Taming Overconfident Prediction on Unlabeled Data From Hindsight

Jing Li, Yuangang Pan, and Ivor W. Tsang, Fellow, IEEE

Abstract—Minimizing prediction uncertainty on unlabeled data is a key factor to achieve good performance in semi-supervised learning (SSL). The prediction uncertainty is typically expressed as the entropy computed by the transformed probabilities in output space. Most existing works distill low-entropy prediction by either accepting the determining class (with the largest probability) as the true label or suppressing subtle predictions (with the smaller probabilities). Unarguably, these distillation strategies are usually heuristic and less informative for model training. From this discernment, this article proposes a dual mechanism, named adaptive sharpening (ADS), which first applies a soft-threshold to adaptively mask out deterministic and negligible predictions, and then seamlessly sharpens the informed predictions, distilling certain predictions with the informed ones only. More importantly, we theoretically analyze the traits of ADS by comparing it with various distillation strategies. Numerous experiments verify that ADS significantly improves state-of-the-art SSL methods by making it a plug-in. Our proposed ADS forges a cornerstone for future distillation-based SSL research.

Index Terms—Classification, distillation, probability transformation, semi-supervised learning (SSL).

I. INTRODUCTION

LEARNING with partially labeled data implicitly requires the model to exploit the missing labels on its own. This form of learning paradigm, known as semi-supervised learning (SSL), is of practical significance as labeling costs are now passed on to the subsequent algorithm design.

In the context of the classification task, we attribute the success of most SSL algorithms to two indispensable components designed for unlabeled data. 1) Consistency regularization. It assumes that each unlabeled sample \( u \) should have a consistent prediction with its transformed counterpart \( T(u) \). Such a self-supervised modeling on unlabeled data can be viewed as extending the conventional similarity regularization over neighboring samples [2], [3], [4] to meaningful transformations, providing a favorable optimization direction for classification. Recent work [5], [6], [7], [8] shows that consistency regularization with suitable data augmentations has greatly boosted SSL performance in terms of different types of data. A more comprehending understanding of consistency states that it tries to find a smooth manifold with the help of unlabelled (and augmented) data, and many representative SSL methods follow this insight, including Π Model [9], Temporal Ensembling [10], Mean Teacher [11], UDA [12], and so forth. 2) Prediction distillation. The predictions on unlabeled data are often of high uncertainty due to the lack of supervised information, similar to that of unsupervised clustering analysis [13], [14]. Prediction distillation solves this problem by enhancing the prediction certainty based on the current output, and minimum entropy (ME) [15] is one of the most popular strategies. With such a distillation for classifying unlabelled samples, the decision boundary will be encouraged to depart from the region where ambiguous predictions reside (low certainty area). Many SSL models [15], [16], [17], [18] in literature have benefited from this component. It is worth noting that learning from model outputs can be efficiently implemented, while some other studies pursue the same goal with more sophisticated models (See Section II).

Generally, given a labeled set \( \mathcal{L} \) and an unlabeled set \( \mathcal{U} \), an SSL model parameterized by \( \theta \) typically minimizes the following objective:

\[
J(\mathcal{L}, \mathcal{U}; \theta) = J_S + \alpha J_C + \beta J_D
\]

(1)

where \( J_S \) denotes a supervised loss on the labeled set \( \mathcal{L} \), \( J_C \) and \( J_D \) denote the consistency loss and distillation loss on unlabelled data \( \mathcal{U} \), respectively. \( \alpha \) and \( \beta \) are nonnegative weight factors for balancing the three losses. We note that (1) is simple but powerful and most SSL models can be described as a specific form of it, as stated in the parallel work [19], a very recent survey on deep SSL. Despite the great boost of consistency regularization to SSL, extra expertise is usually required to customize the effective data augmentation with respect to different data types [20]. Conversely, prediction distillation manages to learn from the model output, serving as a more general and practical tool in the machine learning community. Besides the aforementioned ME, many other distillation strategies and their variants have been developed lately, such as Sharpening (SH) [21], [22], [23], pseudo-labeling (PL) [6], [24], [25], negative sampling (NS) [26]. From the optimization perspective, although many of them progressively...
encourage partial unlabeled data to reach the low-entropy state, existing strategies unavoidably introduce incorrect distillations because neural networks may not be well calibrated during the training process [27]. The corresponding predictions are then dubbed overconfident predictions\(^1\) in this study. That means for unlabeled predictions, not all distillations are beneficial.

Many studies on SSL can be viewed as efforts to circumvent the underlying challenge in the target problem. By discriminating the importance of unlabeled data, [28] learns the sample-wise weights for unlabeled data through an additive labeled validation set. Instead, one can also leverage the self-paced technique [29], [30] to select reliable unlabeled samples. Lately, [31] proposes to reduce the bias of PL by approximating the calibration criterion from the overall uncertainty. These methods practically require a lot of extra effort, albeit some improvements have been made. Specifically, the labeled validation set needed in [28] is not always allowed in real applications. Self-paced strategy [29] may slow down the convergence rate, whose reported performance cannot rival the state-of-the-arts yet in their article. The approximated calibration executed in [31] is implemented by Monte Carlo dropout [32] which demands multiple inferences compared with standard SSL training. Apart from these studies, to obtain reliable labels for unlabelled data, the recent work SLA [33] interprets the label assignment process as an optimal transportation (OT) problem between examples and classes which is helpful, especially given the proper class distribution and allowable large training batch. Differently, our research scope restricts remedying overconfident predictions by tailor-designing a more intelligent prediction distillation strategy which is expected to be less computational and more extendable.

Looking back on the execution of an SSL model equipped with a distillation component, we realize the overconfident prediction problem could be mitigated if we stop greedily encouraging every probability distribution to be one-hot and focus on informative classes instead. With this hindsight, we present adaptive sharpening (ADS), a novel and simple prediction distillation method. By replacing the activation function with sparsemax transformation, ADS performs adaptive selection on the logits layer, which prepares for the consequent loss computation. As a result, ADS concentrates on a set of promising classes automatically selected for each unlabeled sample and avoids the unnecessary back-propagation for nontarget classes as well, which is beyond sample selections [29], [30]. Additionally, ADS constructs a mild target label distribution by exponentially raising the sparse probabilities, leading to an informative optimization for minimizing prediction uncertainty.

We borrow the name of SH used in [22] and [34] to our method, but ADS is essentially distinct from SH according to the comparison in the left panel of Fig. 1. We theoretically analyze the working mechanism of ADS and demonstrate its superiority through numerous experiments. The major contributions of this article are summarized as follows.

1) We propose a new distillation strategy, named ADS, to distill low-entropy predictions for unlabeled data.

2) We analyze the connection and difference between ADS and other distillation strategies through extensive comparisons. In particular, ADS adaptively masks out the overconfident and negligible predictions and promotes informed predictions only which is more instructive.

3) We verify the superiority of our ADS by combining it with advanced SSL models, including virtual adversarial training (VAT) [17], MixMatch [22], and FixMatch [6]. The experimental results on different benchmarks demonstrate that ADS achieves significant improvements over existing technologies.

This article is organized as follows. In Section II, we present a series of recent SSL works that are closely related to our study. Section III revisits various distillation strategies which are analyzed from an entropic view. The proposed ADS and its corresponding SSL training model are introduced in Section IV, followed by the theoretical justification for its properties in Section V. Experiments in Section VI demonstrate the efficacy of ADS. Section VII concludes the whole work and the last part is for appendixes.

II. RELATED WORK

The most related SSL works can be expressed from the following two perspectives.

A. Consistency Regularization

The concept of consistency can date back to the graph-based SSL work [2] where similar training pairs are expected to have the same label assignment. Recent SSL research has extended consistency to a more general case where natural unlabeled data and their transformed counterparts should correspond to the same output. This is closely related to the thought of self-supervised learning and many SSL studies [9], [21], [23], [35] are furthering this direction. The adopted transformations play an important role therein, and they may include multiple passes of a single input with adversarial perturbations [36], randomized data augmentations [37], [38], etc. Beyond the above consistency, using the prediction of weak augmentation to inducing the prediction of corresponding strong augmentation [6] lately shows superior performance. Summarily, consistency regularization is helpful because it facilitates the model to be insensitive to the possible disturbances, which is exactly desired for classification tasks.

B. Prediction Distillation

Existing research approaching certain predictions over unlabeled data typically adopts a post-processing distillation [39]. The pioneer strategy, such as ME [15] and its variant [9], plainly punishes the high entropy predictions, enabling the model to have a decision boundary that traverses low-density regions. Thus, SSL models equipped with ME will exactly have one-hot predictions for all unlabelled data in theory. Differently, SH [34] constructs a mild target and aligns with it in each step, showing a good performance in MixMatch [22]. PL [24] is well-known and widely applied in

---

\(^1\)This problem is called over-confirmation for PL in [25].
recent research [6], [25], [29], [30]. Opposite to the above approaches, NS [26] lately proposes to punish the prediction of the classes to which the sample less likely belongs. It is viewed as a safer option since searching for the negative classes based on the model output introduces less risk than directly picking up the true class. Other studies such as UPS [31] and NP-Match [40] leverage the model uncertainty to correct the confidence of pseudo labels in a manner of a sample-level selection method with the sacrifice on computation efficiency.

For those reinterpretating SSL from other perspectives, such as generation-based [41], adaptation-based [42], and OT-based [33], we recommend readers to refer to two recent surveys [19], [43] for broader explorations.

**Notation:** All the vectors are written in boldface. For a vector \( p \), the \( k \)th entry is denoted as \( p_k \), and \( p_{(k)} \) is the \( k \)th sorted coordinate of \( p \) in a descending order. The classification network with model parameters \( \theta \) is denoted by \( f(\cdot; \theta) \), whose emitted logits vector is represented by \( z \).

### III. Prediction Distillation Strategy Revisit

In this section, we first present the formulation of existing distillation strategies and then analyze how they work from an entropic view.

**A. Formulation**

In a \( K \)-way classification problem, the softmax function produces strictly positive predictions (probabilities) for a logits vector \( z \in \mathbb{R}^K \) by componentwise computing

\[
\text{softmax}_i(z) = \frac{\exp(z_i)}{\sum_k \exp(z_k)}. \tag{2}
\]

For a compact statement, we further denote \( p = \text{softmax}(z) \). Now we formulate different distillation strategies as follows.

1) **ME:** The loss for ME of an unlabelled sample is written as

\[
\mathcal{J}^\text{ME}_D(p) = -\sum_{i=1}^{K} p_i \log p_i. \tag{3}
\]

Notice that ME is originally defined on unlabeled data \( \mathcal{U} \) and it also applies to augmented unlabeled data, i.e., \( T(\mathcal{U}) \).

2) **SH:** SH function [22], [44] constructs the target label by adjusting the temperature of the current categorical distribution \( p \)

\[
\text{SH}_i(p, \lambda) = \frac{p_i^{1/\lambda}}{\sum_k p_k^{1/\lambda}} \tag{4}
\]

where \( \lambda (\lambda > 0) \) is the temperature that controls how sharp the output distribution looks. Then the corresponding distillation loss is formulated as

\[
\mathcal{J}^\text{SH}_D(p) = \text{Dist}(\text{SH}(p, \lambda), p) \tag{5}
\]

where \( \text{Dist}(\cdot) \) represents any feasible distance measure over distributions, such as Kullback–Leibler (KL) divergence or Wasserstein distance. We use \( \text{Dist}(\cdot) \) as a general distance in the rest of the article when there are multiple alternatives.

3) **PL:** Naive Pseudo-Labeling [24] could be described as a hard version of sharpening, and it picks the class which has the maximum predicted probability to be the target label

\[
\text{PL}_i(p) = \begin{cases} 1, & \text{if } i = \arg \max_j p_j \\ 0, & \text{otherwise.} \end{cases} \tag{6}
\]

In practice, a stricter condition applies [6] that requires the maximum probability to exceed a predefined threshold \( \tau_{PL} \). This means that PL does sample selection in each round of optimization. For each selected sample, the distillation loss is written in a similar manner to SH

\[
\mathcal{J}^\text{PL}_D(p) = \text{Dist}(\text{PL}(p), p). \tag{7}
\]

4) **NS:** Picking negative classes by a threshold \( \tau_{NS} \) and minimizing their probabilities comes to the distillation loss of

\[
\sum_{i=1}^{K} (\mathbb{I}(i < \tau_{NS}) p_i) \tag{8}
\]

where \( \mathbb{I}(\cdot) \) is the indicator function. This loss term is proved to better approximate the true likelihood of unlabeled data in [26] and we interpret it as one of the distillation baselines.

**B. Entropic View of Distillation**

We observe that existing distillation strategies are in pursuit of low-entropy predictions via post-processing the model predictions in different ways. According to their formulations, we summarize them into the left of Fig. 1 where they are compared in terms of candidate classes selection, the sparsity of activation, and the way of uncertainty minimization. Before diving into more details, we need informally introduce the following definitions.

The labeled data can be viewed as anchor points that initialize the model and induce the unlabeled prediction. Given \( 0 < \theta_1 \ll 0.5 \ll \theta_2 < 1 \), we adhere to the philosophy of this thought and partition the intact probability space into four intervals

1) Negligible: \( \{p_i \mid p_i < \theta_1, i = 1, 2, \ldots, K\} \);
2) Ambiguous: \( \{p_i \mid p_i = (1/K), i = 1, 2, \ldots, K\} \);
3) Informed: \( \{p_i \mid \theta_1 < p_i \leq \theta_2, p_i \neq (1/K), i = 1, 2, \ldots, K\} \);
4) Determinate: \( \{p_i \mid p_i > \theta_2, i = 1, 2, \ldots, K\} \).

The above definitions allow us to roughly categorize each prediction as one of the four types. Thus, we can see that ME and SH consider all classes as the candidates and aggressively minimize uncertainties over all categories except ambiguous ones which provide no useful information for distillation. By simply setting \( \theta_1 = \theta_{NS} \) and \( \theta_2 = \theta_{PL} \), we conclude that PL and NS resort to heuristic thresholds to ensure that distillation acts on relatively certain (determinate or negligible) predictions only. As a result, ME, SH, and PL suffer the problem of overconfident predictions [25], [28] as they are enhancing the determinate predictions (type I in Fig. 1). Notice that NS tries to penalize the negligible predictions, which seems to be safer. However, we demonstrate that it actually...
has a similar effect to PL for binary class classification as further discussed in Section V-A.

For the sake of a better understanding of the different distillation strategies, we display four types of prediction in a binary-class classification case on the right of Fig. 1. As we have interpreted all distillation strategies as uncertainty minimization, we indeed decompose each distillation strategy in this probability space. It is observed ambiguous prediction is special and it only takes a point. \( \theta_1 \) and \( \theta_2 \) do not coexist in existing methods, and they are not always empirically satisfying \( \theta_1 + \theta_2 = 1 \). In this work, we aim to leverage the informed predictions only, which remedies the overconfident predictions by doing nothing to the extremely certain predictions (More benefits can be seen in Section V-B). Particularly, as in different training stages and for different unlabeled samples, the informed predictions should be relatively different and dynamic. Thus, the informed predictions are not quite aligned with the ones defined by hard thresholds used in PL and NS.

IV. ADS BASED SSL FRAMEWORK

According to the understanding of prediction distillation indicated by the left of Fig. 1, we propose ADS, a dual distillation mechanism that only sharpens the informed predictions to avert the overconfident issue. An architecture of ADS from logits to losses is showcased in Fig. 2. We also analyze the drawbacks of them is the adoption of a predefined threshold, which is not robust enough to account for individual variability and model variability at different training stages, dwindling its effectiveness in practical implementations.

Therefore, we need a flexible sampler, which can automatically determine a partial set of promising classes as candidates for the true label, being adaptive to each unlabelled sample at different training stages. In this study, we study this problem from neural networks’ emission, i.e., logits vector \( z \). Instead of applying a truncation-based approach, we apply sparsemax [45] to logits, which is written as

\[
\text{sparsemax}(z) = [z_i - \tau(z)]_+ \tag{9}
\]

where \([u]_+ := \max(0, u)\).

Proposition 1: The form of \( \tau(z) \) in (9) is expressed as follows. Let \( z_{(1)} \geq z_{(2)} \geq \cdots \geq z_{(K)} \) be the sorted coordinates of \( z \), and define \( \kappa(z) := \max\{k \in \{1, \ldots, K\} | 1 + k z_{(k)} > \sum_{j \neq k} z_{(j)}\} \). Then, \( \tau(z) = (((\sum_{j \leq k} z_{(j)}) - 1)/\kappa(z)) \).

Note that \( \kappa(z) \) and \( \tau(z) \) in Proposition 1 is not manually designed because of the fact that sparsemax is the closed-form solution of the Euclidean projection problem [46]. One can find more details about the optimization in [45].

In this study we are interested in the following properties of the sparsemax: 1) \( \kappa(z) \) implies the support of sparsemax(\( z \)). Given a logits vector \( z \), the form of \( \kappa(z) \) determines the smallest \( K - \tau(z) \) coordinates will produce zero probabilities. 2) \( \sum_i [z_i - \tau(z)]_+ = 1 \) always holds where \( \tau(z) \) is dependent on \( z \). Compared with the hard thresholds on predictions, \( \tau(z) \) can be viewed as an adaptive soft-threshold on logits. We are in favor of these two properties as sparsemax is expected to gradually rule out the unrelated classes in SSL, which serves as an elegant sampler for distillation.

Selecting proper candidate classes motivates the employment of sparsemax. This transformation also implies the informed predictions needed by the distillation component.

| Distillation Strategies | Candidates selection | Activation in target label | Uncertainty minimization |
|-------------------------|----------------------|---------------------------|-------------------------|
| Minimum Entropy (ME)    | None                 | Single                    | I, II, III              |
| Sharpening (SH)         | None                 | Pull                      | I, II, III              |
| Pseudo-Labeling (PL)    | Hard threshold       | Single                    | I                       |
| Negative Sampling (NS)  | Hard threshold       | Partial                   | II                      |
| ADaptive Sharpening (ADS)| Soft threshold       | Partial                   |                         |

Fig. 1. Comparison among various distillation strategies, each of which is viewed as a two-stage process by first selecting candidate classes and then aligning their predictions with the categorical target label distribution. (I) Enhance determinate predictions, (II) promote informed predictions, (III) suppress negligible predictions (left). Different strategies turn out to have different fashions to minimize prediction uncertainty. We showcase the binary classification for simplicity where \( p = (s, 1 - s) \) where \( 0 \leq s \leq 1 \) (right).
We focus on the completeness of our method here and leave the analyses of how ADS fulfills our hindsight in Section V-A.

2) Sharpening on Sparse Probabilities: Prediction distillation acts as a self-training process. To iteratively refine the classification model, we define the distillation loss of ADS as a distance, e.g., KL-divergence, between the raw prediction \( p \) and the constructed target label distribution \( q \)

\[
J_D = \sum_{u \in \mathcal{U}} \text{KL}(q||p) = \sum_{u \in \mathcal{U}} \sum_i q_i \log \frac{q_i}{p_i} \tag{10}
\]

where \( p_i = \text{sparsemax}(f(u; \theta)) \), where \( \text{sg} \) stands for the stop-gradient operator that is defined during back-propagation. This operation enables that minimizing \( J_D \) induces \( p \) to approach \( q \). Similar to (4), the second step is calculating each element of \( q \), which is a normalized probability raised from \( \hat{p}_i \) with the power of \( r \)

\[
q_i = \frac{\hat{p}_i^r}{\sum_j \hat{p}_j^r} \tag{11}
\]

Obviously, if \( \hat{p}_i = 0 \), we have \( q_i = 0 \), and if \( \hat{p} = (1/K)1 \), we have \( q = \hat{p} \). Otherwise, as \( r \to \infty \), the categorical distribution \( q \) will approach a delta distribution.

We emphasize that although ADS borrows the idea of SH for constructing the target label distribution, they have different functionalities during distillation. Particularly, we will show ADS does not aggressively distill relatively certain predictions in Section V-A, which mitigates the overconfident risk.

Fig. 2 shows the process of our distillation design starting from derived logits. Note that (10) is not restrictive to the original unlabeled data \( \mathcal{U} \); it is applicable to augmented unlabeled data \( T(\mathcal{U}) \) as well if data augmentation techniques are employed.

B. In Conjunction With Other Losses

As the supervised loss and the consistency regularization are both constructed over predictions, unlike other post-processing distillation strategies, ADS indeed has a direct impact on these two losses as well.

In terms of the consistency regularization, it restricts each unlabeled sample to reach a consistent prediction with its transformed counterpart. Thus for ADS, the consistency loss \( J_C \) could be generally formulated as

\[
J_C = \text{Dist} (\text{sparsemax}(z), \text{sparsemax}(z')) \tag{12}
\]

where \( z \) and \( z' \) are logits vectors of the inspected unlabeled sample and its counterpart, and \( \text{Dist}() \) could be any proper distance function. Suppose \( \text{sparsemax}_i(z) = \text{sparsemax}_i(z') = 0 \) holds for the \( i \)th class. In this case, class \( i \) will not contribute to the consistency loss. With more classes like this, the consistency loss of ADS will focus on a few of confusing classes only. From (2), the condition of equality of \( \text{softmax}_i(z) = \text{softmax}_i(z') \) is much stricter. That means softmax-based consistency will be distracted by the unrelated classes during model training. Taking image classification as an example, an unlabeled image is categorized to \textit{leopard} only if all its transformed counterparts reach a consensus on \textit{leopard}. In practice, the improvement of consistency regularization essentially lies in its constantly confirmed decision among the confusing classes. Intuitively, an augmented \textit{leopard} image might be somehow close to a \textit{cat} while it should less similar to a \textit{dog}. Thus, putting more attention on confusing classes is expected to achieve better performance which is also referred to in previous research [32].

Particularly, we also allow the networks’ output of labeled data to pass the sparsemax. However, if the model assigns zero probability to the gold label, the training sample would be ruled out [45]. Although there exists some possible workaround like adding a small constant to the probabilities and then doing re-normalization, we instead use the following supervised loss as an alternative of cross entropy proposed by the recent work [47]:

\[
J_S = \frac{1}{2} (||y - z||^2 - ||\text{sparsemax}(z) - z||^2) \tag{13}
\]

where \( y \) is the ground truth encoded in a one-hot format. Notice that the \( \ell_2 \)-norm-based loss trickily averts the aforementioned optimization dilemma, and the derived gradient with respect to \( z \) has a closed form, making a feasible backward propagation.

V. THEORETICAL ANALYSES

In this section, we theoretically justify the principle of ADS, analyze its superiority from the view of the transformation function, and present its computation cost.

A. ADS Promotes Informed Predictions

We highlight that ADS does distillation via promoting informed predictions only, showing an efficient distillation mechanism compared with existing strategies.

Regarding a two-class case, we denote the neural networks’ output as \( z = (u, 0) \). In terms of same \( z \), suppose \( \text{softmax}_1(z) = s \) and \( \text{sparsemax}_1(z) = s' \). In particular, we have \( s \neq s' \) if \( u \neq 0 \). Since existing distillation strategies work over softmax output, we consider \( s' \) as a function of \( s \) and rewrite the distillation loss \( J_D \) (refer to the detailed derivation in Appendix A), making a clear and direct comparison for all methods in the same probability space. Fig. 3 plots the target probability, the prediction distillation loss, and the corresponding gradient versus the first-dimensional probability \( s \) in terms of each distillation strategy. We remind readers to read this figure from the shape and trend of every curve other than the scale as they can be equipped with different weights during formulations. Our main conclusions are as follows.

1) ADS is the unique strategy focusing on informed predictions only. According to Fig. 3(b), we observe that the proposed ADS has zero penalties when \( s \) is negligible
or determinate. Notably, ADS is viewed as a corrective SH by masking out the determinate predictions to avoid introducing overconfident risk.

2) PL and NS are consistent strategies for binary classification case. PL and NS have the complementary philosophy from Fig. 3(a). Given proper thresholds, they are shown to have the same form of loss and gradient according to Fig. 3(b) and (c).

3) ME has the unbounded gradients. ME takes the highest loss value but produces zero gradient on ambiguous predictions according to Fig. 3(b) and (c). Meanwhile, when \( s \) is relatively small or large, the gradient of ME is large, exposed to the risk of unstable optimization.

From the above analyses, ADS will outperform other distillation strategies if the assumption holds that leveraging informed predictions only is sufficient for distillation. From Fig. 3 we realize the terminology “informed” is factually specified by a corresponding threshold in the probability space of softmax output. Now we are exploring a more general case where data could be categorized into one of the multiple classes.

**Theorem 1:** For an unlabeled sample \( u \) whose softmax output is \( p \in \mathbb{R}^K (K > 2) \), the distillation loss \( J_D \equiv 0 \) for ADS in (10) holds if \( p_{(1)} > e p_{(2)} \), where \( p_{(1)}, p_{(2)} \) are the first two largest coordinates of \( p \) and \( e \) is Euler number.

**Proof:** Let \( z \in \mathbb{R}^K \) denote the logits of \( u \), i.e., \( z = f(u; \theta) \), and \( p = \text{softmax}(z) \). According to the definition of softmax, i.e., (2), we have \( p_i = (e^z_i / C) \) for \( i = 1, 2, \ldots, K \), where \( C = \sum e^z_i \). Then we can rewrite \( z_i = \ln(C p_i) \), for \( i = 1, 2, \ldots, K \). Let \( p_{(1)}, p_{(2)} \) are the first two largest coordinates of \( p \), and we have element expression

\[
z_{(1)} = \ln(C p_{(1)}), \quad z_{(2)} = \ln(C p_{(2)})
\]

where \( z_{(1)}, z_{(2)} \) denote the corresponding first two largest coordinates of \( z \). Apart from the trivial case of \( p_{(1)} = p_{(2)} = \cdots = p_{(K)} = (1/K) \), \( J_D \equiv 0 \) holds iff sparsemax(z) reaches one-hot. According to the closed-form solution of sparsemax in Section IV-A1, we obtain

\[
1 + 2z_{(2)} \leq z_{(1)} + z_{(2)}.
\]

Taking (14) into (15), we arrive

\[
1 + \ln(C p_{(2)}) \leq \ln(C p_{(1)}) \Rightarrow p_{(1)} \geq e p_{(2)}
\]

which completes the proof.

From Theorem 1 we can see that the “informed” predictions spoken of in the output space of softmax refer to the categorical probability \( p \) which satisfies \( p_{(1)} < e p_{(2)} \) and \( p \neq (1/K)1 \). That means, unlike existing distillation strategies, ADS does not encourage the relatively certain predictions to further become extremely certain since the distillation loss \( J_D \equiv 0 \) holds as long as \( p_{(1)} \geq e p_{(2)} \), and thus the issue of overconfident predictions is mitigated. The following corollary quantifies the soft threshold used by ADS.

**Corollary 1:** For a \( K \)-way semi-supervised classification problem, the determinate predictions and negligible predictions for ADS are masked out by the sample dependent threshold \( \theta_1 \in [(e/(e + K - 1)), ((e/(e + 1))] \) and \( \theta_2 \in [(e^p/(\rho + e^K(K - \rho))), (e^p/(\rho + e^K))] \) in the corresponding softmax output space, respectively, where \( e \) is the Euler number and \( \rho \) is the population of nonzero predictions.

We leave the proof of this corollary to Appendix B. Notably, we clarify that the determinate predictions are spoken of in the context of multiclass classification where only a single class is the groundtruth. In other words, a prediction is said determinate in terms of its potential to be the real label. In addition, the negligible predictions are handled similarly, but its population for a single unlabelled sample is at least 1. In particular, for ADS the negligible predictions do not coexist with the determinate prediction.

**B. ADS Facilitates Entropy Minimization**

Entropy minimization is a golden principle that has been demonstrated effective in the existing SSL research [15], [17], [18]. In this section, we point out this principle is fundamentally related to the probability transformation function used in neural networks, e.g., softmax activation. It typically involves projecting a logits vector \( z \) on the probability simplex with an optimized problem of

\[
p^* = \arg \min_{p \in \Delta^{K-1}} \{- (z, p) - H(p)\}
\]

where \( \Delta^{K-1} \) is the probability simplex with freedom of \( K - 1 \), and \( -H(p) \) is a convex function, serving as a regularizer.

When \( H(p) \) is implemented using Shannon entropy, i.e., \( H(p) = -\sum_i p_i \log p_i \), the closed-form solution of (17)
is the softmax transformation, a.k.a. the maximum entropy transformation. When \( H(p) \) is replaced with Gini entropy, namely \( H(p) = (1/2) \sum_i p_i (1 - p_i) \), (17) becomes the aforementioned Euclidean projection, and its solution is also known as the sparsemax transformation, i.e., Eq (9).

Back to the ME principle, existing distillation methods built on softmax can be viewed as minimizing the prediction entropy in a post-processing manner [39], which conflicts with the function of the regularizer \( H(p) \) in (17). As Shannon entropy is a stronger penalty than Gini entropy, softmax would intensify contradictions compared with sparsemax [47]. Therefore, we argue that our ADS facilitates entropy minimization since it looks for a balance between probability transformation and predication distillation.

C. Complexity of ADS

Typically, ADS is implemented via an additional loss term which does not bring extra model parameters. Compared with other distillation strategies exhibited in the left of Fig. 1, the bottleneck of computing ADS is the involved sparsemax, i.e., (9). It is because all there distillation losses are built on output probabilities and they have a consistent formulation (See (3), (5), (7), (8), (10)). During the forward propagation, computing sparsemax probabilities incurs a the complexity of \( O(K \log K) \) due to the operation of sorting logits. By using the median pivot and partitioning, the expected complexity can be reduced to \( O(K) \) [48], which is practically comparable with the componentwise computation of softmax (2). Regarding the backward propagation, the product between the Jacobian and a given vector is required. In case of sparsemax, it is computed with \( O(K \kappa(z)) \), where \( \kappa(z) \) is the number of nonzero probabilities precomputed in the forward propagation [45]. This sublinear time complexity is advantageous once many nongold classes are excluded. Furthermore, the wall-time validation about its time efficiency is empirically investigated in Section VI-F.

VI. EXPERIMENTS

A. Experimental Setup

We conduct empirical comparisons following the experimental setting of [26] except FixMatch for which we adopt the codebase from [6]. For a fair comparison, the default network architecture for all datasets except MNIST and ImageNet is Wide ResNet-28-2 [49] with 1.5M parameters. Regarding MNIST, we employ a seven-layer convolutional neural network, and for ImageNet we use a ResNet-50 network architecture. The batch size for unlabeled data is 64. In terms of the labeled data, the batch size is set as the number of labeled data if it is smaller than 64, and set as 64 otherwise. We do not exhaustively adjust the network parameters for different benchmarks, such as the number of scales or filters, so as to best reproduce the results.

In particular, we use “\( X + Y \)” to dub a method by adding a distillation strategy \( Y \) to an SSL algorithm \( X \), and use “\( X-Y \)” to dub an SSL algorithm \( X \) whose distillation component is replaced by \( Y \).

B. Study on VAT

The VAT [36] serves as a powerful SSL algorithm without requiring manual data augmentation [18]. It contains a consistency loss which has a similar form with (12) but optimizes the adversarial perturbation \( \nu_{adv} = \epsilon(||g||) \), where \( \epsilon \) is a predefined perturbation scale and \( g \) is approximated by the gradient on a randomly sampled unit vector \( r \), that is

\[
\nabla_r \text{Dist}(\text{softmax}(f(x; \theta)), \text{softmax}(f(x + \nu; \theta))). \tag{18}
\]

VAT is demonstrated to be improved if it is added to a ME according to [17]. Based on this fact, we implement various distillation strategies on VAT for a direct and fair comparison. To this end, we construct each baseline by replacing the distillation loss \( J_D \) of (1) with differential distillation strategies. Experiments are conducted on MNIST and CIFAR-10 with 20 and 4000 labeled training examples, respectively.

Table I presents the test error of various distillation strategies based on VAT. It is observed that each distillation loss benefits to SSL and helps to the better performance. More importantly, our ADS achieves the lowest test error compared with both vanilla VAT and other variants. In particular, the improvement to the second-best on the two datasets is around 4.7% and 1.5%, respectively.

To have a close look at the distillation component for MNIST dataset on which a better improvement has been achieved, for each method, we visualize the distillation loss \( J_D \) and average dominant probability. Average dominant probability is calculated over all unlabeled training samples \( \bar{p}_{(1)} = (1/|U|) \sum_{x \in U} p_{(1)} \) which reflects the overall certainty of unlabeled predictions. We record the two measures at each epoch and their results are shown as Fig. 4.

In terms of our ADS, the distillation loss decreases stably with the increment of average dominant probability. The curve of \( \bar{p}_{(1)} \) reaches a cusp and levels off, with more modest increases after \( \bar{p}_{(1)} = 0.91 \). For the other four strategies, the distillation loss decreases dramatically, and the average dominant probability surges within a small number of epochs. It is observed that their values of \( \bar{p}_{(1)} \) locate at the cusp which are all much larger than that of ADS’s. That is to say, they apt to optimize the learning model to an extremely certain state but inevitably introduce overconfident distillations. Instead, our ADS gradually optimizes the model by using informed predictions, averting the overconfident risks naturally. Let \( U_\ell \) and \( U_{\ell} \) denote the correctly and incorrectly classified unlabeled training examples, and we can rewrite average dominant probability as \( \bar{p}_{(1)} = (1/|U|)(\sum_{x \in U} p_{(1)} + \sum_{x \in \ell U} p_{(1)}) \). For the correct ones, they are expected to be more certain but are not supposed to be extremely certain in practice. A piece of evidence for this claim is that in a supervised task, penalizing the low-entropy prediction has been demonstrated
to be beneficial to model generalization. Label smoothing [50] is exactly one simple trick following this philosophy. For the incorrect ones, their dominant predictions should be small as they provide a wrong optimization direction. To sum up, the relatively smaller $\bar{p}(1)$ of ADS than competitors suggests better performance by considering two aspects. This is another hindsight by reviewing existing methods.

### C. Improvement on Advanced SSL Algorithms

To evaluate the efficacy of our ADS on the SSL algorithm which benefits from data augmentations, we plug ADS in two state-of-the-art models MixMatch [22] and FixMatch [6]. Since vanilla MixMatch has already incorporated SH and vanilla FixMatch has adopted PL as their distillation component, respectively, we do not conduct the corresponding ablation study. Notably, ME does not construct an explicit target label, and thus cannot be applied to data augmentation-based SSL algorithms, which require distilling explicit labels for augmented unlabeled data. Following [6], [35], we evaluate the remaining baselines on four benchmarks CIFAR-10, CIFAR-100, SVHN, and STL-10, where different numbers of labeled data are used.

Table II shows that ADS-based methods achieve the best performances compared with other combinations, demonstrating the superiority of ADS over other distillation strategies, i.e., SH, PL, and NS. In addition, we observe that MixMatch + NS also achieves some improvement over vanilla MixMatch. Because it is a simple compensation for oversusing extremely certain predictions [See the distillation loss of SH and NS in Fig. 3(b)]. Note that although ADS improves both base models, the improvement on FixMatch is not as large as that on MixMatch, and the latter even exists in some failure case i.e., CIFAR-10 with 250 labels. We explain this phenomenon as follows. ADS improves MixMatch essentially by reducing the influence of determinate predictions which are ignored by SH used in MixMatch. Such improvement is attributed to the property of class-wise selection (Section IV-A1) of our ADS. Regarding FixMatch, it leverages weakly augmented samples to guide the training of strongly augmented samples. Thus, filtering out less confident predictions using PL, particularly in the early stages of training, can effectively improve model performance. However, our ADS improves FixMatch by leveraging the relatively informed predictions overall. To sum up, our ADS that promotes informed predictions is beneficial for the SSL model to distill confident and reliable outputs.

### D. Safety With Different Backbone Structures

Intuitively, it should be safe to do candidate selection when every class is initialized to be evenly activated. Nevertheless, a concern arises when neural networks are unintentionally initialized which may prefer some specific classes at the very beginning but unfortunately fail to cover the true label. To address this concern, we apply two different neural network architectures on CIFAR-10 with 250 training labels as an example. One is CNN13 [11] which is typically initialized with the uniform distribution, and the other is the default ResNet which is initialized with the normal distribution. We record the average activations $\bar{m}$ and Top-$m$ accuracy over all unlabeled training data in Fig. 5, where example-wise $m$ is the number of nonzero outputs. Note that “Top-$m$ accuracy” denotes the success rate of partial activations including the true label, and it degenerates to the standard accuracy if $m = 1$.

Fig. 5(a) shows that CNN13 activates all the classes at the beginning ($\bar{m} \approx 10$) and then rules out unrelated classes till convergence. Contrastingly, the number of activations for ResNet surges from the one-hot state ($\bar{m} = 1$) in the first few epochs, then drops to a low value, and gradually reaches a steady value afterward. Fig. 5(b) shows that ResNet achieves better performance than CNN13 in terms of both two

![Fig. 4. Converged curves of distillation loss $J_{D}$ and average dominant probability $\bar{p}(1)$ for unlabeled training samples on MNIST dataset. We smooth the loss values for a better visualization. For ADS, $\bar{p}(1)$ is collected and calculated by replacing sparsemax with softmax which does not change the training process. (a) ME. (b) SH. (c) PL. (d) NS. (e) ADS.](image-url)
and fixed Top-$m$ experiments. Classification [51], and we compare them as a new group of picked (with the remaining truncated) and distilled during the counterparts in which only Top-3 and Top-2 predictions are reasonable because not all the augmentations are meaningful for rational observations imply that the selected candidates of ADS are accuracy as we expect, there exists a considerable gap between reject our model might yield confident predictions. It is measurements although it starts with a biased initialization [Fig. 5(a)], experimentally demonstrating that initialization is not an issue for the candidate selection of ADS. In addition, since the standard accuracy is upper bounded by Top-$m$ accuracy as we expect, there exists a considerable gap between CNN13 (ResNet) and its corresponding Top-1 accuracy. This observation implies that the selected candidates of ADS are always meaningful.

Notably, neither ResNet nor CNN13 converges to a one-hot format in the above experiment; the converged average activations $\tilde{m}$ of them are around 3 and 2, respectively. A possible reason for this phenomenon is that the distillation loss of MixMatch is explicitly defined on the manually augmented unlabeled data instead of the natural ones. For example, if an unlabeled sample is generated via some strong corruption, our model might reject to yield confident predictions. It is rational because not all the augmentations are meaningful for classification. A further question is if they are equivalent to the counterparts in which only Top-3 and Top-2 predictions are picked (with the remaining truncated) and distilled during the entire training stage. This thought is also known as set-valued classification [51], and we compare them as a new group of experiments. Fig. 5(c) presents the accuracy comparison between ADS and fixed Top-$m$ selection. We can see both two backbones have better accuracy when ADS is used, where their activations are dynamically changed indicated by Fig. 5(a). In addition, it is also observed that the top-$m$ selection suffers intense fluctuation on the accuracy, which is rooted in the unstable optimization of the plain truncation. Particularly, the accuracy drop for CNN13 is more severe than ResNet, which suggests another benefit of residual structure.

E. Observation of Prediction Histograms

We further verify the mechanism of ADS by recording the prediction frequency of unlabeled training data on MNIST and CIFAR-10 dataset. To this end, we evenly partition the prediction space into ten intervals and let each of them serve as a bin. Each bin is a half-closed interval except the last one. For example, the first bin is defined as $[0, 0.1)$ and the last is $[0.9, 1)$. Each prediction of an unlabeled sample, i.e., $p_i (1 \leq i \leq K)$, will fall in a bin based on its value. The experiments are executed following the settings of Section VI-C where VAT serves as the base model.

Fig. 6 presents the numerical distribution of prediction values over a series of epochs, where the maximum epoch is 100 because they are found not clearly altered after 100 epochs optimization. For MNIST, the prediction values are around 0.1 for every sample at the beginning. Then they gradually shift to other bins with the process of optimization. Eventually, almost all the predictions fall into the first and last bin, which means the predictions are distilled to be sufficiently certain. Regarding CIFAR-10, the results are a bit different. The initialization shows the model has some preference as mentioned in Section VI-D. In addition, the predictions grouped into the last bin are relatively fewer than that of MNIST. This is because the discriminative information for CIFAR-10 images is harder to capture via unlabeled data. In other words, given a semi-supervised model and training data, ADS tries to utilize the informed predictions and leaves the unreliable samples underfitted (This will not incur a low test confidence, please refer to Appendix C for an example).

F. Running Time Comparison

To demonstrate the analyses in Section V-C, we record the wall-time cost per epoch of different methods. The first group of experiments collect the execution time of VAT with different distillation strategies on CIFAR-10 with 4000 labels. The second and third groups collect the time cost with the baselines of MixMatch and FixMatch on CIFAR-100 with 10 000 labels, respectively. For a fair comparison, each method is run for 250 epochs with the same computation resources provided to all. The results are collated and shown as Fig. 7.

From the left of Fig. 7, we can observe that: 1) all the distillation components introduce an extra computation to the vanilla VAT (i.e., “W/O”); they commonly increase around 10 s per epoch. 2) compared with the other four distillation strategies, the additional cost of using ADS can be considered negligible, averaging less than 3 s per epoch. Again, from the right of Fig. 7, we can also see that ADS does not cost much to compute, which agrees with our analyses in Section V-C.

G. Scalability to ImageNet

We evaluate ADS on ImageNet to verify its scalability. FixMatch is employed here due to its superior performance. As a convention of [12], 10% training data are used as labeled data and the rest are treated as unlabeled samples. For a fair comparison, we do not exhaustively modify model hyper-parameters. The reported top-1 and top-5 error rate of FixMatch are $28.54 \pm 0.52\%$ and $10.87 \pm 0.28\%$, respectively. By conducting FixMatch-ADS on this task, we observe that
Fig. 6. Numerical distribution of prediction values of VAT+ADS on unlabeled training data. The top row shows the result on MNIST and the bottom row shows the result on CIFAR-10. The initialization of the network is used as their default. (a) Epoch = 1. (b) Epoch = 10. (c) Epoch = 20. (d) Epoch = 30. (e) Epoch = 40. (f) Epoch = 50. (g) Epoch = 100. (h) Epoch = 1. (i) Epoch = 10. (j) Epoch = 20. (k) Epoch = 30. (l) Epoch = 40. (m) Epoch = 50. (n) Epoch = 100.

TABLE III

| Method          | Time (s) |
|-----------------|----------|
| VAT             | 14.72±0.23 |
| FixMatch        | 5.92±0.14 |
| Sp              | 13.15±0.25 |
| Sp+ME           | 13.59±0.20 |
| Sp+PL           | 14.26±0.35 |
| Sp+NS           | 13.02±0.32 |
| ADS             | 12.40±0.31 |

Fig. 7. Running time (s) comparisons over epochs. VAT (i.e., “W/O”) with different distillation strategies on CIFAR-10 (4000 labels) (left). MixMatch and FixMatch based methods on CIFAR-100 (10 000 labels) (right).

the classification performance is improved on both two metrics to a certain degree. FixMatch-ADS achieves 26.90 ± 0.58% top-1 error rate and 10.25 ± 0.24% top-5 error rate. Meanwhile, by recording the running time, we find out that compared with the naive FixMatch, the extra time cost per epoch for FixMatch-ADS is averagely less than 20 s, demonstrating that ADS is an easy plug-in distillation strategy for SSL methods in practice.

H. Ablation Study

In this section, VAT and FixMatch are used as baselines which are executed on CIFAR-10 with all but 4000 labeled data to deconstruct ADS and explore the impact of the sharpening power $r$.

1) Irreplaceable SH in ADS: Our ADS consists of two main components: sparsemax for candidates selection, and sharpening for enhancing the sparse predictions. We justify their dependence by replacing the sharpening of ADS with other distillation strategies. The experimental results are shown as Table III, where test errors are reported.

Table III exhibits that: 1) ADS achieves the best performance (with comparable variance values) among different combinations. 2) Sparsemax solely also improves the baseline methods because it encourages the consistency loss to focus on the confusing classes only. 3) However, combining sparsemax with other distillation approaches, such as Sparsemax + ME and sparsemax + PL, may hurt the performance. This is because both ME and PL expect the single activation, failing to take the advantages in candidates selection of sparsemax. 4) Sparsemax + NS seems a good combo, while getting the proper thresholds is not easy as NS is found threshold sensitive [26]. 5) Finally, we conclude that sparsemax and sharpening both contribute to ADS, and this combination is always better than the others.

2) Lazy Choice for the Value of Power $r$: The default value of the power $r$ follows the convention of the MixMatch [22], i.e., $r = 2$. We justify this lazy choice for ADS by studying its impact with different $r$ values. To this end, we execute VAT+ADS and FixMatch-ADS on CIFAR-10 (4000 labels) by traversing $r$ from 1 to 4 with the step size of 0.5. The test errors are eventually recorded in Fig. 8.

From Fig. 8, we observe that: 1) A larger $r$, e.g., $r > 3$, often leads to an unsatisfactory performance with slightly larger variances. This is understandable because a class with the highest probability but not the actual true label will be over-encouraging, making the model trapped in some local minimums. 2) Although $r = 2$ is not always the best choice for ADS (FixMatch-ADS achieves the best performance when...
Considering the distillation loss with the power \( r = 2 \), we have the target probability

\[
t = \begin{cases} 1, & s > \frac{e}{e + 1} \\ \frac{1}{e + 1} \leq s \leq \frac{e}{e + 1} \\ 0, & s < \frac{1}{e + 1} \end{cases}
\]  

(21)

where \( t^* \) is calculated as follows:

\[
t^* = \frac{(\ln \frac{s}{r} + 1)^2/4}{(\ln \frac{s}{r} + 1)^2/4 + (1 - (\ln \frac{s}{r} + 1)/2)^2} = \frac{1}{1 + (\ln \frac{e - s}{e + s})^2}.
\]  

(22)

It is easily verified that \( t(s) \) is an odd function and \( t^* \big|_{s=0.5} = 0.5 \). Based on (21), we know the distillation loss (10) is nonzero iff \( (1/(e + 1)) \leq s \leq (e/(e + 1)) \)

\[
J_D(s) = t \ln t + (1 - t) \ln (1 - t)
\]  

reduced form

As \( t \) will stop the gradients from propagation, let \( s^* := (\ln (s/(1 - s)) + 1)/2 \), and only the reduced form has the gradient

\[
\nabla_t J_D = \frac{t - s}{s^*}, \quad \nabla_s J_D^* = \frac{t^* - s^*}{1 - s^*} \left( \frac{1}{s^*} + \frac{1}{1 - s^*} \right).
\]  

(24)

The above derivatives are used to complete the plot of ADS in Fig. 3, and other distillation methods are displayed in a similar manner.

**APPENDIX B**

**Proof for Corollary 1**

*Proof:* A prediction \( p \) is said determinate for ADS if \( p = p(1) \) and \( p(1) \geq e p(2) \). Obviously, for a \( K \)-way classification the maximum \( \theta_1 \) is picked if \( p(3) = p(4) = \cdots = p(K) = 0 \). Combining with the equality \( \sum_k p(k) = 1 \), we have \( \arg \max \theta_1 = (e/(e + 1)) \). Meanwhile, the minimum \( \theta_1 \) is picked if \( p(2) = p(3) = \cdots = p(K) \). Thus we have arg min \( \theta_1 = (e/(e + K - 1)) \).

Let \( \rho \) denote the population of nonzero outputs given a logits vector \( z \). According to the solution of sparsemax, if \( \rho < K \), we have the inequality

\[
1 + (\rho + 1)z_{(\rho + 1)} \leq \sum_{j=1}^{\rho+1} z_{(j)}.
\]  

(25)

Following the derivation of (16), we rewrite (25) as:

\[
1 + (\rho + 1) \ln C p(\rho + 1) \leq \sum_{j=1}^{\rho+1} \ln C p(j) \quad \Rightarrow p(\rho + 1) \leq \left( \frac{1}{\rho} \prod_{j=1}^{\rho} p(j) \right)^{1/2}.
\]  

(26)
which suggests \(\arg\max \theta \) over bins, and it is shifted leftward by a half of bin-width to visually align the dashed gray line in the right subfigure denotes the ideal average confidence the definition of calibration. To this end, we simply replace sparsemax with

![Fig. 9. Calibration performance on the test data of MNIST. Note that](image)

We use the expected calibration error (ECE) metric \([54]\) to evaluate VAT

Fitting may incur a concern about the calibration performance. Even without any help of other model calibration techniques.

\[\text{ECE} \text{ (in percent)} = \frac{1}{N} \sum_{i=1}^{N} \left( \frac{1}{B} \sum_{b=1}^{B} \left( \hat{y}_i^b - y_i \right)^2 \right),\]

where \(\hat{y}_i^b\) is the predicted probability of class \(y_i\) for the \(b\)-th bin. The minimum of \(\text{ECE}\) is derived if \(\tilde{p}(\rho + 1) = (\rho + e)\).

The above bounds complete the proof to the range of threshold for \(\theta_1\) and \(\theta_2\) in ADS in terms of softmax output space. \(\square\)

**APPENDIX C**

**EXAMPLE OF CALIBRATION EVALUATION**

Leaving a few of hard unlabelled training samples underfit may incur a concern about the calibration performance. We use the expected calibration error (ECE) metric \([54]\) to evaluate VAT + ADS on MNIST as an example. In a similar manner to \([27]\), we collect the test performance and calculate the samples proportion and accuracy over evenly distributed bins partitioned by prediction confidences (i.e., dominant probabilities).

Fig. 9 shows the experimental results, from which we obtain that: 1) The majority of test samples are predicted with a very high confidence, and some low confidence bins are even empty. This implies that the existence of some underfitted unlabelled samples will not lead the low confidence on test data prediction. 2) The average confidence is slightly greater than the accuracy and the output confidence is only a bit lower than the ideal case. 3) Quantitatively, we calculate \(\text{ECE} = 5.2\%\) which is not a big value for calibration error. Hence, we roughly say the model with ADS is well-calibrated even without any help of other model calibration techniques.

**REFERENCES**

[1] O. Chapelle and B. Scholkopf, *Semi-Supervised Learning*. Cambridge, MA, USA: MIT Press, 2006.

[2] X. Zhu, Z. Ghahramani, and J. D. Lafferty, “Semi-supervised learning using Gaussian fields and harmonic functions,” in *Proc. 20th Int. Conf. Mach. Learn.*, 2003, pp. 912–919.

[3] R. Hadsell, S. Chopra, and Y. LeCun, “Dimensionality reduction by learning an invariant mapping,” in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2006, pp. 1735–1742.

[4] L. Wang, R. Chan, and T. Zeng, “Probabilistic semi-supervised learning via sparse graph structure learning,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 2, pp. 853–867, Feb. 2021.

[5] T. Miyato, A. M. Dai, and I. Goodfellow, “Adversarial training methods for semi-supervised text classification,” 2016, arXiv:1605.07725.

[6] K. Sohn et al., “FixMatch: Simplifying semi-supervised learning with consistency and confidence,” 2020, arXiv:2001.07685.

[7] Y. Wang, W. Wang, Y. Liang, Y. Cai, J. Liu, and B. Hooi, “NodeAug: Semi-supervised node classification with data augmentation,” in *Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2020, pp. 207–217.

[8] T. Huynh, A. Nibali, and Z. He, “Semi-supervised learning for medical image classification using imbalanced training data,” *Comput. Methods Programs Biomed.*, vol. 216, Apr. 2022, Art. no. 106628.

[9] M. Sajjadi, M. Javanmardi, and T. Tasdizen, “Regularization with stochastic transformations and perturbations for deep semi-supervised learning,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 1163–1171.

[10] L. Laine and T. Aila, “Temporal ensembling for semi-supervised learning,” 2016, arXiv:1610.02242.

[11] A. Tarvainen and H. Valpola, “Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 1195–1204.

[12] Q. Xie, Z. Dai, E. Hovy, M.-T. Luong, and Q. V. Le, “Unsupervised data augmentation for consistency training,” in *Proc. NIPS*, vol. 33, 2020, pp. 6256–6268.

[13] S. Li, F. Liu, L. Jiao, P. Chen, and L. Li, “Self-supervised self-organizing clustering network: A novel unsupervised representation learning method,” *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Jun. 29, 2022, doi: 10.1109/TNNLS.2022.3185638.

[14] P. Zhu, J. Li, Y. Wang, B. Xiao, S. Zhao, and Q. Hu, “Collaborative decision-reinforced self-supervision for attributed graph clustering,” *IEEE Trans. Neural Netw. Learn. Syst.*, early access, May 18, 2022, doi: 10.1109/TNNLS.2022.3171583.

[15] Y. Grandvalet and Y. Bengio, “Semi-supervised learning by entropy minimization,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2005, pp. 529–536.

[16] G. Niu, B. Dai, M. Yamada, and M. Sugiyama, “Information-theoretic semi-supervised metric learning via entropy regularization,” *Neural Comput.*, vol. 26, no. 8, pp. 1717–1762, Aug. 2014.

[17] T. Miyato, S. Maeda, M. Koyama, and S. Ishii, “Virtual adversarial training: A regularization method for supervised and semi-supervised learning,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 8, pp. 1979–1993, Aug. 2019.

[18] A. Oliver, A. Odena, C. A. Raffel, E. D. Cubuk, and I. Goodfellow, “Realistic evaluation of deep semi-supervised learning algorithms,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 3235–3246.

[19] Y. Yang, Z. Song, I. King, and Z. Xu, “A survey on deep semi-supervised learning,” *IEEE Trans. Knowl. Data Eng.*, early access, Nov. 8, 2022, doi: 10.1109/TKDE.2022.3220219.

[20] T. Karras, M. Aittala, J. Hellsten, S. Laine, J. Lehtinen, and T. Aila, “Training generative adversarial networks with limited data,” 2020, arXiv:2006.06676.

[21] Q. Xie, Z. Dai, E. Hovy, M.-T. Luong, and Q. V. Le, “Unsupervised data augmentation for consistency training,” 2019, arXiv:1904.12848.

[22] D. Berthelot, N. Carlini, I. Goodfellow, N. Papernot, A. Oliver, and C. A. Raffel, “MixMatch: A holistic approach to semi-supervised learning,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 5049–5059.

[23] D. Berthelot et al., “RemixMatch: Semi-supervised learning with distribution alignment and augmentation anchoring,” 2019, arXiv:1911.09785.

[24] D. H. Lee, “Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks,” in *Proc. Workshop Challenges Represent. Learn.*, 2013, vol. 3, no. 2, pp. 1–6.

[25] E. Arazo, D. Ortego, P. Albert, N. E. O’Connor, and K. McGuinness, “Pseudo-labeling and confirmation bias in deep semi-supervised learning,” in *Proc. Int. Joint Conf. Neural Netw.* (IJCNN), Jul. 2020, pp. 1–8.

[26] J. Chen, V. Shah, and A. Kyrillidis, “Negative sampling in semi-supervised learning,” in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 1704–1714.
