Letter datasets for test/training process. In the FSL scenario, we only train our network with all 80% of labeled data. The implementation uses the PyTorch framework Paszke et al. (2019), training was optimized using SGD with nesterov momentum of 0.9 Botev et al. (2016), learning rate of 0.03, cosine annealing for 5 iterations and 512 epoch for each iteration. More implementation details are provided in Table 3. Experiments are executed on Nvidia volta V100 GPU with 32 GB VM.

Table 3: Experiment setup details for CIFAR-100 and Letter

	CIFAR-100	Letter
Optimizer	SGD	SGD
Learning rate	0.03	0.03
Network	CNN-13	FFNN
Max epochs (M)	512	512
Labeled dataset size (n)	400	104
Unlabeled dataset size (m)	49600	17896
Train/Test size	50000/10000	18000/2000
Batch size	512	512
Max Iterations (I)	5	5
λ_u	0.8	0.8
λ_h	(0.4, 0.04)	(0.4, 0.04)
τ_p	(0.3,0.7)	(0.3,0.7)
β	0.992	0.994

For FDEM-SSL, regularization weights (λ_u, λ_h) inspired by Arazo et al. (2020), we selected $\lambda_u = 0.8$ and $\lambda_h = 0.4$ for FDEM-SSL across all FD-ERMs. By setting a small τ_p, we assign more pseudo-labels to unlabeled data samples. However, this increase in pseudo-labeled data samples is expected to result in more inaccurate pseudo-labels, where we have mismatches between the pseudo-labels and the true labels of the unlabeled data samples. In addition, our experiments show that for the $\tau_p \in (0.5, 1)$ the results won't change significantly.Inspired by Aminian et al. (2022), we set β to $\frac{n}{m+n}$.

E.1 τ_p Effect

We simulate the FDP-SSL for Letter Dataset with different τ_p in Table 4. We consider JS-ERM function and the baseline (KL) for this part.

Table 4: Accuracy of FDP-SSL for different τ_p with Letter Dataset

Loss	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
KL(Baseline)	52.60 ± 1.72	53.15 ± 2.02	55.45 ± 2.45	58.87 ± 2.13	60.50 ± 1.20	61.45 ± 0.98	$61.69 \pm .74$	61.72 ± 0.35	61.95 ± 0.86	62.20 ± 1.04
JS	61.0 ± 0.32	61.1 ± 0.34	61.3 ± 0.70	61.67 ± 0.94	61.70 ± 0.57	62.60 ± 0.89	62.40 ± 0.73	62.50 ± 0.90	62.90 ± 0.68	62.45 ± 0.60

As we observe the JS-ERM, which is the best performing FDP-SSL, is robust to the label noise with small variance at different τ_p. However, FDP-SSL based on KL divergence is sensitive to variation in τ_p. Note that, by decreasing τ_p, we have more noise in pseudo-labels. However, more unlabeled samples are utilized.

E.2 FDP-SSL AND FDEM-SSL

We also compare the results of FDEM-SSL with FDP-SSL in Table 5. We can observe that FDEM-SSL has a better performance in comparison with FDP-SSL in CIFAR-100.

Table 5: Accuracy of FDP-SSL and FDEM-SSL. We consider $\tau_p = 0.3$. For FDEM-SSL, we assume $\lambda_u = 0.8$ and $\lambda_h = 0.4$.

FD-ERM	LETTER		CIFAR-100	
	FDP-SSL	FDEM-SSL	FDP-SSL	FDEM-SSL
KL	58.87 ± 2.13	$\mathbf{59.14 \pm 0.65}$	67.80 ± 0.75	70.49 ± 0.51
χ^2	56.52 ± 0.67	57.60 ± 0.93	68.02 ± 1.06	69.05 ± 0.48
Pow, ($p = 1.2$)	58.55 ± 1.04	59.10 ± 0.93	67.20 ± 0.34	71.14 ± 0.46
JS	$\mathbf{61.67 \pm 0.94}$	57.49 ± 1.29	$\mathbf{72.43 \pm 1.06}$	$\mathbf{73.34 \pm 0.50}$

E.3 BALANCE EFFECT

As mentioned in FDP-SSL and FDEM-SSL, after each pseudo-labeling iteration, we balance the pseudo-labeled data samples. In Table 6, we conducted FDP-SSL and FDEM-SSL algorithms without balancing (imbalance), in order to show how FDP-SSL and FDEM-SSL can handle imbalance pseudo-labels in the training stage. Note that in this setup, we set $\tau_p = 0.3$. We can observe that under the imbalance scenario in pseudo-labeled data samples, the χ^2-ERM has a better performance in comparison with other FD-ERMs. For example, the accuracy of χ^2-ERM under balancing and imbalance for $\tau_p = 0.3$ and FDEM-SSL in CIFAR-100 is 54.17 ± 0.50 and 50.0 ± 0.48, respectively.

Table 6: Accuracy of FDP-SSL and FDEM-SSL under no Balancing. We consider $\tau_p = 0.3$ for FDP-SSL. For FDEM-SSL, we assume $\lambda_u = 0.8$ and $\lambda_h = 0.04$.

FD-ERM	LETTER		CIFAR-100	
	FDP-SSL /NB	FDEM-SSL /NB	FDP-SSL /NB	FDEM-SSL /NB
KL	45.55 ± 0.75	52.1 ± 2.48	19.46 ± 0.24	35.84 ± 0.94
χ^2	$\mathbf{53.9 \pm 1.25}$	$\mathbf{54.17 \pm 0.50}$	$\mathbf{43.14 \pm 0.47}$	$\mathbf{50.0 \pm 0.48}$
Pow, ($p = 1.2$)	43.74 ± 0.56	53.7 ± 1.09	31.45 ± 0.11	45.36 ± 1.18
JS	39.05 ± 1.15	41.13 ± 1.01	7.46 ± 0.12	46.45 ± 2.16

E.4 ACCURACY OF PSEUDO-LABELING AND NUMBER OF SELECTED SAMPLES

As we discussed, it is worthwhile to mention that the pseudo-labeling procedure can incur label noise during the training phase. To facilitate a more meaningful comparison, we present the accuracy of the pseudo-labeling process during the final iteration for both FDP-SSL and FDEM-SSL algorithms. In the case of FDEM-SSL, we determine pseudo-labels based on the highest softmax output (soft-labels). As presented in Table 7, when considering a value of $\tau_p = 0.7$, the quantity of pseudo-labeled samples is notably lower compared to the number of pseudo-labeled samples observed for $\tau_p = 0.3$, as depicted in Table 8. The accuracy of the pseudo-labeling process for $\tau_p = 0.7$ is higher than the accuracy of pseudo-labeling for $\tau_p = 0.3$. Therefore, using higher τ_p can help us to reduce the noise of the pseudo-labeling process. For the balance effect, we also provided the accuracy of pseudo-labeling process in Table 9.

Table 7: Comparison of accuracy of the pseudo-labeling process for the last iteration (number of pseudo-labeled samples) in FDP-SSL for CIFAR-100 ($n = 400$, $m = 49600$) and LETTER ($n = 104$, $m = 17896$) datasets with assuming $\tau_p = 0.7$.

D-ERM	**LETTER**	**CIFAR-100**
KL	81.41 ± 0.72 (595 ± 6)	96.76 ± 0.59 (39800 ± 87)
χ^2	62.23 ± 2.07 (179 ± 14)	97.04 ± 0.13 (10700 ± 132)
Pow, ($p = 1.2$)	80.36 ± 0.58 (523 ± 15)	97.53 ± 1.28 (39100 ± 124)
JS	70.51 ± 0.49 (4836 ± 121)	87.00 ± 0.28 (2350 ± 32)

Table 8: Comparison of accuracy of the psuedo-labeling process for the last iteration (number of pseudo-labeled samples) in FDP-SSL and FDEM-SSL. For FDEM-SSL, we consider the pseudo-label with the maximum output of softmax. We consider $\tau_p = 0.3$. For FDEM-SSL, we assume $\lambda_u = 0.8$ and $\lambda_h = 0.4$.

D-ERM	**LETTER**		**CIFAR-100**	
	FDP-SSL	**FDEM-SSL**	**FDP-SSL**	**FDEM-SSL**
KL	58.76 ± 1.87 (9117 ± 546)	5677 ± 2.02 (8684 ± 470)	86.18 ± 1.25 (34200 ± 470)	87.72 ± 1.40 (38066 ± 208)
χ^2	68.84 ± 0.46 (625 ± 4)	68.05 ± 0.84 (627 ± 12)	81.79 ± 1.5 (34850 ± 500)	85.49 ± 1.09 (28766 ± 404)
Pow, ($p = 1.2$)	63.43 ± 0.50 (1872 ± 150)	62.21 ± 0.47 (1092 ± 177)	85.53 ± 1.4 (33666 ± 660)	86.40 ± 1.53 (35600 ± 400)
JS	57.66 ± 1.38 (11136 ± 380)	51.26 ± 0.86 (13693 ± 439)	86.15 ± 0.37 (2391 ± 11)	85.02 ± 0.93 (38600 ± 529)

Table 9: Comparison of accuracy of the psuedo-labeling process for the last iteration (number of pseudo-labeled samples) in FDP-SSL and FDEM-SSL under no Balancing. For FDEM-SSL, we consider the pseudo-label with the maximum output of softmax. We consider $\tau_p = 0.3$ for FDP-SSL. For FDEM-SSL, we assume $\lambda_u = 0.8$ and $\lambda_h = 0.04$.

D-ERM	**LETTER**		**CIFAR-100**	
	FDP-SSL /NB	**FDEM-SSL /NB**	**FDP-SSL /NB**	**FDEM-SSL /NB**
KL	43.18 ± 1.18 (16540 ± 250)	56.27 ± 2.89 (12351 ± 394)	20.16 ± 0.80 (49472 ± 130)	40.48 ± 0.81 (47688 ± 817)
χ^2	61.58 ± 1.21 (8462 ± 450)	67.38 ± 1.01 (6543 ± 484)	51.95 ± 0.23 (48986 ± 264)	62.51 ± 1.17 (46315 ± 1012)
Pow, ($p = 1.2$)	44.62 ± 1.94 (14729 ± 511)	60.44 ± 1.20 (9763 ± 502)	37.35 ± 0.36 (49259 ± 416)	53.61 ± 1.31 (47292 ± 1353)
JS	34.04 ± 1.41 (17881 ± 100)	40.53 ± 1.33 (16912 ± 314)	51.61 ± 1.82 (48531 ± 612)	48.62 ± 5.37 (48802 ± 563)