Robust Semi-supervised Learning by Wisely Leveraging Open-set Data

Yang Yang, *Member, IEEE*, Nan Jiang, Yi Xu, and De-Chuan Zhan

Abstract—Open-set Semi-supervised Learning (OSSL) holds a realistic setting that unlabeled data may come from classes unseen in the labeled set, *i.e.*, out-of-distribution (OOD) data, which could cause performance degradation in conventional SSL models. To handle this issue, except for the traditional in-distribution (ID) classifier, some existing OSSL approaches employ an extra OOD detection module to avoid the potential negative impact of the OOD data. Nevertheless, these approaches typically employ the entire set of open-set data during their training process, which may contain data unfriendly to the OSSL task that can negatively influence the model performance. This inspires us to develop a robust open-set data selection strategy for OSSL. Through a theoretical understanding from the perspective of learning theory, we propose **Wise Open**-set Semi-supervised Learning (WiseOpen), a generic OSSL framework that selectively leverages the open-set data for training the model. By applying a gradient-variance-based selection mechanism, WiseOpen exploits a friendly subset instead of the whole open-set dataset to enhance the model's capability of ID classification. Moreover, to reduce the computational expense, we also propose two practical variants of WiseOpen by adopting low-frequency update and loss-based selection respectively. Extensive experiments demonstrate the effectiveness of WiseOpen in comparison with the state-of-the-art.

Index Terms—Semi-supervised Learning, OOD Detection, Open-set Data.

1 Introduction

Semi-supervised learning (SSL) [1], [2] leverages the ubiquitous unlabeled data to break the limitation of supervised learning (SL) caused by the huge human and financial costs in obtaining labeled data [3], [4], [5], [6]. There exist various techniques for SSL, such as consistency regularization [7], [8], [9] and entropy minimization [10], [11]. Moreover, some recent holistic approaches [12], [13], [14], [15], [16], [17] which integrate the techniques from dominant SSL paradigms have successfully achieved excellent performance on many benchmarks.

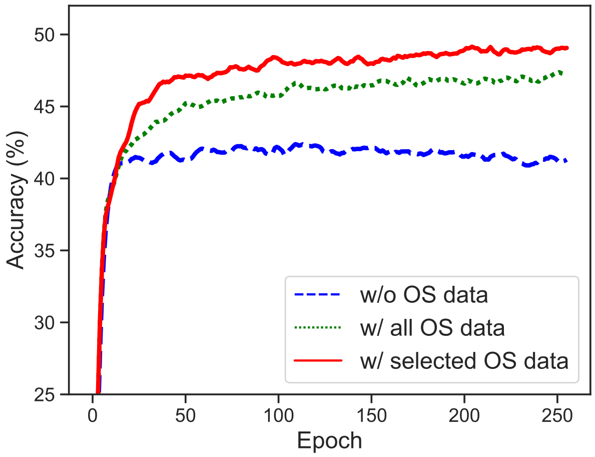


Fig. 1: An example of models' performance (testing accuracy on ID classification) with different strategies of using the **o**pen-**s**et data (OS data) illustrates the effectiveness of selectively leveraging OS data during the training process. Experiments are conducted on Tiny-ImageNet at 120 seen classes with 50 labels for each class. We employ the following methods: (1) Labeled Only (w/o OS data), an SL method only trained with labeled data; (2) OpenMatch [18] (w/ all OS data), an OSSL method trained with all OS data; and (3) WiseOpen-L on top of OpenMatch(w/ selected OS data), an OSSL method trained with selected OS data.

Despite all these achievements acquired by SSL, traditional SSL typically makes the assumption that labeled data and unlabeled data share the same class space [19], [20]. However, in real applications, the unlabeled training dataset may contain the data from classes unseen in the labeled, *i.e.*, OOD data, which may induce the existing SSL models to overconfidently misclassify data from unseen classes to nearby seen classes [21], [22], [23]. Aiming at promoting SSL to more realistic scenarios, OSSL [18], [24], [25], [26] has been widely investigated. An ideal OSSL model should have the capability of tackling the following task: classifying the ID testing instances under the potential interference of the open-set training data. Most existing OSSL approaches [18], [25], [26] usually apply an OOD detection module in addition to the traditional ID classifier, for the purpose of acquiring the capability of differentiating OOD data from ID data, thereby avoiding their potential negative impacts on ID classifier training. Typically, all open-set (OS) data are in-

- *Yang Yang is with the school of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing 210094, China.*
  E-mail: yyang@njust.edu.cn
- *Nan Jiang and De-Chuan Zhan are with the National Key Laboratory for Novel Software Technology, Nanjing University, and also with the School of Artificial Intelligence, Nanjing University, Nanjing 210023, China.*
  E-mail: jiangn@lamda.nju.edu.cn, zhandc@nju.edu.cn
- *Yi Xu is with the School of Control Science and Engineering, Dalian University of Technology, Dalian 116081, China.*
  E-mail: yxu@dlut.edu.cn

Corresponding authors: Yi Xu and De-Chuan Zhan.

volved in these OSSL models' training, which may comprise both friendly data and unfriendly data. Here *friendly data* means the data are beneficial to the considered OSSL task, while unfriendly can be the opposite. Due to the possible existence of unfriendly data, the model effectiveness may be affected if recklessly use the complete OS data set. On the other hand, discarding all OS training data is also not a wise choice, since it can result in the loss of valuable information contained in friendly OS data, ultimately leading to unsatisfactory performance. We can observe from Figure 1 that the method using all OS data has better ID classification accuracy compared with the method without using OS data, which reflects that in this example, certain OS data (friendly data) can enhance the ID classification accuracy. Furthermore, the method with selected OS data has the best performance on ID classification. That is to say, in this example, excluding certain unfriendly OS data can further improve the ID classification performance, revealing that the selection of OS training data is essential for the OSSL task. Moreover, to give an insight into what data can be friendly and to better demonstrate that utilizing friendly OS data can enhance model effectiveness while using unfriendly OS data can lead to performance degradation, we derive a theoretical analysis in Section 3.2. Inspired by both these empirical findings and theoretical analyses, we aim to carefully leverage selected OS data in OSSL.

To this end, we propose a generic OSSL framework called **Wise Open**-set Semi-supervised Learning (**WiseOpen**), which selectively exploits the OS data depending on the gradient variance. Specifically, based on the theoretical analysis in Section 3.2 that friendly open-set data can promote better generalization, for every epoch, WiseOpen first calculates the loss gradient of the limited labeled data as the approximate expectation of the gradient of the loss. Then based on the approximate gradient expectation and the gradient of each open-set instance, it computes the gradient variance for each open-set instance, and open-set data with smaller gradient variance will be selected as friendly data to train the OSSL model along with the limited labeled data. By applying this Gradient-Variance-based Selection Mechanism (**GV-SM**), the model can achieve better performance by the wiser exploitation of the open-set data. Nevertheless, it is of high time cost to calculate the gradient variance for each instance in every epoch. To handle this tricky situation, we provide two practical and economic variants of WiseOpen named **WiseOpen-E**conomic and **WiseOpen-L**oss. WiseOpen-E sets a larger interval of updating the friendly open-set data set obtained by GV-SM, which can effectively make a balance between the time cost and the model performance. But meanwhile, WiseOpen-E will inevitably suffer from the stale selection issue which may harm the model performance. On the other hand, WiseOpen-L employs loss values as a substitute for the gradient variance to select friendly data, which can address the time-consuming problem without raising the stale-selecting problems. WiseOpen-L can lead to inclusiveness of some previous SSL and OSSL methods using certain loss-based or confidence-based selection mechanisms, like [14], [15], [27]. We theoretically demonstrate the rationality and feasibility of replacing GV-SM in WiseOpen and WiseOpen-E with Loss-based Selection Mechanism (**L-SM**) in WiseOpen-L. Experiments on CIFAR-10/100 [28] and Tiny-ImageNet [29] show that our naive approach, *i.e.*, WiseOpen can achieve outstanding performance on the OSSL tasks by wisely leveraging the OS data while the two variants can also outperform the baselines. To summarize, our main contributions are:

- From the perspective of learning theory, we put forward an insight into the necessity of selectively leveraging the friendly open-set data in OSSL scenarios.
- We propose a robust general OSSL framework WiseOpen that employs GV-SM to wisely select friendly open-set data. This provides the OSSL community with a plug-and-play module to enhance the models' performance.
- We further provide WiseOpen-E and WiseOpen-L as two practical variants of WiseOpen, which can make the selection procedure more computation-friendly while still yielding performance improvements.
- The effectiveness of our proposed WiseOpen and its variants is demonstrated by extensive experiments on three popular benchmark datasets.

2 Related Work

Semi-supervised Learning. SSL aims at leveraging the unlabeled data to improve the model's performance without the extra cost of data annotation. With the advancement of deep learning [30], a number of deep SSL methods have been reported. For example, entropy minimization methods [11] focus on preventing the model from producing a flat prediction. Consistency regularization methods [7], [8], [9], [31] encourage the model to output the same results between differently augmented inputs. Additionally, holistic approaches like MixMatch [12] and FixMatch [14] successfully integrate some prior SSL techniques in a framework to gain better performance. Specifically, FixMatch is a simple but effective method that jointly employs consistency regularization and pseudo-labeling techniques and applies a fixed threshold for selecting high-confident unlabeled data to train the model. After the success of FixMatch, recent works like Dash [15], FlexMatch [32], and FreeMatch [27] further explore how to determine the suitable confidence thresholds according to model's learning status so that better exploit unlabeled data for better performance. Despite the achievements acquired by these SSL methods, they typically hold the assumption that the labeled data and unlabeled data share the same class space. When it comes to the realistic open-set scenario in this paper, they usually fail to do the job, as they may misclassify the data from unseen classes to certain nearby seen classes overconfidently.

Open-set Semi-supervised Learning. OSSL, an emerging branch of SSL, considers a more realistic scenario that the unlabeled data may come from unseen classes except for seen classes. A variety of methods [18], [24], [25], [26], [33], [34], [35], [36], [37] has been proposed in recent years. For example, D3SL [25] applies meta-learning to obtain a weight function for alleviating the impact of OOD data. MTC [26] employs a multi-task curriculum framework to leverage the unlabeled data in calculating the MixMatch [12] loss. OpenMatch [18] adopts one-vs-all classifiers with soft open-set consistency regularization and incorporates FixMatch to handle the OSSL tasks. OpenCos [35] utilizes the model pre-trained by contrastive learning to identify the pseudo-ID and pseudo-OOD data which will be used for fine-tuning the pre-trained model with certain SSL loss plus an auxiliary loss that assigns soft labels to pseudo-OOD data. Although these methods have been reported to get respectable performance, they typically utilize all open-set data to train the models, which can lead to performance degradation caused by unfriendly open-set data. One recent method IOMatch [37] has proposed to select confident pseudo-ID data while calculating unlabeled inlier loss, and exclude unlabeled data with low confidence in open-set predictions while calculating open-set loss where all unseen classes are regarded as one single class. In other words, IOMatch separately designs the selection mechanism specialized for each single loss based on the prediction confidence, aiming at conquering the dilemma of unreliable results in the training procedure. In contrast, with an insight into selecting friendly data from the whole learning task perspective, we propose applying GV-SM or L-SM over the unified unsupervised loss, which is more generic, and is orthogonal and complementary to prior methods.

Out-of-distribution Detection. OOD detection aims at detecting outliers that belong to the different distributions from the training distributions while ensuring that the ID classification is not adversely affected [38]. Existing OOD detection methods [39], [40], [41], [42], [43], [44] have been reported to achieve excellent performance in tackling this task. However, these methods usually acquire abundant labeled data from seen classes and some methods [45], [46], [47] even additionally utilize external OOD knowledge in the training phase. In contrast, OSSL only has limited access to the labeled data of seen classes and needs to handle the unlabeled data from unseen classes in training, which renders the task more challenging.

3 WiseOpen: Wisely Leverage the Open-set Data

In this section, we first provide the preliminary of our work. Then through a theoretical understanding, we demonstrate the effectiveness of excluding unfriendly open-set data and utilizing friendly open-set data. Based on this, we then specify the selection mechanisms applied in our proposed approaches, which can help models gain better performance in ID classification by wisely leveraging the open-set data. Moreover, at the end of this section, we will clarify the relationships between our proposed frameworks and the existing OSSL methods.

3.1 Preliminary

Let $\mathcal{S} = \{(\mathbf{x}_i^l, \mathbf{y}_i)\}_{i=1}^{N_l}$ be the labeled training data set, where $\mathbf{x}_i^l$ is an ID example from one of the K seen classes and $\mathbf{y}_i$ is its one-hot label. Let $\mathcal{U} = \{\mathbf{x}_i^u\}_{i=1}^{N_u}$ be the unlabeled training data set, where $\mathbf{x}_i^u$ is an unlabeled training instance drawn from the seen classes or unseen classes. Typically, the overall objective function $\mathcal{L}$ of OSSL, no matter what specific techniques are applied, can be written as

$$\mathcal{L} = \mathcal{L}_s(\theta; \mathcal{S}) + \mathcal{L}_u(\theta; \mathcal{U}), \tag{1}$$

TABLE 1: Frequently used notations along with their mathematical meaning.

Notation	Mathematical Meaning
K	Number of the seen classes.
$\mathcal{S} = \{(\mathbf{x}_i^l, \mathbf{y}_i)\}_{i=1}^{N_l}$	Labeled training dataset containing N_l labeled pairs$(\mathbf{x}_i^l, \mathbf{y}_i)$.
$\mathcal{U} = \{\mathbf{x}_i^u\}_{i=1}^{N_u}$	Original unlabeled training dataset containing N_u instance $\mathbf{x}_i^u$.
$\mathcal{U}_t$	Selected unlabeled subset in the t-th epoch.
$\mathcal{L}, \mathcal{L}_s, \mathcal{L}_u$	The overall loss, supervised loss, and unsupervised loss.
θ	Parameters of the model.
$g(\theta)$	Stochastic gradient of loss function computed at θ.
E[$\cdot$]	Mathematical expectation of some random variable.
ERB($\cdot$)	The excess risk bound given the model parameters.
$\lvert \cdot \rvert$	Cardinality of the given set.

where $\mathcal{L}_s$ and $\mathcal{L}_u$ represent supervised loss and unsupervised loss respectively and θ is the parameters of the model. The main goal of OSSL is to learn an ID classification model described by the parameter θ optimized by minimizing $\mathcal{L}$ on $\mathcal{S}$ and $\mathcal{U}$. And the most tricky problem is how to handle the open-set data set $\mathcal{U}$ so that better ID classification performance can be gained.

Previous studies have presented different answers to this tricky problem. Taking OpenMatch [37] as an example, given a labeled example $\mathbf{x}_i^l$, it will acquire a K-dimensional probability vector $\mathbf{p}(\theta; \mathbf{x}_i^l)$ by a traditional ID classifier and K 2-dimensional probability vector $\{\mathbf{q}^k(\theta; \mathbf{x}_i^l) = (q_0^k(\theta; \mathbf{x}_i^l), q_1^k(\theta; \mathbf{x}_i^l))\}_{k=1}^K$ by an OOD detector which is composed of K binary sub-classifier. Here $q_0^k(\theta; \mathbf{x}_i^l)$ indicates the probability of being an inlier of class k while $q_1^k(\theta; \mathbf{x}_i^l)$ indicates not. For simplicity, we will hide the model's parameters θ and reduce the notation $\mathbf{p}(\theta; \mathbf{x}_i^l)$ and $\mathbf{q}^k(\theta; \mathbf{x}_i^l) = (q_0^k(\theta; \mathbf{x}_i^l), q_1^k(\theta; \mathbf{x}_i^l))$ to $\mathbf{p}(\mathbf{x}_i^l)$ and $\mathbf{q}^k(\mathbf{x}_i^l) = (q_0^k(\mathbf{x}_i^l), q_1^k(\mathbf{x}_i^l))$, respectively. Then to train the model upon labeled training data set $\mathcal{S}$, OpenMatch will compute the following losses:

$$\mathcal{L}_{ce}(\theta; \mathcal{S}) = \frac{1}{|\mathcal{S}|} \sum_{(\mathbf{x}_i^l, \mathbf{y}_i) \in \mathcal{S}} H(\mathbf{y}_i, \mathbf{p}(\mathbf{x}_i^l)), \tag{2}$$

$$\mathcal{L}_{ova}(\theta; \mathcal{S}) = -\frac{1}{|\mathcal{S}|} \sum_{(\mathbf{x}_i^l, \mathbf{y}_i) \in \mathcal{S}} \log q_0^{y_i}(\mathbf{x}_i^l) + \min_{k \neq y_i} \log q_1^k(\mathbf{x}_i^l), \tag{3}$$

where $H(\cdot, \cdot)$ denotes the standard cross-entropy loss, $|\mathcal{S}| = N_l$ denotes the cardinality of S, and y_i denotes the ground truth of $\mathbf{x}_i^l$, *i.e.*, $\mathbf{x}_i^l$ belongs to class y_i. On the other hand, given an unlabeled instance $\mathbf{x}_i^u$, OpenMatch first applies standard random cropping $\alpha(\cdot)$ as weak data augmentation to obtain $\alpha_0(\mathbf{x}_i^u)$ and $\alpha_1(\mathbf{x}_i^u)$. Then it will calculate the following losses:

$$\mathcal{L}_{em}(\theta; \mathcal{U}) = -\frac{1}{|\mathcal{U}|} \sum_{\mathbf{x}_i^u \in \mathcal{U}} \sum_{j=0}^{1} \sum_{k=1}^{K} \mathbf{q}^k(\alpha_j(\mathbf{x}_i^u)) \log \mathbf{q}^k(\alpha_j(\mathbf{x}_i^u)), \tag{4}$$

$$\mathcal{L}_{oc}(\theta;\mathcal{U}) = \frac{1}{|\mathcal{U}|}\sum_{\mathbf{x}_i^u \in \mathcal{U}}\sum_{k=1}^{K}\|\mathbf{q}^k(\alpha_0(\mathbf{x}_i^u)) - \mathbf{q}^k(\alpha_1(\mathbf{x}_i^u))\|_2^2. \quad (5)$$

Moreover, it adopts FixMatch over the pseudo-ID instances which can be formulated as

$$\mathcal{L}_{fm}(\theta;\mathcal{U}) = \frac{1}{|\mathcal{U}|}\sum_{\mathbf{x}_i^u \in \mathcal{U}}\mathcal{M}(\alpha(\mathbf{x}_i^u))H(\hat{\mathbf{y}}_i^u, \mathbf{p}(\mathcal{A}(\mathbf{x}_i^u))), \quad (6)$$

where $\mathcal{A}(\cdot)$ indicates strong data augmentation like RandAugment [48] and $\mathcal{M}(\alpha(\mathbf{x}_i^u)) = \mathbb{I}(q_0^{\hat{y}_i^u}(\alpha(\mathbf{x}_i^u)) > 0.5) \cdot \mathbb{I}(\max(\mathbf{p}(\alpha(\mathbf{x}_i^u))) > \rho)$ in which ρ is a threshold, and $\hat{y}_i^u = \arg\min \mathbf{p}(\alpha(\mathbf{x}_i^u))$ represents the pseudo-label of $\mathbf{x}_i^u$ and $\hat{\mathbf{y}}_i^u$ will be its one-hot version. To summarize, the overall objective function of OpenMatch can be written as

$$\begin{aligned}\mathcal{L} = &\underbrace{\mathcal{L}_{ce}(\theta;\mathcal{S}) + \mathcal{L}_{ova}(\theta;\mathcal{S})}_{\mathcal{L}_s(\theta;\mathcal{S})} \\ &+ \underbrace{\lambda_1\mathcal{L}_{em}(\theta;\mathcal{U}) + \lambda_2\mathcal{L}_{oc}(\theta;\mathcal{U}) + \lambda_3\mathcal{L}_{fm}(\theta;\mathcal{U})}_{\mathcal{L}_u(\theta;\mathcal{U})}, \quad (7)\end{aligned}$$

where λ_1, λ_2 and λ_3 are trade-off parameters. It can be observed that OpenMatch recklessly employs the whole open-set data set $\mathcal{U}$ to train the model which is common in previous studies. Note that although in Eq.6, $\mathcal{M}(\alpha(\mathbf{x}_i^u))$ is applied for obtaining confident pseudo-ID instances, Eq.4 and Eq.5 still utilize the whole unlabeled data set. Thus, the entire model of OpenMatch actually still leverages all open-set data. Oppositely, we argue that it is necessary to selectively leverage the open-set data $\mathcal{U}$ which will be demonstrated in the following theoretical analysis.

3.2 Theoretical Understanding

In this subsection, we aim to understand the learning task from the perspective of generalization. For the simplicity of generalization analysis, we abstract the key points of the learning task and make it more math-friendly. For example, we do not consider data augmentations in our analysis. To this end, we formulate it as the following risk minimization (RM) problem, which is commonly used in Statistical Learning Theory:

$$\min_{\theta}\mathcal{L}(\theta) := \mathrm{E}_{\zeta\sim\mathcal{D}}[\ell(\theta;\zeta)], \quad (8)$$

where θ is the parameter to be learned, ζ is the training data following a unknown distribution $\mathcal{D}$, ℓ is the loss function and $\mathrm{E}[\cdot]$ is the expectation. Generally, it is impossible to know the loss function $\mathcal{L}(\theta)$ explicitly due to the unknown distribution of $\mathcal{D}$. Instead of solving problem (8) directly, we usually consider the following empirical risk minimization (ERM) problem: $\min_\theta \widehat{\mathcal{L}}(\theta) := \frac{1}{|\widehat{\mathcal{D}}|}\sum_{\zeta\in\widehat{\mathcal{D}}}\ell(\theta;\zeta)$, where $\widehat{\mathcal{D}}$ is a sampled training data set with sample size $|\widehat{\mathcal{D}}|$. To solve the ERM problem, stochastic gradient descent (SGD) is widely employed, whose key update step is $\theta_{t+1} = \theta_t - \eta\widehat{g}(\theta_t)$ with the learning rate $\eta > 0$, where θ_t indicates the parameter in t-th epoch. Due to the labeled (ID) and unlabeled (ID and OOD) data containing in $\widehat{\mathcal{D}}$, the stochastic gradient of loss function computed at θ_t can be rewritten as

$$\widehat{g}(\theta_t) = \lambda g_{\mathrm{id}}(\theta_t) + (1-\lambda)(\tau g_{\mathrm{fr}}(\theta_t) + (1-\tau)g_{\mathrm{uf}}(\theta_t)), \quad (9)$$

where $\lambda, \tau \in [0,1]$ are two constants, g_{id} is the stochastic gradient computed by ID data while g_{fr} and g_{uf} are two stochastic gradients computed by friendly unlabeled data and unfriendly unlabeled data respectively. We use the gradient variances to measure the distance between labeled data and open-set data. Specifically, we bound the variance for each stochastic gradient in the following assumptions.

Assumption 1 (Bounded variance [49]). *The stochastic gradient is unbiased,* $\mathrm{E}[\widehat{g}(\theta)] = \nabla\mathcal{L}(\theta)$. *The stochastic gradient* $g_{id}(\theta)$ *is variance bounded, i.e., there exists a constant* $\sigma^2 > 0$, *such that*

$$\mathrm{E}[\|g_{id}(\theta) - \nabla\mathcal{L}(\theta)\|^2] \leq \sigma^2.$$

Assumption 2 (Weak Growth Condition [50], [51]). *The stochastic gradient of* $g_{fr}(\theta)$ *and* $g_{uf}(\theta)$ *are variance bounded, i.e., there exists a constant* $\sigma^2 > 0$, *such that*

$$\mathrm{E}[\|g_{fr}(\theta) - \nabla\mathcal{L}(\theta)\|^2] \leq \frac{\epsilon}{2}\|\nabla\mathcal{L}(\theta)\|^2 + \sigma^2.$$

$$\mathrm{E}[\|g_{uf}(\theta) - \nabla\mathcal{L}(\theta)\|^2] \leq \frac{\nu}{2}\|\nabla\mathcal{L}(\theta)\|^2 + \sigma^2.$$

where $\epsilon > 0$ *is a small constant and* $\nu \gg 1$ *is a large enough constant.*

Since $\nu \gg 1$ is large enough, we can consider that the variance for $g_{\mathrm{uf}}(\theta)$ is much larger than the variance for $g_{\mathrm{fr}}(\theta)$. We consider the friendly data to have small variance so we suppose $\epsilon > 0$ is small. Similarly, we consider $\nu \gg 1$ is large enough for unfriendly data. In generalization analysis, we are interested in the excess risk bound (ERB):

$$\mathrm{ERB}(\widehat{\theta}) := \mathcal{L}(\widehat{\theta}) - \mathcal{L}(\theta_*), \quad (10)$$

where $\widehat{\theta}$ is a solution obtained by an algorithm and $\theta_* \in \arg\min_\theta \mathcal{L}(\theta)$ is the optimal solution of problem (8). For the convenience of analysis, we make the following widely used assumptions for the loss function.

Assumption 3 (Smoothness [52]). $\mathcal{L}(\theta)$ *is smooth with an* L*-Lipchitz continuous gradient, i.e., it is differentiable and there exists a constant* $L > 0$ *such that*

$$\|\nabla\mathcal{L}(\theta) - \nabla\mathcal{L}(\theta')\| \leq L\|\theta - \theta'\|.$$

This is equivalent to

$$\mathcal{L}(\theta) - \mathcal{L}(\theta') \leq \langle\mathcal{L}(\theta'), \theta - \theta'\rangle + \frac{L}{2}\|\theta - \theta'\|^2.$$

Assumption 4 (Polyak-Łojasiewicz condition [53]). *There exists a constant* $\mu > 0$ *such that*

$$2\mu(\mathcal{L}(\theta) - \mathcal{L}(\theta_*)) \leq \|\nabla\mathcal{L}(\theta)\|^2,$$

where $\theta_* \in \arg\min_\theta \mathcal{L}(\theta)$ *is a optimal solution.*

It is worth noting that the Polyak-Łojasiewicz condition has been theoretically [54] and empirically [55] observed in training deep neural networks. Weaker than many other conditions like strong convexity, restricted strong convexity, and weak strong convexity [56], it has been widely used to establish the convergence of non-convex optimization [57], [58], [59].

Under the standard assumptions on loss function, we have the following theorem for ERB. Due to the space limitation, we include the proof in the supplementary.

Theorem 1. *Under assumptions 1, 2, 3, 4, we have the following ERB in expectation:*
(a) when all data are used: by setting $\eta \leq \frac{1}{(1-\lambda)(\tau\epsilon+(1-\tau)\nu)L}$*, we have*

$$ERB(\widehat{\theta}_{id+uf+fr}) \leq O\left(\mathcal{L}(\theta_0) - \mathcal{L}(\theta_*)\right);$$

(b) when only labeled data are used: by setting $\eta = \frac{2}{n\mu}\log\left(\frac{n\mu^2(\mathcal{L}(\theta_0)-\mathcal{L}(\theta_*))}{\sigma^2 L}\right)$*, we have*

$$\begin{aligned} ERB(\widehat{\theta}_{id}) &\leq \frac{L\sigma^2}{n\mu^2} + \frac{2L\sigma^2}{n\mu^2}\log\left(\frac{n\mu^2(\mathcal{L}(\theta_0) - \mathcal{L}(\theta_*))}{\sigma^2 L}\right) \\ &\leq O\left(\frac{\log(n)}{n}\right); \end{aligned}$$

(c) when labeled and friendly data are used: by setting $\eta = \frac{2}{(n+m)\mu}\log\left(\frac{(n+m)\mu^2(\mathcal{L}(\theta_0)-\mathcal{L}(\theta_*))}{\sigma^2 L}\right)$*, we have*

$$\begin{aligned} ERB(\widehat{\theta}_{id+fr}) \leq& \frac{L\sigma^2}{(n+m)\mu^2} \\ &+ \frac{2L\sigma^2}{(n+m)\mu^2}\log\left(\frac{(n+m)\mu^2(\mathcal{L}(\theta_0) - \mathcal{L}(\theta_*))}{\sigma^2 L}\right) \\ \leq& O\left(\frac{\log(n+m)}{(n+m)}\right), \end{aligned}$$

where n *and* m *are the sample sizes of labeled data and friendly data, respectively,* $\widetilde{O}(\cdot)$ *suppresses a logarithmic factor and constants.*

The results of Theorem 1 show that (1) the algorithm could not reduce the objective due to the large variance arising from unfriendly open-set data; (2) by using friendly open-set data, the algorithm could significantly reduce the objective, and it has better generalization by comparing with the one only using labeled data. The theoretical findings inspire us to design a selection method for wisely leveraging open-set data. Specifically, one may carefully select and use friendly open-set data during training progress to improve the performance of the learning task.

3.3 Wise Selection Mechanism

Inspired by the theoretical understanding above, we aim to wisely select and exploit a subset of the original unlabeled data set $\mathcal{U}_t \subseteq \mathcal{U}$ that consists of the open-set data friendly to the model training in the t-th epoch so that we can learn a model from $\mathcal{U}_t$ and $\mathcal{S}$ with better capability of classifying ID instances.

To this end, based on Theorem 1, we design a gradient-variance-based selection mechanism (GV-SM) to discard the unfriendly open-set data with large gradient variance so that we can exploit the remaining relatively friendly open-set data to learn a model with better ID classification capability. Specifically, our proposed GV-SM of the t-th epoch can be formulated as

$$\mathcal{U}_t = \left\{\mathbf{x}_i^u \in \mathcal{U} \mid \|g_{\mathbf{x}_i^u}(\theta_t) - \bar{g}(\theta_t)\| < \sqrt{\rho_t}\right\}, \tag{11}$$

Algorithm 1: WiseOpen Family.

```
Input: Labeled data $\mathcal{S}$, unlabeled data $\mathcal{U}$, model
       parameters $\theta$, epoch $E_{max}$, iteration $I_{max}$,
       learning rate $\eta$, selection interval $e_s$.
for $t \leftarrow 1$ to $E_{max}$ do
    if $t$ % $e_s$ == 0 then
        Obtain $\mathcal{U}_t$ according to Eq.11 or Eq.19;
    else
        Obtain $\mathcal{U}_t = \mathcal{U}_{t-1}$;
    end
    for $iter \leftarrow 1$ to $I_{max}$ do
        Sample batches $\mathcal{B}_l \in \mathcal{S}$ and $\mathcal{B}_u \in \mathcal{U}_t$;
        Compute $\mathcal{L} \leftarrow \mathcal{L}_s(\theta; \mathcal{B}_l) + \mathcal{L}_u(\theta; \mathcal{B}_u)$;
        Update $\theta \leftarrow \theta - \eta\frac{\partial\mathcal{L}}{\partial\theta}$;
    end
end
```

where $g_{\mathbf{x}_i^u}(\theta_t)$ denotes the gradient computed by open-set instances $\mathbf{x}_i^u$, $\bar{g}(\theta_t)$ denotes the estimated expectation of the gradient of the overall objective function $\mathcal{L}$, and ρ_t indicates the threshold in the t-th epoch. In this paper, we utilize the labeled data set $\mathcal{S}$ to obtain $\bar{g}(\theta_t)$:

$$\bar{g}(\theta_t) = \frac{1}{N_l}\sum_{i=1}^{N_l}\frac{\partial\mathcal{L}_s(\theta_t; \mathbf{x}_i^l)}{\partial\theta_t}. \tag{12}$$

Meanwhile, we calculate $g_{\mathbf{x}_i^u}(\theta_t)$ by the following formula:

$$g_{\mathbf{x}_i^u}(\theta_t) = \frac{\partial\mathcal{L}_u(\theta_t; \mathbf{x}_i^u)}{\partial\theta_t}. \tag{13}$$

As for obtaining ρ_t, without loss of generality, we apply the following two simple methods: (1) Top-k, which utilizes the k-th largest gradient variance among $\{g_{\mathbf{x}_i^u}(\theta_t)\}_i^{N_u}$ as ρ_t; (2) Otsu thresholding [60], which adaptively determine ρ_t by maximizing the variance of $g_{\mathbf{x}_i^u}(\theta_t)$ between the selected and discarded open-set data clusters.

By applying this wise selection mechanism, WiseOpen reformulates the typical OSSL objective function in the t-th epoch as

$$\mathcal{L} = \mathcal{L}_s(\theta_t; \mathcal{S}) + \mathcal{L}_u(\theta_t; \mathcal{U}_t). \tag{14}$$

To be more specific, after implementing our proposed WiseOpen on top of OpenMatch, we will rewrite OpenMatch's objective formulated in Eq.7 as

$$\begin{aligned} \mathcal{L} =& \underbrace{\mathcal{L}_{ce}(\theta_t; \mathcal{S}) + \mathcal{L}_{ova}(\theta_t; \mathcal{S})}_{\mathcal{L}_s(\theta_t; \mathcal{S})} \\ &+ \underbrace{\lambda_1\mathcal{L}_{em}(\theta_t; \mathcal{U}_t) + \lambda_2\mathcal{L}_{oc}(\theta_t; \mathcal{U}_t) + \lambda_3\mathcal{L}_{fm}(\theta_t; \mathcal{U}_t)}_{\mathcal{L}_u(\theta_t; \mathcal{U}_t)}. \end{aligned} \tag{15}$$

3.4 Practical Variants

Ideally, the selection procedure should be implemented for every epoch, which is applied in the naive WiseOpen. However, it can be computationally expensive to calculate gradient variance for each unlabeled instance in every epoch, as shown in Table 3 in Section 4.2. Therefore, we proposed two practical variants of WiseOpen, namely WiseOpen-E and WiseOpen-L. The overall framework of the WiseOpen family, *i.e.*, WiseOpen and its variants, is summarized in Algorithm 1.

WiseOpen-E. Before introducing WiseOpen-E, let us consider an example as follows. The updating step of SGD at t-th epoch can be formulated as $\theta_t = \theta_{t-1} - \eta g(\theta_{t-1})$, then after m epochs, we have $\theta_{t+m} = \theta_{t-1} - \eta \sum_{k=0}^{m} g(\theta_{t-1+k})$. By the condition of smoothness variance with parameter L', we have

$$\|g(\theta_{t+m}) - g(\theta_t)\| \le L'\|\theta_{t+m} - \theta_t\| = \eta L' \left\| \sum_{k=1}^{m} g(\theta_{t-1+k}) \right\|. \tag{16}$$

Since η is small, if m is not large, then $\eta L' \left\| \sum_{k=1}^{m} g(\theta_{t-1+k}) \right\|$ is small, indicating that $g(\theta_{t+m})$ and $g(\theta_t)$ are close enough. Inspired by this example, gradients within a small updating interval could be similar. Thus, as a natural idea, an economical version of WiseOpen, WiseOpen-E simply sets an interval e_s of updating the selecting result of open-set data. Specifically, once a selection procedure in the t-th epoch is accomplished, the selected open-set subset will remain unchanged for $(e_s - 1)$ epochs, *i.e.*, $\mathcal{U}_t = \mathcal{U}_{t+1} = \cdots = \mathcal{U}_{t+e_s-1}$, and in the $(t + e_s)$-th epoch another selection procedure will be carried out to update the selected open-set subset. In our experiments of WiseOpen-E, e_s is set as 10 while for the experiments of WiseOpen and WiseOpen-L introduced later, e_s is set as 1. Nevertheless, as a trade-off of obtaining higher efficiency, the stale selection issue will inevitably arise in WiseOpen-E. This means the selected subset may be outdated for the current model training since the selection decision is based on the previous, older model.

WiseOpen-L. Considering in practice, calculating the loss value for each sample is significantly less computationally expensive compared to calculating the gradient variance, if loss value can be employed as a substitute for the gradient variance, then we can achieve computationally friendly data selection without raising the stale selection issue. Inspired by the Polyak-Łojasiewicz condition [53] of a loss function $\ell(\theta)$:

$$\ell(\theta) \le \frac{1}{2\mu}\|\nabla \ell(\theta)\|^2 + \ell(\theta_*), \tag{17}$$

where $\mu > 0$ is a constant and $\theta_* \in \arg\min_\theta \ell(\theta)$ is a optimal solution, we can make an informal connection between loss function and its gradient norm. If we apply the Polyak-Łojasiewicz condition to function $\mathcal{L}_u(\theta; \mathbf{x}_i^u)$, then we have

$$\mathcal{L}_u(\theta_t; \mathbf{x}_i^u) \le \frac{1}{2\mu}\|g_{\mathbf{x}_i^u}(\theta_t)\|^2 + \mathcal{L}_u(\theta_*; \mathbf{x}_i^u).$$

Once $\|g_{\mathbf{x}_i^u}(\theta_t) - \bar{g}(\theta_t)\| \le \sqrt{\rho_t}$ holds in (11), then by $\|g_{\mathbf{x}_i^u}(\theta_t) - \bar{g}(\theta_t)\| \ge \|g_{\mathbf{x}_i^u}(\theta_t)\| - \|\bar{g}(\theta_t)\|$ we have $\|g_{\mathbf{x}_i^u}(\theta_t)\| \le \sqrt{\rho_t} + \|\bar{g}(\theta_t)\|$, so that

$$\mathcal{L}_u(\theta_t; \mathbf{x}_i^u) \le \frac{(\sqrt{\rho_t} + \|\bar{g}(\theta_t)\|)^2}{2\mu} + \mathcal{L}_u(\theta_*; \mathbf{x}_i^u). \tag{18}$$

That is to say, if $\mathbf{x}_i^u \in \mathcal{U}_t$ in (11), i.e. $\|g_{\mathbf{x}_i^u}(\theta_t) - \bar{g}(\theta_t)\|$ is upper bounded, then $\mathcal{L}_u(\theta_t; \mathbf{x}_i^u)$ can aslo be upper bounded. Therefore, we propose another variant of WiseOpen, called WiseOpen-L which applies a loss-based selection mechanism (L-SM) that selects the friendly open-set data with smaller loss values to construct $\mathcal{U}_t$:

$$\mathcal{U}_t = \left\{ \mathbf{x}_i^u \in \mathcal{U} \mid \mathcal{L}_u(\theta_t; \mathbf{x}_i^u) < \rho_t' \right\}, \tag{19}$$

where in this paper, the threshold ρ_t' is also acquired by the Top-k or Otsu thresholding methods.

3.5 Relationships to Previous Methods

Inclusiveness to Previous Methods. Firstly, our proposed WiseOpen and its variants are designed upon the unified formulation of the overall objectives in the OSSL scenarios. Therefore, we are proposing robust generic OSSL frameworks that can be easily implemented into the existing OSSL and SSL methods to improve performance. Moreover, WiseOpen-L, as one of our proposed accelerated variants, actually can be seen as a general version of some previous SSL and OSSL methods that adopt confidence-based selection mechanisms. For example, when we exclude $\mathbb{I}(q^{\hat{y}_i^u}(\alpha(\mathbf{x}_i^u)) > 0.5)$ in $\mathcal{M}(\alpha(\mathbf{x}_i^u))$ of $\mathcal{L}_{fm}$ formulated in Eq.6, it will be the exact unsupervised loss $\mathcal{L}_u$ in FixMatch [14]. And the remaining selecting component $\mathbb{I}(\max(\mathbf{p}(\alpha(\mathbf{x}_i^u))) > \rho)$ can converted to $\mathbb{I}(-\log\max(\mathbf{p}(\alpha(\mathbf{x}_i^u))) < -\log\rho)$, which actually can be seen as a kind of loss-based selection mechanism depending on the cross-entropy loss over $\alpha(\mathbf{x}_i^u)$ using its one-hot pseudo-label.

Differnce between Exisiting Selection Mechanisms. Although previous studies have exploited various selection mechanisms, we distinguish our proposed method in the following aspects: (1) Our frameworks apply the selection mechanism on the whole learning task level, or in other words, on the unified unsupervised loss level. In contrast, the previous methods usually employ selection mechanisms on a single loss function level. For example, in OpenMatch, $\mathcal{L}_{fm}$ formulated in Eq.6 uses 0.5 and ρ as the threshold to select the pseudo-ID instances with high confidence in ID classification. But this selection is just applied to this single loss while $\mathcal{L}_{em}$ and $\mathcal{L}_{oc}$ formulated in Eq.4 and Eq.5 still leverage the whole open-set data, which may harm the model's performance due to the existence of the unfriendly open-set data. (2) As far as we know, different from the popular confidence-based selection mechanisms used in the previous work [14], [18], [27], [37], we are the first to propose selecting open-set data based on the gradient variance with solid theoretical analysis in the OSSL scenarios.

4 EXPERIMENTS

In this section, we introduce the comprehensive settings of experiments and provide sufficient evaluations to demonstrate the effectiveness of our proposed frameworks.

4.1 Experimental Settings

Datasets. Following [18], [21], [26], [61], [62], we choose three benchmark datasets to evaluate the efficacy of WiseOpen, namely:

- CIFAR-10 [28], a dataset consisting of 10 classes, of which each class contains 5,000 and 1,000 images for training and testing respectively;
- CIFAR-100, a dataset consisting of 100 classes, of which each class contains 500 and 100 images for training and testing respectively;

TABLE 2: The hyper-parameter k for the Top-k threshold.

(a) k utilized in the Top-k threshold for GV-SM followed with the proportion to the whole open-set.

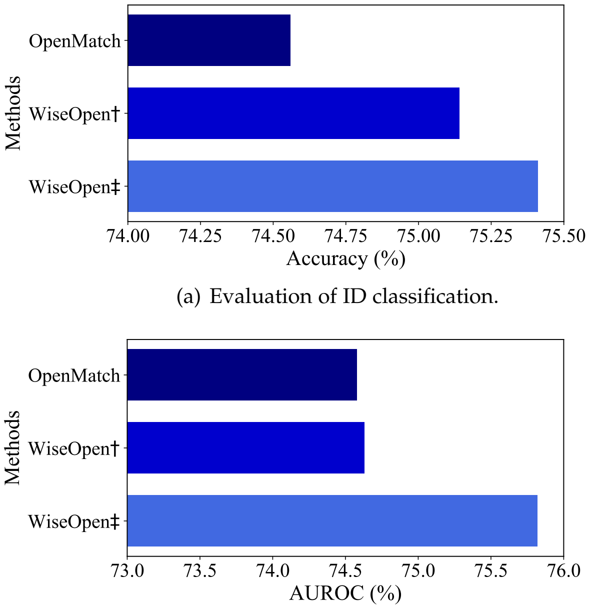
	CIFAR-10			CIFAR-100		Tiny-ImageNet
Vanilla Baselines	50 labels	100 labels	400 labels	50 labels	100 labels	50 labels
FixMatch	2470 (5%)	2455 (5%)	2365 (5%)	880 (2%)	410 (1%)	1760 (2%)
FreeMatch	2470 (5%)	2455 (5%)	4730 (10%)	6600 (15%)	4100(10%)	17600 (20%)
MTC	4940 (10%)	4910 (10%)	2365 (5%)	4400 (10%)	4100 (10%)	4400 (5%)
OpenMatch	2470 (5%)	2455 (5%)	2365 (5%)	4400 (10%)	4100 (10%)	4400 (5%)
IOMatch	2470 (5%)	982 (2%)	946 (2%)	880 (2%)	2050 (5%)	1760 (2%)

(b) k utilized in the Top-k threshold for L-SM followed with the proportion to the whole open-set.

	CIFAR-10			CIFAR-100		Tiny-ImageNet
Vanilla Baselines	50 labels	100 labels	400 labels	50 labels	100 labels	50 labels
MTC	4940 (10%)	4910 (10%)	2365 (5%)	4400 (10%)	4100 (10%)	4400 (5%)
OpenMatch	2470 (5%)	2455 (5%)	2365 (5%)	4400 (10%)	4100 (10%)	4400 (5%)
IOMatch	494 (1%)	491 (1%)	473 (1%)	6600 (15%)	6150 (15%)	22000 (25%)

TABLE 3: Models' training time of original OpenMatch and our proposed frameworks on top of OpenMatch. Experiments are conducted using CIFAR-100 with 100 labels on a single NVIDIA GeForce RTX 4090.

Algorithm	Training Time
OpenMatch	10h 59m
w/ WiseOpen †	43h 01m
w/ WiseOpen ‡	43h 05m
w/ WiseOpen-E †	13h 40m
w/ WiseOpen-E ‡	13h 43m
w/ WiseOpen-L †	11h 36m
w/ WiseOpen-L ‡	11h 43m



(a) Evaluation of ID classification.

(b) Evaluation of OOD detection.

Fig. 2: Performance of original OpenMatch and our proposed WiseOpen on top of OpenMatch. Experiments are conducted on CIFAR-100 with 100 labels per class. † means using Top-k threshold while ‡ means using Otsu threshold.

- Tiny-ImageNet [29], a subset of ImageNet [63] consisting of 200 classes, of which each class contains 500 and 100 images for training and testing respectively.

For all datasets, different from [18], [37], the first K classes are taken as seen classes while the rest of classes are taken as unseen classes, in order to construct a more complicated OSSL scenario followed by [21], [61]. Following [18], we conduct the experiments with different amounts of labeled training examples: 50, 100, or 400 labels per seen class for CIFAR-10, and 50 or 100 labels per seen class for CIFAR-100. As for Tiny-ImageNet, 50 training examples are taken as labeled data for each seen class. Additionally, 50 training examples per seen class are split as a validation set for all experiments. Except for the labeled set and validation set, the remaining training data are taken as unlabeled instances.

To present the way we construct the OSSL scenario more specifically, we take the OSSL scenario constructing procedure on CIFAR-10 as an example. In our experiments, we take the first 6 classes of CIFAR-10 as seen classes, namely airplane, automobile, bird, cat, deer, and dog, while the remaining 4 classes consisting of frog, horse, ship, and truck are taken as unseen classes. We randomly sample the images in the training set to construct a labeled set, an unlabeled set, and a validation set. Considering the setting of CIFAR-10 with 100 labels per class, we will get a labeled training set of 600 images with annotation in total which come from the seen classes, an unlabeled training set of 49,100 images where 29,100 images are from the seen classes while the rest 20,000 images are from the unseen classes, a validating set of 300 images from the seen classes, and a testing set of 10,000 images from the seen and unseen classes.

Baselines and Evaluation. There are three categories of the baseline methods: (1) Labeled Only method, which only trains the model with labeled data using cross-entropy

TABLE 4: Comparison of ID classification accuracy (in %, mean ± standard deviation) on CIFAR-10, CIFAR-100, and Tiny-ImageNet with varying labels per seen class. † means using Top-k threshold while ‡ means using Otsu threshold. Each block consists of the results of the baseline with or without the variants of WiseOpen and the improvements that our proposed frameworks can make. The best results for each data setting in each block are in bold.

	CIFAR-10			CIFAR-100		Tiny-ImageNet
Algorithms	50 labels	100 labels	400 labels	50 labels	100 labels	50 labels
Labeled Only	63.16±0.87	67.26±1.08	83.67±0.29	60.15±0.20	64.96±0.29	42.31±0.43
FixMatch [14]	90.48±0.01	**92.61±0.16**	93.68±0.46	70.16±0.48	**74.17±0.22**	**45.11±0.53**
w/ WiseOpen-E †	91.33±0.15	**92.61±0.32**	93.67±0.20	**70.53±0.48**	74.13±0.41	44.90±0.63
w/ WiseOpen-E ‡	**91.52±0.44**	92.54±0.11	**93.80±0.31**	69.86±0.16	74.12±0.38	44.98±0.57
Δ (mean)	+0.94	-0.03	+0.06	+0.04	-0.04	-0.17
Δ (max)	+1.04	0.00	+0.12	+0.38	-0.04	-0.13
FreeMatch [27]	85.43±0.64	88.44±0.36	89.47±0.29	65.38±0.80	70.20±0.27	42.16±0.84
w/ WiseOpen-E †	**86.09±2.00**	88.36±0.80	89.76±0.75	**65.68±0.52**	**70.32±0.40**	**42.74±0.06**
w/ WiseOpen-E ‡	85.71±0.33	**88.71±0.36**	**90.37±0.76**	65.52±0.47	70.13±0.12	42.49±0.55
Δ (mean)	+0.47	+0.09	+0.60	+0.22	+0.02	+0.46
Δ (max)	+0.66	+0.27	+0.90	+0.30	+0.12	+0.58
MTC [26]	79.00±1.73	80.51±1.67	89.03±0.93	64.22±0.61	70.22±0.57	39.57±0.17
w/ WiseOpen-E †	81.37±2.71	82.58±1.51	**89.73±0.34**	**64.71±0.28**	70.33±0.12	**40.49±0.48**
w/ WiseOpen-E ‡	82.57±0.40	82.17±1.08	89.27±0.45	64.54±0.48	**70.42±0.16**	39.80±0.11
w/ WiseOpen-L †	81.34±1.89	83.45±1.45	89.23±0.87	64.68±0.83	70.16±0.76	39.83±0.28
w/ WiseOpen-L ‡	**82.65±0.32**	**85.69±1.55**	89.54±0.49	64.39±0.40	70.34±0.26	38.99±0.38
Δ (mean)	+2.98	+2.97	+0.42	+0.36	+0.09	+0.21
Δ (max)	+3.65	+5.19	+0.70	+0.49	+0.20	+0.92
OpenMatch [18]	82.45±2.31	91.23±0.94	92.80±0.45	70.23±0.30	74.56±0.46	47.33±0.81
w/ WiseOpen-E †	83.69±1.59	**91.86±0.45**	93.11±0.50	70.93±0.66	75.14±0.33	49.45±0.31
w/ WiseOpen-E ‡	83.35±1.95	91.47±0.53	**93.23±0.34**	**71.67±0.38**	74.55±0.19	49.14±0.33
w/ WiseOpen-L †	83.45±0.95	91.82±0.37	93.12±0.27	71.23±0.59	**75.38±0.58**	**49.75±0.69**
w/ WiseOpen-L ‡	**84.69±0.76**	91.34±0.69	92.93±0.06	71.12±0.31	75.09±0.43	48.74±0.08
Δ (mean)	+1.34	+0.39	+0.30	+1.01	+0.48	+1.94
Δ (max)	+2.24	+0.63	+0.43	+1.44	+0.82	+2.42
IOMatch [37]	91.54±0.32	92.09±0.36	93.46±0.17	69.83±0.59	73.87±0.25	47.86±0.24
w/ WiseOpen-E †	91.78±0.17	**92.25±0.62**	**93.59±0.07**	**70.49±0.28**	74.24±0.41	47.93±0.19
w/ WiseOpen-E ‡	91.77±0.08	92.16±0.18	93.36±0.16	69.97±0.55	74.12±0.12	47.99±0.33
w/ WiseOpen-L †	**91.90±0.16**	**92.25±0.20**	93.56±0.25	70.26±0.55	**74.36±0.45**	48.49±0.40
w/ WiseOpen-L ‡	91.16±0.29	92.02±0.20	93.35±0.08	70.47±0.37	74.33±0.05	**49.18±0.40**
Δ (mean)	+0.11	+0.08	+0.00	+0.47	+0.40	+0.54
Δ (max)	+0.36	+0.16	+0.13	+0.67	+0.49	+1.32

loss; (2) traditional SSL methods, including FixMatch [14] and FreeMatch [27]; (3) OSSL methods, including MTC [26], OpenMacth [18], and IOMatch [37]. As for the evaluation of ID classification, we employ the top-1 accuracy over the testing instances from seen classes. Moreover, to evaluate OOD detection performance, the AUROC [64] over the whole testing set is adopted following [18], [40].

Implementation Details. For fairness, followed by [18], [26], [37], Wide ResNet-28-2 [65] is employed as the backbone representation extractor for all methods. For MTC, we apply their official implementation with Pytorch. For other methods, we utilize https://github.com/kekmodel/FixMatch-pytorch as their pipeline to conduct the experiments. The total number of training epochs for CIFAR-10/100 and Tiny-ImageNet are 512 and 256 respectively, and each epoch consists of 1024 iterations. For all experiments, the batch sizes for labeled data and unlabeled data are 64 and 128 respectively. We employ the SGD with a nesterov momentum of 0.9 as the optimizer, and the learning rate is initialized as 0.03 and decays in a cosine annealing manner. The hyper-parameter k utilized in Top-k thresholds for different scenarios is summarised in Table 2. All experiments can be performed on a single NVIDIA GeForce RTX 4090.

4.2 Main Results

As shown in Figure 2, we first conduct the experiments on top of OpenMatch on CIFRA-100 with 100 labels per seen class to validate the effectiveness of WiseOpen. We can observe that WiseOpen can make respectable performance improvements to the original OpenMatch model. However, as shown in Table 3, WiseOpen is relatively time-consuming on account of the frequent GV-SM that requires

TABLE 5: Comparison of AUROC (in %, mean ± standard deviation) for evaluating OOD detection performance. Higher is better.

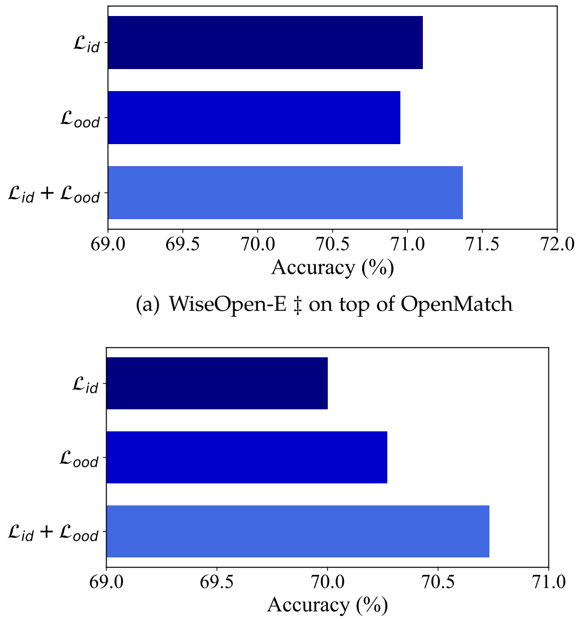
	CIFAR-10			CIFAR-100		Tiny-ImageNet
Algorithms	50 labels	100 labels	400 labels	50 labels	100 labels	50 labels
Labeled Only	56.15±1.81	59.58±2.25	69.95±0.60	67.86±0.71	70.04±0.52	61.49±0.38
FixMatch [14]	38.46±0.62	41.02±0.87	**48.49±1.41**	60.19±0.30	63.14±0.23	58.65±0.82
w/ WiseOpen-E †	**39.26±0.97**	41.20±1.56	47.53±0.44	**60.63±1.07**	62.57±0.12	**59.41±0.70**
w/ WiseOpen-E ‡	38.94±1.22	**42.26±1.10**	48.30±1.03	59.48±0.35	**63.28±0.52**	59.23±0.69
Δ (mean)	+0.64	+0.70	-0.58	-0.14	-0.22	+0.67
Δ (max)	+0.79	+1.23	-0.19	+0.44	+0.14	+0.76
FreeMatch [27]	45.94±1.84	**52.86±1.78**	**64.67±1.12**	**64.92±0.70**	68.71±0.19	**59.58±0.42**
w/ WiseOpen-E †	**47.38±2.03**	51.65±2.27	62.75±1.88	64.27±0.38	67.38±0.29	59.57±0.29
w/ WiseOpen-E ‡	46.46±2.36	52.79±2.37	63.94±2.67	64.22±0.89	**69.01±0.41**	58.84±0.86
Δ (mean)	+0.98	-0.64	-1.33	-0.68	-0.52	-0.38
Δ (max)	+1.44	-0.07	-0.73	-0.65	+0.30	-0.01
MTC [26]	77.77±1.10	80.36±2.13	87.02±0.91	65.40±0.60	64.58±0.26	60.71±0.55
w/ WiseOpen-E †	**79.02±0.35**	**82.02±1.98**	**88.67±0.76**	**66.78±1.79**	65.09±1.25	61.08±0.62
w/ WiseOpen-E ‡	78.10±0.36	79.08±1.74	86.59±2.45	65.97±0.91	64.41±1.25	61.29±0.08
w/ WiseOpen-L †	78.85±0.63	81.64±1.36	88.14±0.88	65.86±0.49	63.23±0.30	61.44±0.28
w/ WiseOpen-L ‡	78.45±0.39	81.65±0.70	86.10±1.71	65.41±1.42	**65.57±1.14**	**62.08±0.50**
Δ (mean)	+0.83	+0.73	+0.36	+0.60	-0.00	+0.76
Δ (max)	+1.25	+1.66	+1.65	+1.38	+0.99	+1.37
OpenMatch [18]	58.70±8.71	**55.60±5.09**	47.90±2.64	73.82±0.16	74.58±0.59	65.89±0.20
w/ WiseOpen-E †	**65.25±9.16**	49.97±5.71	**53.10±4.32**	75.23±0.49	76.12±0.72	66.41±0.15
w/ WiseOpen-E ‡	60.28±4.63	51.05±4.18	48.65±7.29	**75.24±0.34**	75.43±1.39	66.84±0.35
w/ WiseOpen-L †	55.33±4.99	49.70±4.45	44.14±3.55	74.59±0.40	**76.26±1.02**	66.71±0.57
w/ WiseOpen-L ‡	58.40±4.77	47.31±3.68	43.95±1.74	74.88±0.72	74.92±1.63	**67.33±0.21**
Δ (mean)	+1.11	-6.09	-0.44	+1.17	+1.10	+0.93
Δ (max)	+6.55	-4.55	+5.20	+1.42	+1.68	+1.44
IOMatch [37]	**44.54±0.29**	48.02±0.87	61.80±2.33	**67.44±0.88**	69.49±0.32	62.73±0.28
w/ WiseOpen-E †	42.51±2.22	48.34±0.96	61.49±2.17	65.99±0.14	69.36±0.48	**62.96±0.56**
w/ WiseOpen-E ‡	43.10±1.12	48.23±0.45	**63.45±1.04**	66.74±0.50	**69.67±0.10**	62.43±0.22
w/ WiseOpen-L †	44.23±1.74	**48.45±1.05**	60.58±1.18	67.04±0.43	69.43±0.53	62.66±0.60
w/ WiseOpen-L ‡	42.67±0.54	47.60±1.23	60.90±1.35	66.83±0.45	69.63±0.32	62.20±0.64
Δ (mean)	-1.41	+0.13	-0.20	-0.79	+0.03	-0.17
Δ (max)	-0.31	+0.43	+1.65	-0.40	+0.18	+0.23

the computation of gradient variance for each unlabeled instance in every epoch. Therefore, to extensively validate our proposed method, together with the consideration of the computational cost, we compare all SSL and OSSL baselines with WiseOpen's variants on top of their original algorithm. Note that FixMatch and FreeMatch can actually be seen as two variants of WiseOpen-L with their specific techniques as we mentioned in Section 3.5. Thus, for the SSL methods, we omit the WiseOpen-L experiments and only conduct WiseOpen-E experiments. The results are reported in Table 4 and Table 5, where the number of labels per seen class is provided for each column. In all settings, the first 60% of classes are taken as seen classes. From these results, we can obtain the following observations:

(1) Focusing on the ID classification performance among different methods, the Labeled Only method usually has the poorest performance, because of its no access to the open-set data. Being beneficial from the friendly data in the open-set data, traditional SSL methods and OSSL methods can achieve better performance in most cases, while our proposed frameworks can usually further enhance their performance since we exclude some unfriendly open-set data in the training procedure.

(2) Comparing the ID classification performance among our proposed WiseOpen and its two variants, we can observe that WiseOpen can make more sound improvements, particularly when the threshold is an adaptive threshold like the Otsu threshold. Meanwhile, WiseOpen-E and WiseOpen-L are also competent in improving the performance of the models in most scenarios with more acceptable extra training costs.

(3) By simultaneously considering the performance of ID classification and OOD detection, it appears that there is no necessary correlation between these two abilities. The algorithms with the best ID classification performance do not consistently exhibit excellent OOD



(a) WiseOpen-E ‡ on top of OpenMatch

(b) WiseOpen-E ‡ on top of IOMatch

Fig. 3: ID Classification performance of WiseOpen-E ‡ with different losses used in GV-SM on top of OpenMatch and IOMatch. Models are trained with CIFAR-100 at 50 labels.

detection performance and sometimes even perform poorly in distinguishing the ID instances from OOD instances. This phenomenon implicitly shows that the key point of handling the open-set data to enhance the ID classification performance lies in selecting and exploiting the friendly open-set data rather than simply detecting and discarding all OOD instances.

In summary, the main experiment results convincingly demonstrate the effectiveness of our proposed frameworks and correspond with our core idea that we should wisely leverage the open-set data to obtain valuable from the friendly open-set data while preventing being contaminated by the unfriendly ones.

4.3 Ablation Study

Comparison of GV-SM by Different Losses. OSSL methods with OOD detection modules typically incorporate two components in their unsupervised loss to optimize the modeling in the ID domain and the open-set domain respectively. Here we name the two components as $\mathcal{L}_{id}$ and $\mathcal{L}_{ood}$ respectively. For example, in OpenMatch, $\mathcal{L}_{id}$ and $\mathcal{L}_{ood}$ will be $\mathcal{L}_{fm}$ in Eq.6, and the sum of $\mathcal{L}_{em}$ in Eq.4 and $\mathcal{L}_{oc}$ in Eq.5 respectively. For IOMatch, $\mathcal{L}_{id}$ and $\mathcal{L}_{ood}$ will be the $\mathcal{L}_{ui}$ and $\mathcal{L}_{op}$, which are the cross-entropy loss for K-classification and $(K+1)$-classification by considering all unseen classes as a single novel class respectively. We perform the experiments of WiseOpen-E ‡ on top of OpenMatch and IOMatch with different losses used in GV-SM, using CIFAR-100 at 50 labels. As shown in Figure 3, we can observe that utilizing the whole unsupervised loss can promote the model to better performance compared with using single $\mathcal{L}_{id}$ or $\mathcal{L}_{ood}$, which is corresponding to our theoretical analysis.

TABLE 6: ID classification accuracy (in %) of WiseOpen-E and WiseOpen-L on top of OpenMatch and IOMatch in varying mismatching scenarios of CIFAR100 with 50 labels.

Mismatching ratios	0.4	0.5	0.6	0.7
OpenMatch	70.53	73.18	76.18	79.20
w/ WiseOpen-E †	71.85	73.68	77.40	78.93
w/ WiseOpen-E ‡	71.37	**74.06**	76.17	79.20
w/ WiseOpen-L †	**71.88**	73.76	**77.70**	**79.43**
w/ WiseOpen-L ‡	71.53	73.14	76.70	78.67
Δ (mean)	+1.13	+0.48	+0.81	-0.14
Δ (max)	+1.35	+0.88	+1.52	+0.23
IOMatch	70.53	72.74	74.98	77.63
w/ WiseOpen-E †	70.88	**73.20**	**75.52**	76.90
w/ WiseOpen-E ‡	70.73	73.04	75.35	77.20
w/ WiseOpen-L †	**71.02**	72.90	75.32	**78.27**
w/ WiseOpen-L ‡	70.93	73.02	75.07	78.13
Δ (mean)	+0.36	+0.30	+0.34	-0.00
Δ (max)	+0.49	+0.46	+0.54	+0.64

Sensitivity Analysis on Mismatching Ratio. To further demonstrate the robustness of our proposed frameworks, we conduct experiments on CIFAR-100 at 50 labels per class with varying class-mismatching ratios. Specifically, we evaluate the ID classification performance of WiseOpen-E and WiseOpen-L on top of OpenMatch and IOMacth along with the vanilla OpenMatch and IOMacth with varying values of $K \in \{60, 50, 40, 30\}$, *i.e.*, the mismatching ratio varies from 0.4 to 0.7. The experimental results are summarised in Table 6. We can observe that our proposed frameworks can enhance the ID classification capability in most cases, which demonstrates that our proposed frameworks are robust across different class-mismatching ratios.

Effects of Various Optimizers. As summarized in Table 7, we provide the ID classification performance of WiseOpen-E and WiseOpen-L on top of OpenMatch and IOMacth along with their vanilla versions, using the Adam optimizer and RMSProp optimizer. The initial learning rate is set as 0.0003 across all experiments and other hyperparameters remain the same as reported in Section 4.1. 50 labels per class are utilized in all datasets. As we can observe, our proposed methods successfully make improvements in most benchmarks but the performance of WiseOpen-E is not as stable as when using the SGD optimizer. This phenomenon is reasonable since our proposed GV-SM is built upon the theoretical analysis based on the stochastic gradient used in SGD.

4.4 Expansion on OOD Detection

Except for solving the ID classification problem, it's worth noting that models trained using most OSSL algorithms typically also develop the capability of OOD detection. In this subsection, we aim to explore the impact of our proposed frameworks on the models' capacity to detect various OOD instances. Specifically, we evaluate the OOD detection performance of our proposed frameworks on the OOD datasets that are not encountered during the training procedure. Following [66], we employ six benchmarks

TABLE 7: ID classification accuracy (in %) employing Adam optimizer and RMSProp optimizer. 50 labels per seen class are utilized in training models.

	CIFAR-10		CIFAR-100		Tiny-ImageNet	
Algorithms	Adam	RMSProp	Adam	RMSProp	Adam	RMSProp
OpenMatch	78.78	78.57	70.07	68.83	48.10	47.42
w/ WiseOpen-E †	73.10	76.88	69.77	69.52	47.98	**47.98**
w/ WiseOpen-E ‡	74.00	76.20	70.28	68.45	47.77	46.90
w/ WiseOpen-L †	**81.67**	78.93	71.12	68.40	48.18	47.13
w/ WiseOpen-L ‡	81.07	**82.40**	**71.47**	**69.97**	**48.35**	47.57
Δ (mean)	-1.32	+0.04	+0.59	+0.25	-0.03	-0.02
Δ (max)	+2.88	+3.83	+1.40	+1.13	+0.25	+0.57
IOMatch	90.27	91.78	70.60	70.32	45.35	45.28
w/ WiseOpen-E †	**91.37**	91.18	70.75	**70.73**	46.33	45.03
w/ WiseOpen-E ‡	90.55	91.83	**70.85**	70.53	46.13	44.63
w/ WiseOpen-L †	90.00	**92.32**	70.83	70.62	46.00	**45.30**
w/ WiseOpen-L ‡	89.75	91.05	70.68	70.70	**46.40**	45.23
Δ (mean)	+0.15	-0.19	+0.18	+0.33	+0.87	-0.23
Δ (max)	+1.10	+0.53	+0.25	+0.42	+1.05	+0.02

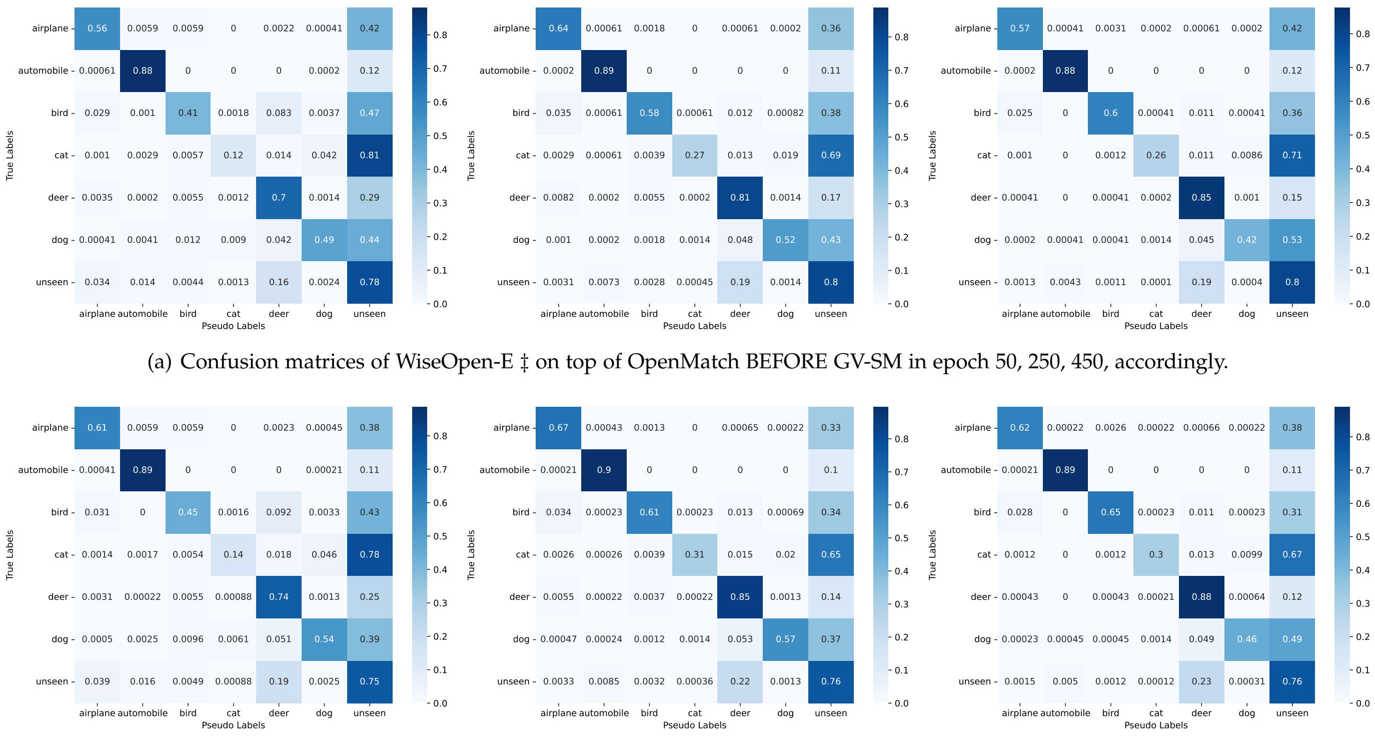
TABLE 8: Evaluation of OOD detection on OOD data unseen in the training set (AUROC in %). Models are trained on Tiny-ImageNet with 50 labeled data per class.

	Unseen OOD Datasets						
Algorithms	LSUN	DTD	CUB	Flowers	Caltech	Dogs	MEAN
MTC	37.51±1.40	35.25±2.60	47.91±3.48	52.28±2.79	47.49±2.93	40.24±4.99	43.45±6.94
w/ WiseOpen-E †	39.39±7.39	37.73±2.75	48.70±0.72	55.24±3.04	51.69±0.79	44.01±3.36	46.13±7.36
w/ WiseOpen-E ‡	41.10±11.44	37.17±1.31	48.38±4.21	49.60±3.29	50.00±3.49	36.76±1.44	43.84±7.85
w/ WiseOpen-L †	34.82±2.45	35.93±0.86	50.06±3.25	54.41±6.50	50.87±2.83	43.24±1.65	44.89±8.25
w/ WiseOpen-L ‡	**45.43±15.31**	**47.81±2.07**	**58.03±4.75**	**60.51±2.96**	**57.26±4.41**	**48.85±1.24**	**52.98±9.06**
Δ (mean)	+2.68	+4.41	+3.38	+2.66	+4.96	+2.97	+3.51
Δ (max)	+7.92	+12.56	+10.11	+8.23	+9.77	+8.61	+9.53
OpenMatch	53.06±2.72	46.84±0.20	57.04±0.15	55.88±1.41	60.00±0.92	**61.13±0.67**	55.66±4.93
w/ WiseOpen-E †	54.38±1.24	47.68±1.38	58.45±1.14	55.72±2.30	61.84±1.06	59.77±1.34	56.31±4.81
w/ WiseOpen-E ‡	**56.41±0.98**	49.20±2.01	59.57±1.62	57.02±2.02	62.43±0.67	59.61±1.31	57.37±4.42
w/ WiseOpen-L †	56.22±0.78	**49.60±0.88**	59.39±0.20	59.62±1.17	**62.97±0.61**	59.81±1.46	**57.93±4.31**
w/ WiseOpen-L ‡	56.01±2.64	49.54±3.20	**59.97±0.14**	**59.81±1.91**	62.10±1.23	58.89±1.53	57.72±4.56
Δ (mean)	+2.70	+2.16	+2.30	+2.16	+2.33	-1.61	+1.68
Δ (max)	+3.35	+2.76	+2.93	+3.93	+2.97	-1.32	+2.28
IOMatch	63.88±1.76	56.53±2.11	63.10±2.07	64.52±4.22	63.71±1.15	**63.71±0.53**	62.57±3.56
w/ WiseOpen-E †	**67.28±0.59**	57.46±1.65	65.14±1.54	67.37±0.38	62.87±0.34	62.77±1.48	63.81±3.57
w/ WiseOpen-E ‡	65.47±1.67	**59.75±0.60**	64.23±1.03	66.29±3.21	63.31±0.64	62.98±1.39	63.67±2.68
w/ WiseOpen-L †	66.16±3.31	57.07±2.41	**66.56±0.84**	67.04±2.40	**63.73±1.03**	63.41±0.98	**63.99±3.96**
w/ WiseOpen-L ‡	64.93±2.04	57.93±2.35	64.18±1.56	**68.85±0.69**	63.65±0.42	62.93±1.09	63.75±3.56
Δ (mean)	+2.08	+1.52	+1.93	+2.86	-0.32	-0.69	+1.23
Δ (max)	+3.40	+3.22	+3.47	+4.33	+0.02	-0.30	+1.42

as OOD datasets unseen in training, namely, LSUN [67], DTD [68], CUB-200 [69], Flowers [70], Caltech [71], and Dogs [72], for models trained on Tiny-ImageNet with 50 labels per seen class. AUROC is employed to evaluate the detection performance. As summarized in Table 8, we can observe that our proposed frameworks can promote the OOD detection performance of the majority of the unseen OOD instances. For instance, in terms of the mean OOD detection performance across all unseen datasets, our frameworks demonstrate an average improvement of 3.51% and a maximum improvement of 9.53% for MTC. This suggests that our proposal is sufficiently safe and will not compromise the potential OOD detection capability of the original model; in fact, it may even enhance it.

TABLE 9: Comparison of ID classification accuracy (in %) among IOMatch (reduced), the proposed WiseOpen variants on top of IOMatch (reduced), and full IOMatch. 50 labels per seen class are utilized in training models.

Datasets	CIFAR-10	CIFAR-100	Tiny-ImageNet
IOMatch	91.90	70.53	47.68
IOMatch (reduced)	91.47	69.97	46.75
w/ WiseOpen-E †	91.40	70.85	47.13
w/ WiseOpen-E ‡	91.18	70.67	47.00
w/ WiseOpen-L †	91.55	70.53	46.83
w/ WiseOpen-L ‡	91.40	70.47	47.37



(a) Confusion matrices of WiseOpen-E ‡ on top of OpenMatch BEFORE GV-SM in epoch 50, 250, 450, accordingly.

(b) Confusion matrices of WiseOpen-E ‡ on top of OpenMatch AFTER GV-SM in epoch 50, 250, 450, accordingly.

Fig. 4: Visualization of the confusion matrices on the unlabeled training set of CIFAR-10.

4.5 Further Analysis

Comparison between Existing Selection Mechanisms. Existing OSSL methods typically apply some OOD detection techniques as selection mechanisms tailored for one specific loss. Previous experimental results in Table 4 have successfully demonstrated that our proposed selection mechanism is orthogonal and complementary to prior selection mechanisms that can improve previous algorithms' performance after effortlessly embedding our proposal. In consideration of the completeness, we further perform the experiments of IOMatch (reduced), which omits the OOD data exclusion process while calculating the ID classification loss $\mathcal{L}_{ui}$. As summarized in Table 9, we can observe that our proposed selection mechanism can improve the ID classification performance of IOMatch (reduced) in most cases. However, IOMatch (reduced) plus our selection mechanisms do not always outperform the full IOMatch. The reason may lie in that the OOD data exclusion is specifically tailored for the unlabeled inlier loss and the omission of such exclusion can potentially introduce biases to ID classification by misclassifying OOD data to seen classes. Our selection mechanism can alleviate the influence and improve the model's performance by selectively leveraging friendly data. However, in some datasets, the negative impacts of discarding OOD detection may be greater than the positive effect brought by our selection mechanism. Therefore, in some instances, the OOD data exclusion may appear to be more effective, while in others, our proposed selection mechanism may lead to more significant improvements. Overall, these empirical results show that although not tailored for the specific loss, our proposal is effective and robust, and can further enhance the model's capability which already utilizes the tailored selection in the training procedure.

Analysis of GV-SM. To further analyse the effectiveness of our proposal, Figure 4 visualizes the confusion matrix of the unlabeled training set before and after applying GV-SM. We can observe that after applying GV-SM, the model can utilize the open-set data set with higher pseudo-labeling accuracy on the seen classes. Meanwhile, the mismatching from unseen classes to the seen class *deer* slightly increases. Upon further investigation, we have discovered that among the unseen classes frog, horse, ship, and truck, the mismatch from horse to deer contributes to the overwhelming majority of mismatches from unseen classes to deer with ratios reaching 98.41%, 99.27%, and 99.81% for epochs 50, 250, and 450, respectively. Intuitively, this type of mismatch is unlikely to negatively impact the distinction between deer and other seen classes, and it may even enhance this distinction by facilitating the learning of similar representations between horses and deer, which are distinct from other seen classes.

5 Conclusion

In this paper, we tackled the realistic OSSL scenario, where models can suffer from performance degradation if recklessly utilize all open-set data which contain instances from unseen classes. Motivated by theoretical insights, we highlight the significance of selectively leveraging open-set data and propose a novel OSSL framework called WiseOpen which wisely leverages the open-set data by using GV-SM. GV-SM enables the model to exclude potentially unfriendly open-set training data with large gradient variance, thereby helping to maintain the purity of the learned knowledge. Furthermore, we also proposed two variants of WiseOpen dubbed WiseOpen-E and WiseOpen-L to mitigate the huge computing cost issue. Sufficient empirical results have demonstrated the effectiveness of our proposal.

Acknowledgments

This work was supported by National Key RD Program of China (2022YFF0712100), NSFC (62276131, 61921006, 08120002), the Fundamental Research Funds for the Central Universities (No.30922010317), the Fundamental Research Funds for the Central University of China (DUT No. 82232031), and Collaborative Innovation Center of Novel Software Technology and Industrialization.

References

[1] O. Chapelle, B. Schölkopf, and A. Zien, Eds., *Semi-Supervised Learning*. The MIT Press, 2006.

[2] X. Zhu and A. B. Goldberg, "Introduction to semi-supervised learning," *Synth. Lect. Artif. Intell. Mach. Learn.*, vol. 3, no. 1, pp. 1–130, 2009.

[3] Y.-F. Li, L.-Z. Guo, and Z.-H. Zhou, "Towards safe weakly supervised learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 1, pp. 334–346, Jun. 2019.

[4] Z.-H. Zhou, "A brief introduction to weakly supervised learning," *Nat. Sci. Rev.*, vol. 5, no. 1, pp. 44–53, 2018.

[5] Y. Yang, Z.-Y. Fu, D.-C. Zhan, Z.-B. Liu, and Y. Jiang, "Semi-supervised multi-modal multi-instance multi-label deep network with optimal transport," *IEEE Trans. Knowl. Data Eng.*, vol. 33, no. 2, pp. 696–709, Feb. 2021.

[6] Y. Yang, D.-W. Zhou, D.-C. Zhan, H. Xiong, Y. Jiang, and J. Yang, "Cost-effective incremental deep model: Matching model capacity with the least sampling," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 4, pp. 3575–3588, Apr. 2023.

[7] S. Laine and T. Aila, "Temporal ensembling for semi-supervised learning," in *Proc. Int. Conf. Learn. Representations*, 2017.

[8] 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.

[9] 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.

[10] Y. Grandvalet and Y. Bengio, "Semi-supervised learning by entropy minimization," in *Proc. Adv. Neural Inf. Process. Syst.*, 2004, pp. 529–536.

[11] D.-H. Lee *et al.*, "Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks," in *Proc. Int. Conf. Mach. Learn.*, vol. 3, no. 2, 2013, p. 896.

[12] D. Berthelot, N. Carlini, I. J. Goodfellow, N. Papernot, A. Oliver, and C. Raffel, "Mixmatch: A holistic approach to semi-supervised learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 5050–5060.

[13] D. Berthelot, N. Carlini, E. D. Cubuk, A. Kurakin, K. Sohn, H. Zhang, and C. Raffel, "Remixmatch: Semi-supervised learning with distribution alignment and augmentation anchoring," *CoRR*, vol. abs/1911.09785, 2019.

[14] K. Sohn, D. Berthelot, N. Carlini, Z. Zhang, H. Zhang, C. Raffel, E. D. Cubuk, A. Kurakin, and C. Li, "Fixmatch: Simplifying semi-supervised learning with consistency and confidence," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 596–608.

[15] Y. Xu, L. Shang, J. Ye, Q. Qian, Y. Li, B. Sun, H. Li, and R. Jin, "Dash: Semi-supervised learning with dynamic thresholding," in *Proc. Int. Conf. Mach. Learn.*, vol. 139, 2021, pp. 11 525–11 536.

[16] Y. Yang, H. Wei, Z.-Q. Sun, G.-Y. Li, Y. Zhou, H. Xiong, and J. Yang, "S2osc: A holistic semi-supervised approach for open set classification," *ACM Trans. Knowl. Discov. Data*, vol. 16, no. 2, pp. 34:1–34:27, Sep. 2021.

[17] Z. Zhao, L. Zhou, L. Wang, Y. Shi, and Y. Gao, "Lassl: label-guided self-training for semi-supervised learning," in *Proc. AAAI Conf. Artif. Intell.*, vol. 36, no. 8, 2022, pp. 9208–9216.

[18] K. Saito, D. Kim, and K. Saenko, "Openmatch: Open-set consistency regularization for semi-supervised learning with outliers," in *Proc. Adv. Neural Inf. Process. Syst.*, 2021, pp. 25 956–25 967.

[19] Y. Yang, H. Wei, H. Zhu, D. Yu, H. Xiong, and J. Yang, "Exploiting cross-modal prediction and relation consistency for semisupervised image captioning," *IEEE Trans. Cybern.*, vol. 54, no. 2, pp. 890–902, Feb. 2024.

[20] Y. Yang, D.-C. Zhan, Y.-F. Wu, Z.-B. Liu, H. Xiong, and Y. Jiang, "Semi-supervised multi-modal clustering and classification with incomplete modalities," *IEEE Trans. Knowl. Data Eng.*, vol. 33, no. 2, pp. 682–695, Feb. 2021.

[21] L.-Z. Guo, Y.-G. Zhang, Z.-F. Wu, J.-J. Shao, and Y.-F. Li, "Robust semi-supervised learning when not all classes have labels," in *Proc. Adv. Neural Inf. Process. Syst.*, 2022, pp. 3305–3317.

[22] W. Xi, X. Song, W. Guo, and Y. Yang, "Robust semi-supervised learning for self-learning open-world classes," in *Proc. IEEE Int. Conf. Data Min.*, 2023, pp. 658–667.

[23] Y. Yang, Y. Zhang, X. SONG, and Y. Xu, "Not all out-of-distribution data are harmful to open-set active learning," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 36, 2023, pp. 13 802–13 818.

[24] Y. Chen, X. Zhu, W. Li, and S. Gong, "Semi-supervised learning under class distribution mismatch," in *Proc. AAAI Conf. Artif. Intell.*, vol. 34, no. 04, 2020, pp. 3569–3576.

[25] L.-Z. Guo, Z.-Y. Zhang, Y. Jiang, Y.-F. Li, and Z.-H. Zhou, "Safe deep semi-supervised learning for unseen-class unlabeled data," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 3897–3906.

[26] Q. Yu, D. Ikami, G. Irie, and K. Aizawa, "Multi-task curriculum framework for open-set semi-supervised learning," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 438–454.

[27] Y. Wang, H. Chen, Q. Heng, W. Hou, Y. Fan, Z. Wu, J. Wang, M. Savvides, T. Shinozaki, B. Raj, B. Schiele, and X. Xie, "Freematch: Self-adaptive thresholding for semi-supervised learning," in *Proc. Int. Conf. Learn. Representations*, 2023.

[28] A. Krizhevsky, G. Hinton *et al.*, "Learning multiple layers of features from tiny images," *Tech. Rep.*, 2009.

[29] L. Yao and J. Miller, "Tiny imagenet classification with convolutional neural networks," *CS 231N*, vol. 2, no. 5, p. 8, 2015.

[30] Y. LeCun, Y. Bengio, and G. E. Hinton, "Deep learning," *Nat.*, vol. 521, no. 7553, pp. 436–444, 2015.

[31] Q. Xie, Z. Dai, E. H. Hovy, T. Luong, and Q. Le, "Unsupervised data augmentation for consistency training," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 6256–6268.

[32] B. Zhang, Y. Wang, W. Hou, H. Wu, J. Wang, M. Okumura, and T. Shinozaki, "Flexmatch: Boosting semi-supervised learning with

curriculum pseudo labeling," in *Proc. Adv. Neural Inf. Process. Syst.*, 2021, pp. 18 408–18 419.

[33] V. Nair, J. F. Alonso, and T. Beltramelli, "Realmix: Towards realistic semi-supervised deep learning algorithms," *CoRR*, vol. abs/1912.08766, 2019.

[34] J. Huang, C. Fang, W. Chen, Z. Chai, X. Wei, P. Wei, L. Lin, and G. Li, "Trash to treasure: Harvesting OOD data with cross-modal matching for open-set semi-supervised learning," in *Proc. IEEE Int. Conf. Comput. Vis*, 2021, pp. 8290–8299.

[35] J. Park, S. Yun, J. Jeong, and J. Shin, "Opencos: Contrastive semi-supervised learning for handling open-set unlabeled data," in *Proc. Eur. Conf. Comput. Vis.*, vol. 13802, 2022, pp. 134–149.

[36] Z. Huang, J. Yang, and C. Gong, "They are not completely useless: Towards recycling transferable unlabeled data for class-mismatched semi-supervised learning," *IEEE Trans. Multim.*, vol. 25, pp. 1844–1857, Jun. 2023.

[37] Z. Li, L. Qi, Y. Shi, and Y. Gao, "Iomatch: Simplifying open-set semi-supervised learning with joint inliers and outliers utilization," in *Proc. IEEE Int. Conf. Comput. Vis*, 2023, pp. 15 870–15 879.

[38] J. Yang, P. Wang, D. Zou, Z. Zhou, K. Ding, W. Peng, H. Wang, G. Chen, B. Li, Y. Sun, X. Du, K. Zhou, W. Zhang, D. Hendrycks, Y. Li, and Z. Liu, "Openood: Benchmarking generalized out-of-distribution detection," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 35, 2022, pp. 32 598–32 611.

[39] X. Du, Z. Wang, M. Cai, and Y. Li, "VOS: learning what you don't know by virtual outlier synthesis," in *Proc. Int. Conf. Learn. Representations*, 2022.

[40] D. Hendrycks and K. Gimpel, "A baseline for detecting misclassified and out-of-distribution examples in neural networks," in *Proc. Int. Conf. Learn. Representations*, 2017.

[41] S. Padhy, Z. Nado, J. Ren, J. Z. Liu, J. Snoek, and B. Lakshminarayanan, "Revisiting one-vs-all classifiers for predictive uncertainty and out-of distribution detection in neural networks," *CoRR*, vol. abs/2007.05134, 2020.

[42] Y. Sun, Y. Ming, X. Zhu, and Y. Li, "Out-of-distribution detection with deep nearest neighbors," in *Proc. Int. Conf. Mach. Learn.*, vol. 162, 2022, pp. 20 827–20 840.

[43] H. Wei, R. Xie, H. Cheng, L. Feng, B. An, and Y. Li, "Mitigating neural network overconfidence with logit normalization," in *Proc. Int. Conf. Mach. Learn.*, vol. 162, 2022, pp. 23 631–23 644.

[44] Y. Yang, Z.-Q. Sun, H. Zhu, Y. Fu, Y. Zhou, H. Xiong, and J. Yang, "Learning adaptive embedding considering incremental class," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 3, pp. 2736–2749, Mar. 2023.

[45] D. Hendrycks, M. Mazeika, and T. G. Dietterich, "Deep anomaly detection with outlier exposure," in *Proc. Int. Conf. Learn. Representations*, 2019.

[46] J. Yang, H. Wang, L. Feng, X. Yan, H. Zheng, W. Zhang, and Z. Liu, "Semantically coherent out-of-distribution detection," in *Proc. IEEE Int. Conf. Comput. Vis*, 2021, pp. 8281–8289.

[47] Q. Yu and K. Aizawa, "Unsupervised out-of-distribution detection by maximum classifier discrepancy," in *Proc. IEEE Int. Conf. Comput. Vis*, 2019, pp. 9517–9525.

[48] E. D. Cubuk, B. Zoph, J. Shlens, and Q. V. Le, "Randaugment: Practical automated data augmentation with a reduced search space," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2020, pp. 3008–3017.

[49] S. Ghadimi and G. Lan, "Stochastic first-and zeroth-order methods for nonconvex stochastic programming," *SIAM J. Optim.*, vol. 23, no. 4, pp. 2341–2368, 2013.

[50] D. P. Bertsekas and J. N. Tsitsiklis, "Neuro-dynamic programming: an overview," in *Proc. IEEE Conf. Decis. Control*, vol. 1, 1995, pp. 560–564.

[51] L. Bottou, F. E. Curtis, and J. Nocedal, "Optimization methods for large-scale machine learning," *SIAM Rev Soc Ind Appl Math*, vol. 60, no. 2, pp. 223–311, 2018.

[52] Y. Nesterov, *Introductory lectures on convex optimization : a basic course*, ser. Applied optimization. Kluwer Academic Publ., 2004.

[53] B. T. Polyak, "Gradient methods for minimizing functionals," *Zhurnal Vychislitel'noi Matematiki i Matematicheskoi Fiziki*, vol. 3, no. 4, pp. 643–653, 1963.

[54] Z. Allen-Zhu, Y. Li, and Z. Song, "A convergence theory for deep learning via over-parameterization," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 242–252.

[55] Z. Yuan, Y. Yan, R. Jin, and T. Yang, "Stagewise training accelerates convergence of testing error over SGD," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 2604–2614.

[56] H. Karimi, J. Nutini, and M. Schmidt, "Linear convergence of gradient and proximal-gradient methods under the polyak-łojasiewicz condition," in *Lect. Notes Comput. Sci.*, vol. 9851, 2016, pp. 795–811.

[57] X. Li, Z. Zhuang, and F. Orabona, "Exponential step sizes for non-convex optimization," *CoRR*, vol. abs/2002.05273, 2020.

[58] Z. Charles and D. S. Papailiopoulos, "Stability and generalization of learning algorithms that converge to global optima," in *Proc. Int. Conf. Mach. Learn.*, vol. 80, 2018, pp. 744–753.

[59] Z. Li and J. Li, "A simple proximal stochastic gradient method for nonsmooth nonconvex optimization," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 5569–5579.

[60] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Trans. Syst. Man Cybern.*, vol. 9, no. 1, pp. 62–66, Jan. 1979.

[61] K. Cao, M. Brbic, and J. Leskovec, "Open-world semi-supervised learning," in *Proc. Int. Conf. Learn. Representations*, 2022.

[62] L. Han, H. Ye, and D. Zhan, "On pseudo-labeling for class-mismatch semi-supervised learning," *CoRR*, vol. abs/2301.06010, 2023.

[63] J. Deng, W. Dong, R. Socher, L. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2009, pp. 248–255.

[64] J. Davis and M. Goadrich, "The relationship between precision-recall and ROC curves," in *Proc. Int. Conf. Mach. Learn.*, W. W. Cohen and A. W. Moore, Eds., vol. 148, 2006, pp. 233–240.

[65] S. Zagoruyko and N. Komodakis, "Wide residual networks," in *Proc. Br. Mach. Vis. Conf.*, 2016.

[66] J. Tack, S. Mo, J. Jeong, and J. Shin, "CSI: novelty detection via contrastive learning on distributionally shifted instances," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 11 839–11 852.

[67] F. Yu, Y. Zhang, S. Song, A. Seff, and J. Xiao, "LSUN: construction of a large-scale image dataset using deep learning with humans in the loop," *CoRR*, vol. abs/1506.03365, 2015.

[68] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, and A. Vedaldi, "Describing textures in the wild," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2014, pp. 3606–3613.

[69] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, "The caltech-ucsd birds-200-2011 dataset," *Tech. Rep.*, 2011.

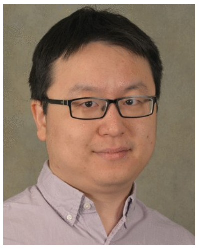
[70] M. Nilsback and A. Zisserman, "A visual vocabulary for flower classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, vol. 2, 2006, pp. 1447–1454.

[71] G. Griffin, A. Holub, and P. Perona, "Caltech-256 object category dataset," *Tech. Rep.*, 2007.

[72] A. Khosla, N. Jayadevaprakash, B. Yao, and F.-F. Li, "Novel dataset for fine-grained image categorization: Stanford dogs," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, vol. 2, no. 1, 2011.

Yang Yang received the Ph.D. degree in computer science, Nanjing University, China in 2019. At the same year, he became a faculty member at Nanjing University of Science and Technology, China. He is currently a Professor with the school of Computer Science and Engineering. His research interests lie primarily in machine learning and data mining, including heterogeneous learning, model reuse, and incremental mining. He has published prolifically in refereed journals and conference proceedings, including IEEE Transactions on Knowledge and Data Engineering (TKDE), ACM Transactions on Information Systems (ACM TOIS), ACM Transactions on Knowledge Discovery from Data (TKDD), ACM SIGKDD, ACM SIGIR, WWW, IJCAI, and AAAI. He was the recipient of the the Best Paper Award of ACML-2017. He serves as PC in leading conferences such as IJCAI, AAAI, ICML, NeurIPS, etc.

Nan Jiang is currently working towards the M.S. degree with the National Key Laboratory for Novel Software Technology, School of Artificial Intelligence, Nanjing University, China. His research interests lie primarily in machine learning and data mining. He is currently working on semi-supervised learning.



Yi Xu received the PhD degree in computer science from the University of Iowa, Iowa City, Iowa USA, in 2019. He is currently a professor with the School of Control Science and Engineering, Dalian University of Technology, China. His research interests are machine learning, optimization, deep learning, and statistical learning theory. He has published more than twenty papers in refereed journals and conference proceedings, including IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), Transactions on Machine Learning Research, NeurIPS, ICML, ICLR, CVPR, AAAI, IJCAI, UAI. He serves as a SPC/PC/Reviewer in leading conferences such as NeurIPS, ICML, ICLR, CVPR, AAAI, IJCAI, etc.

De-Chuan Zhan received the PhD degree in computer science, Nanjing University, China in 2010, then he became a faculty member with the Department of Computer Science and Technology at Nanjing University, China. He is currently a professor with the School of Artificial Intelligence at Nanjing University. His research interests are mainly in machine learning, data mining, and mobile intelligence. He has published more than 90 papers in leading international journal/conferences. He serves as an editorial board member of IDA and IJAPR, and serves as SPC/PC in leading conferences, such as IJCAI, AAAI, ICML, NeurIPS, etc.