

# PROMPT-DRIVEN EFFICIENT OPEN-SET SEMI-SUPERVISED LEARNING

**Anonymous authors**

Paper under double-blind review

## ABSTRACT

Open-set semi-supervised learning (OSSL) has attracted growing interest, which investigates a more practical scenario where out-of-distribution (OOD) samples are only contained in unlabeled data. Existing OSSL methods like OpenMatch learn an OOD detector to identify outliers, which often update all modal parameters (*i.e.*, full fine-tuning) to propagate class information from labeled data to unlabeled ones. In this work, we suggest using a visual prompt-driven mechanism to obtain higher computational efficiency in the OSSL task. To this end, we propose a prompt-driven efficient OSSL framework, called *OpenPrompt*, which can propagate class information from labeled to unlabeled data with only a small number of trainable parameters. We propose a prompt-driven joint space learning mechanism to detect OOD data by maximizing the distribution gap between ID and OOD samples in unlabeled data, thereby our method enables the outliers to be detected in a new way. The experimental results on three public datasets show that *OpenPrompt* outperforms state-of-the-art methods with less than 1% of trainable parameters. More importantly, *OpenPrompt* achieves a 4% improvement in terms of AUROC on outlier detection over a fully supervised model on CIFAR10.

## 1 INTRODUCTION

The goal of semi-supervised learning (SSL) is to improve a model’s performance by leveraging unlabeled data Sohn et al. (2020). It can significantly improve recognition accuracy by propagating the class information from a small set of labeled data to a large set of unlabeled data without additional annotation cost Li et al. (2021); Wang et al. (2021). Existing SSL methods are built on the assumption that labeled and unlabeled data share the same distribution space. However, due to how it was collected, unlabeled data may contain new categories, such as outliers, that are never seen in the labeled data Saito et al. (2021), resulting in lower SSL performance. To address this issue, Open Set SSL (OSSL) is proposed, the task of which is to classify in-distribution (ID) samples into the correct class while identifying out-of-distribution (OOD) samples as outliers Yu et al. (2020).

Typical OSSL methods like MTC Yu et al. (2020) use a joint optimization framework to update the network parameters and the OOD score alternately. OpenMatch Saito et al. (2021) uses an OVA network that can learn a threshold to distinguish OOD samples from ID samples. Both of them follow the training strategy of current SSL methods, which **1)** propagate class information from a small set of labeled data to a large set of unlabeled data by fine-tuning all model parameters. Then use **2)** an additional structured OOD detector to identify outliers unseen in the labeled data that the unlabeled data may contain. However, such a mechanism leads to expensive computational costs.

As a new paradigm, prompting has shown outstanding effects in NLP Radford et al. (2021); Shin et al. (2020); Jiang et al. (2020), which can make the model directly applicable to downstream tasks without introducing new parameters. Recently, prompting has been applied to computer vision tasks Jia et al. (2022); Bahng et al. (2022); Sung et al. (2022), which can greatly reduce the number of trainable parameters by modifying a small number of pixels to guide frozen vision models to solve new tasks. However, when the new task has OOD samples that have never appeared in the labeled data without supervision, the effectiveness of the prompt remains to be verified.

To this end, our study starts with the visual prompt-based OSSL task, where we hope to achieve **1)** class information propagation and **2)** OOD detection only by modifying a few pixels. First, we propose a prompt-driven efficient OSSL framework (*OpenPrompt*), which can find the outliers from

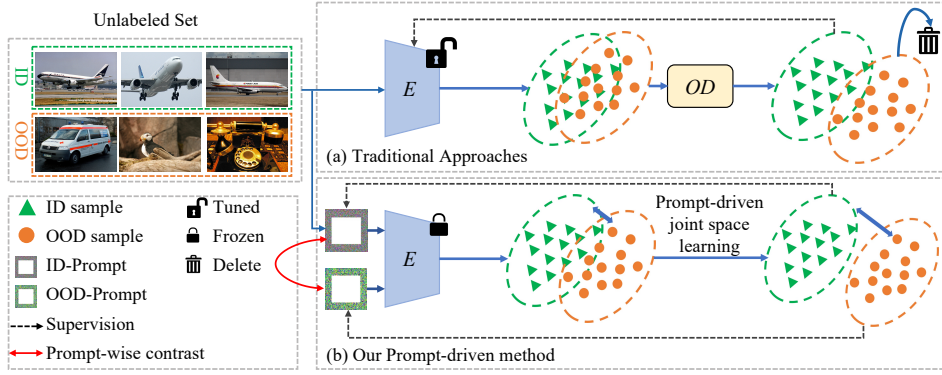


Figure 1: **Comparison of (a) traditional OSSL approaches and (b) our prompt-driven efficient OSSL**, where  $E$  represents visual-encoder and  $OD$  denotes outlier detector. (b) achieves class information propagation and OOD detection through only a small number of learnable parameters with a joint space learning mechanism, while (a) needs to fully fine-tune the whole model and an additional OOD detector. Different from (a) which discards OOD samples after detection, (b) reuses them to train an OOD-specific prompt to push the OOD samples away from the ID ones.

unlabeled data with a small number of trainable parameters. Then, to find outliers without supervision, we project the representations of all samples into a prompt-driven joint space, which enlarges the distribution gap between ID and OOD samples (see Fig. 1). Instead of directly discarding the detected OOD samples (see Fig. 1 (a)), we make use of the detected OOD samples and feed them into the network to construct OOD-specific prompts. With prompts trained through ID data serving as positive samples, the OOD-specific prompts are regarded as negative samples to push OOD data away from ID samples (see Fig. 1 (b)). Such a prompt-wise contrastive representation can further shape the joint space by exploring the structural information of labeled ID and OOD samples.

Overall, our contributions can be summarized as follows: • We propose a novel and efficient prompt-driven OSSL framework (termed *OpenPrompt*), which can only update a small number of learnable parameters to match the performance of full fine-tuning methods. • We develop a prompt-driven joint space learning strategy to enlarge the distribution gap between ID and OOD data by visual prompts without supervision. • We utilize a prompt-wise contrastive representation, *i.e.*, ID-specific and OOD-specific prompt, to further shape the joint space instead of discarding OOD samples after detection. • Experimental results show that *OpenPrompt* performs better than other SOTA methods with no more than 1% trainable parameters. Besides, *OpenPrompt* achieves the best performance in detecting outliers compared with other methods.

## 2 RELATED WORK

**Semi-supervised learning (SSL).** The SSL methods assume the labeled and unlabeled data are from the same classes and propagate class information from labeled to unlabeled data to improve a model’s performance Rasmus et al. (2015); Berthelot et al. (2019); Verma et al. (2022); Zheng et al. (2022); French et al. (2019). Existing pseudo-labeling Lee et al. (2013) and consistency regularization Laine & Aila (2017) based SSL approaches show great performance on many benchmark datasets. They usually set a selection threshold to make sure all pseudo-labels used are reliable. Different from previous methods, U<sup>2</sup>PL Wang et al. (2022) treats unreliable pseudo-labels as negative samples for contrastive learning. Several methods DeVries & Taylor (2017); Yang et al. (2022); Yun et al. (2019); Yuan et al. (2021) utilize different data augmentations with self-training to maintain the consistency of the model. All these existing works often propagate class information from labeled to unlabeled data using a full fine-tuning adaption, however, it will require large computing resources by adapting the whole model to the unlabeled data.

**Open-set Semi-supervised learning (OSSL).** OSSL relaxes the SSL assumption by presuming that unlabeled data contains instances from novel classes, which will lower the performance of SSL approaches. Existing OSSL approaches use outlier detectors to recognize and filter those instances to improve the model’s robustness. MTC introduces a joint optimization framework, which updates the

network and the discriminating score of unlabeled data alternately Yu et al. (2020). UASD generates soft targets as regularisers to empower the robustness of the proposed SSL network Chen et al. (2020). OpenMatch applies a soft consistency loss to the outlier detector by outputting the confidence score of a sample to detect outliers Saito et al. (2021). By contrast, our *OpenPrompt* uses visual prompts to distinguish the representation gaps between ID and OOD samples and applies prompt-driven joint space learning to enlarge these gaps for outlier detection. Experimental results demonstrate that our approach achieves great performance through this prompt-driven joint space learning mechanism.

**Prompting.** Prompting is a *modus operandi* method, which aims to adapt pre-trained models to downstream tasks by modifying the data space Radford et al. (2021). With manually chosen prompt candidates, GPT-3 performs well on downstream transfer learning tasks Brown et al. (2020). Recent studies in computer vision treat prompt as task-specific vectors and by tuning prompts instead of the models’ parameters, the pre-trained models could also be adapted to downstream tasks with the same performance. VPT injects learnable prompt tokens to the vision transformer and keeps the backbone frozen during the downstream fine-tuning stage, which achieves great performance with only a small amount of learnable parameters Jia et al. (2022). Bahng *et al.* applies learnable paddings to the data space as a visual prompt which uses fewer parameters to realize large-scale model adaption Bahng et al. (2022). Inspired by those recent studies on vision prompts, we proposed an effective prompt-based OSSSL framework. Different from existing approaches, we apply a prompt for a robust adaption by rejecting noisy samples during the downstream fine-tuning stage.

### 3 METHODS

**Problem setting.** Our task is to train an effective classification network via OSSSL. In OSSSL, we divide the used dataset into two parts: the labeled data  $\mathcal{X}_l = \{(x_i, y_i)\}_{i=1}^n$  and unlabeled data  $\mathcal{X}_u = \{(x_i)\}_{i=1}^m$ , and the class spaces of labeled and unlabeled data are denoted as  $\mathcal{O}_l$  and  $\mathcal{O}_u$ , respectively. In this study, we assume that  $\mathcal{O}_l \subset \mathcal{O}_u$  and  $\mathcal{O}_l \neq \mathcal{O}_u$ . Besides, the unlabeled data that belong to the class space  $\mathcal{O}_l$  are called in-distribution (ID) samples, while the unlabeled data only belonging to the label space  $\mathcal{O}_u$  are called out-of-distribution (OOD) samples. Therefore, the goal of OSSSL is to train a robust model to classify ID samples into the correct classes while learning to effectively detect OOD samples.

#### 3.1 OVERVIEW OF *OpenPrompt*

The main challenge of this task is to find an efficient way to train a robust network when the labels and categories of the used dataset are both imbalanced. Inspired by recent studies Jia et al. (2022); Bahng et al. (2022) on visual prompts, we propose a prompt-driven efficient OSSSL framework (termed *OpenPrompt*), to improve the efficiency of OSSSL task. In detail, the proposed framework includes two stages, *i.e.*, conducting pre-training on labeled data and fine-tuning on unlabeled data. In this way, a visual prompt is inserted into the data space Bahng et al. (2022), which can propagate class information from labeled to unlabeled data with only a small amount of trainable parameters while keeping the model frozen. Moreover, instead of introducing additional structures as the outlier detector, we propose a prompt-driven joint space learning mechanism to detect OOD samples.

To make full use of the detected OOD samples, we propose a prompt-wise contrastive representation strategy, which can further enlarge the distribution gap between ID and OOD samples. Instead of discarding the OOD samples as other OSSSL approaches Saito et al. (2021); Yu et al. (2020), we feed them into the network and train an OOD-specific visual prompt as negative samples for a prompt-wise contrastive representation.

As shown in Fig. 2, our framework utilizes a Teacher-Student structure, including a teacher model and a student model. Each model has four components: (1) a visual-encoder  $E(\cdot)$ , (2) a learnable visual prompt  $v_\phi$  parameterized by  $\phi$  in the form of pixels, (3) a prompt-driven joint space  $S(\cdot)$ , and (4) a closed-set classifier  $C(\cdot)$ .

In the pre-training stage,  $\mathcal{X}_l = \{(x_i, y_i)\}_{i=1}^n$  are feed into the  $E(\cdot)$  with the prompt  $v_\phi$  to obtain the feature representations  $F_l = \{f_i \in \mathbb{R}^D\}_{i=1}^n$ , which are then feed into the classifier  $C(\cdot)$  to obtain the prediction results  $\hat{Y} = \{\hat{y}_i\}_{i=1}^n$ . The labeled objective function  $\mathcal{L}_l$  is used to infer the loss with the ground-truth  $Y = \{y_i\}_{i=1}^n$ . Noted that, all parameters are set to be learned in this stage for training a reliable classification model and a visual prompt. Since there are no OOD samples contained during

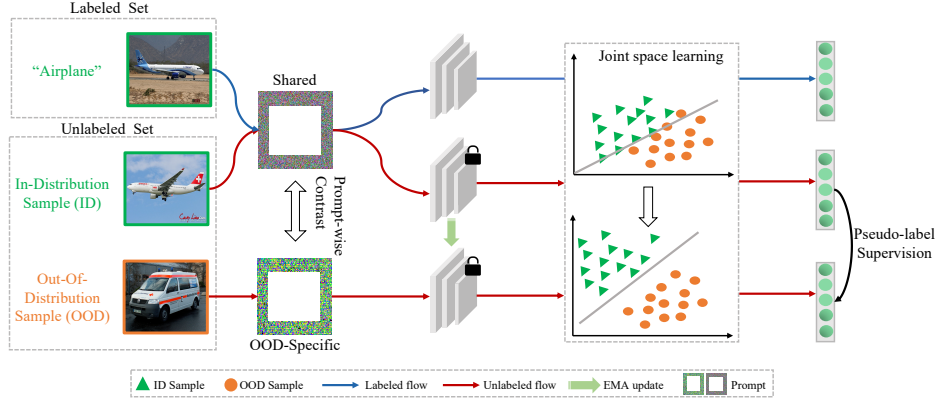


Figure 2: Overview of our proposed framework. In unlabeled flow, two different augmented inputs are fed into the network to obtain classification results. The joint space learning mechanism is used to enlarge the distribution difference distinguished by the shared ID-prompt for OOD detection. The pseudo-label supervision only uses confident pseudo-labels, *i.e.*, a threshold is set for selection.

this stage, it can not find an appropriate binary classifier in the joint space to distinguish ID and OOD data. Therefore, we set multiple binary-classifier candidates in this stage, which are used to be selected in the fine-tuning stage. See details of the candidate selection in §3.2.

In the fine-tuning stage, we feed  $\mathcal{X}_u = \{x_i\}_{i=1}^m$  into both the teacher and student networks with different augmentations, to obtain the feature representations  $F_u = \{f_i \in \mathbb{R}^D\}_{i=1}^m$  for unlabeled data. The joint space takes  $F_u$  as inputs and determines whether they belong to ID or OOD samples. Features of ID samples will be sent to the classifier and the final class prediction is calculated by  $\hat{Y} = \{\hat{y}_i\}_{i=1}^m = C(F_u)$ . Features of OOD samples will not be used for classification and all OOD samples will be resent to the network with an initialized OOD-specific visual prompt  $\bar{v}_\phi$  to train an OOD-specific prompt as negative samples for contrastive learning. The supervision signals used for the unlabeled objective function  $\mathcal{L}_u$  are pseudo labels received from the student network, and the classification results are from the teacher network. Noted that in this stage, only prompt  $v_\phi$  and OOD-specific prompt  $\bar{v}_\phi$  are updated through the objective function while the remaining network remains frozen. Following Tarvainen & Valpola (2017), the student network is updated via optimizing  $\mathcal{L}_l$  and  $\mathcal{L}_u$ , while the teacher network is updated via exponential moving average (EMA).

The labeled objective function  $\mathcal{L}_l$  only contains a supervised loss, while the unlabeled objective function contains three components, *i.e.*, (i) supervised objective, (ii) consistency objective, and (iii) contrastive learning objective, which can be formulated by

$$\mathcal{L}_u = \mathcal{L}_S + \mathcal{L}_C + \mathcal{L}_{CL}, \quad (1)$$

where  $\mathcal{L}_S$  denotes supervised loss,  $\mathcal{L}_C$  denotes consistent loss, and  $\mathcal{L}_{CL}$  denotes contrastive learning loss which is calculated through the prompt  $v_\phi$  and OOD-specific prompt  $\bar{v}_\phi$ .

### 3.2 PROMPT-DRIVEN JOINT SPACE LEARNING

#### 3.2.1 PROMPT DESIGN

We introduce a parameterized visual prompt  $v_\phi$  in the form of pixels to the input image  $x$  and form a prompted input  $x + v_\phi$ . Following Bahng et al. (2022),  $v_\phi$  is designed using the padding template to achieve the best performance. Therefore, given the prompt size  $p$ , the actual number of parameters is  $2Cp \times (H + W - 2p)$ , where  $C$ ,  $H$ , and  $W$  are the image channels, height, and width respectively.

#### 3.2.2 JOINT SPACE LEARNING

**Pre-training stage.** Different from Jia et al. (2022); Bahng et al. (2022), we learn a visual prompt in the pre-training stage, which is used to propagate the class information of labeled data to unlabeled ones in the fine-tuning stage. Specifically, the prompted input  $x + v_\phi$  is fed into the visual-encoder  $E(\cdot)$  to obtain the output feature representations. Then, the feature representations  $F_l = \{f_i \in \mathbb{R}^D\}_{i=1}^n$

are fed into the joint space  $S(\cdot)$  to enlarge the distribution gap for OOD detection. We assume that all ID samples form a cluster  $\mathcal{K}$  in the joint space, which can be defined by

$$\mathcal{K} = \{f_i, k_{ic}\}_{i=1}^n, \quad (2)$$

where  $k_{ic}$  denotes the cluster center of the ID samples. It is critical to initialize a binary classifier for OOD detection. We build a circle through the ID cluster  $\mathcal{K}$  and use its tangent lines as the initial binary classifiers. Given the cluster center  $k_{ic}$ , we define the radius  $r$  of the circle as follows:

$$r = \max_{i \in n} d(f_i, k_{ic}), \quad (3)$$

where  $d(\cdot, \cdot)$  denotes an Euclidean distance. To ensure all ID samples are correctly detected, we choose the furthest sample and use the distance between it and the cluster center as the radius. As mentioned in §3.1, multiple binary classifier candidates are set for selection in the fine-tuning stage. To achieve this, we choose the top  $N$  furthest samples from the cluster center and build tangent lines on each of them. Due to the furthest sample is used to form the circle, therefore we have  $N - 1$  samples which are not on the boundary of the circle. As the tangent line of a circle must pass through a point on the boundary of the circle, we push these  $N - 1$  points outward some distance to ensure that they lie on the boundary of the circle. Finally, the initial classifier candidates can be defined by

$$D_j(f_i) = \begin{cases} F(f_i), & j = 1 \\ F(f_i) + (r - d(f_i, k_{ic})), & 1 < j \leq N \end{cases} \quad (4)$$

where  $F(\cdot)$  denotes the tangent function Thurston (1964). It is worth noting that we can obtain promising performance when  $N = 5$  as discussed in §4.3.

**Fine-tuning stage.** The parameterized visual prompt is also added to the unlabeled input image in this stage. It is worth noting that unlike others Jia et al. (2022) which learn a prompt from an initialized one in this stage, our method here inherits the visual prompt  $v_\phi$  learned from the pre-training stage. Given feature representations  $F_u = \{f_i \in \mathbb{R}^D\}_{i=1}^m$  obtained through the prompted unlabeled images, we then feed  $F_u$  into the joint space  $S(\cdot)$  for OOD detection. Since the labeled and unlabeled ID samples share similar classes, the outputs of  $D_j(F_l)$  should be close to those of  $D_j(F_u)$  from ID data. Therefore, we can detect OOD data by:

$$\begin{cases} x_i \text{ is ID sample } x_{id}, & \text{if } |D_j(f_i) - \overline{D_j(F_l)}| \leq 0.1 \\ x_i \text{ is OOD sample } x_{ood}, & \text{otherwise} \end{cases} \quad (5)$$

where  $\overline{D_j(F_l)}$  denotes the average of the outputs of  $D_j(F_l)$ .

Then we can get the rate  $r_j$  of OOD samples in unlabeled data using the  $j$ -th binary classifier candidate:

$$r_j = \frac{n_j}{m}, \quad (6)$$

where  $n_j$  denotes the number of detected OOD samples. The candidate with the largest rate will be selected as the initial binary classifier  $D(\cdot)$  for both the student and teacher networks.

Then we resend  $F_u = \{f_i \in \mathbb{R}^D\}_{i=1}^m$  back to the joint space  $S(\cdot)$  in both two networks to conduct OOD detection and update the binary classifier. Due to the binary classifier being built on ID samples from labeled data, which will cause it biased toward them, therefore we need to modify the binary classifier. The linear classifier can be justified through a calibrated stacking method Chao et al. (2016), which directly shifts the decision boundary through a fixed value. However, this will degrade the performance in detecting ID samples. Inspired by Baek et al. (2021), instead of shifting, we modulate the decision boundary with the Apollonius circle. Specifically, we first use the initial  $D(\cdot)$  to detect OOD samples from unlabeled data and calculate the initial cluster center  $k_{oc}$  of these samples. Then, given a feature  $f_i$  from  $F_u$ , we can compute two distances from  $k_{ic}$  and  $k_{oc}$  to  $f_i$  by

$$d_1 = d(f_i, k_{ic}) \text{ and } d_2 = d(f_i, k_{oc}), \quad (7)$$

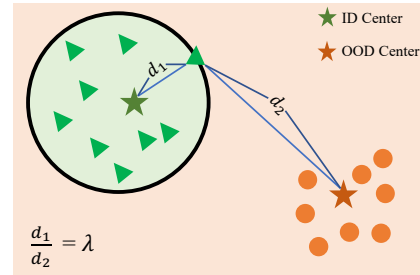


Figure 3: Apollonius circle Kim (2009) built on two centers. Definition of Apollonius circle is given in Appendix A.1.

where  $d_1 \leq d_2$ . Therefore, the Apollonius circle used as the binary-classifier can be formulated with an adjustable parameter  $\lambda$  as follows:

$$\mathcal{A}(\lambda) = \{f_i \mid d_1 : d_2 = \lambda\}, \quad (8)$$

where  $\mathcal{A}(\lambda)$  denotes the boundary of the Apollonius circle (see Fig 3). Therefore, the decision rule of the binary classifier is modulated as follows:

$$D_{ac}(f_i) = c_{id} \mathbb{1} \left[ \frac{d_1}{d_2} \leq \lambda \right] + c_{ood} \mathbb{1} \left[ \frac{d_1}{d_2} > \lambda \right], \quad (9)$$

where  $c_{id}$  or  $c_{ood}$  denotes an identification that is ID or OOD sample, and  $\mathbb{1}$  denotes an indicator function that returns 1 if the argument is true, and 0 otherwise. Noted that two cluster centers  $k_{ic}$  and  $k_{oc}$  are updated through the new detected ID and OOD samples after each detection process. After that, we then send all ID samples to the classifier  $C(\cdot)$  for prediction and use the detected OOD samples for prompt-wise contrastive representation learning (as mentioned in §3.2).

**Prompt-wise contrastive representation.** Existing approaches often filter out the OOD samples once detect them. We notice that those unused data could be used to train an OOD-specific prompt as negative samples to further improve the model performance. Therefore, at the end of each training epoch, we collect all the detected OOD samples and resend them to the network with a new initial visual prompt. We call this prompt an OOD-specific prompt  $\overline{v_\phi}$  and treat them as negative samples to apply contrastive learning.

### 3.3 OBJECTIVE FUNCTION

As discussed in §3.1, our overall objective function includes supervised loss  $\mathcal{L}_S$ , consistent loss  $\mathcal{L}_C$ , and contrastive learning loss  $\mathcal{L}_{CL}$ .

**Supervised loss.** Following existing classification works, we use cross-entropy (CE) loss  $\mathcal{L}_{CE}$  as the supervised loss. Noted that for the pre-training stage, we directly computer the  $\mathcal{L}_{CE}$  through ground-truth  $\mathcal{L}_{CE}(Y, \hat{Y})$ . However, for the fine-tuning stage, there are no ground-truth labels used as supervision signals. Therefore, we set the prediction results from student network as pseudo-labels  $\hat{Y} = \{\hat{y}_i\}_{i=1}^n$  for supervision signals and use a threshold Sohn et al. (2020)  $\eta = 0.7$  to make sure all used pseudo-labels are reliable.

**Consistent loss.** For consistent loss, different from existing works which calculate the per pixel-level consistency of each feature map, we calculate the consistency through prompting and joint space learning. As mentioned in §3.1, we conduct different augmentation strategies on the input samples. Since two different augmented inputs use the same prompt, they should share similar feature representations. Therefore, we use the Euclidean distance in the joint space as the consistency loss:

$$\mathcal{L}_C = d(k_{ic}, f_s) + d(k_{oc}, f_s) - d(k_{ic}, f_t) - d(k_{oc}, f_t), \quad (10)$$

where  $f_s$  and  $f_t$  denote features obtained from student and teacher networks, respectively.

**Contrastive learning loss.** We resend OOD samples with a random initialized visual prompt  $\overline{v_\phi}$  to both the student and teacher networks, and then use  $\mathcal{L}_C$  for optimization to get the OOD-specific prompt. Since this prompt is only trained using OOD samples, we treat this one as a negative sample. Our goal is to enlarge the distribution gap between  $v_\phi$  and  $\overline{v_\phi}$ . Therefore, the contrastive learning loss can be defined by

$$\mathcal{L}_{CL} = 1 - \mathcal{L}_{\cos}(v_\phi, \overline{v_\phi}), \quad (11)$$

where  $\mathcal{L}_{\cos}$  denotes the cosine similarity loss.

## 4 EXPERIMENTS

**Implementation Details.** Following Saito et al. (2021), we implemented our network based on Wide ResNet-28-2 Zagoruyko & Komodakis (2016) (for CIFAR10 and CIFAR100) and ResNet-18 (for ImageNet-30). The models are trained on one NVIDIA Titan X with a 12-GB GPU. We use the

Table 1: Average (with standard deviation) AUROC results. Paras. denotes the total learnable parameters of the model based on Wide ResNet-28-2 for CIFAR 10 and CIFAR100, and based on ResNet-18 for ImageNet-30. The number of classes for ID and OOD and the number of labeled samples for each class are shown in each column.

Dataset	Paras.(M)	CIFAR10			CIFAR100		CIFAR100		ImageNet-30
No. of ID / OOD		6 / 4			55 / 45		80 / 20		20 / 10
No. of Labeled samples		50	100	400	50	100	50	100	10 %
Labeled Only	- / -	63.9 $\pm$ 0.5	64.7 $\pm$ 0.5	76.8 $\pm$ 0.4	76.6 $\pm$ 0.9	79.9 $\pm$ 0.9	70.3 $\pm$ 0.5	73.9 $\pm$ 0.9	80.3 $\pm$ 1.0
FixMatch	23.83/33.22	56.1 $\pm$ 0.6	60.4 $\pm$ 0.4	71.8 $\pm$ 0.4	72.0 $\pm$ 1.3	75.8 $\pm$ 1.2	64.3 $\pm$ 1.0	66.1 $\pm$ 0.5	88.6 $\pm$ 0.5
MTC	23.36/32.66	96.6 $\pm$ 0.6	98.2 $\pm$ 0.3	98.9 $\pm$ 0.1	81.2 $\pm$ 3.4	80.7 $\pm$ 4.6	79.4 $\pm$ 2.5	73.2 $\pm$ 3.5	93.8 $\pm$ 0.8
OpenMatch	23.44/32.73	99.3 $\pm$ 0.3	99.7 $\pm$ 0.2	<b>99.3<math>\pm</math>0.2</b>	87.0 $\pm$ 1.1	86.5 $\pm$ 2.1	86.2 $\pm$ 0.6	86.8 $\pm$ 1.4	96.4 $\pm$ 0.7
<b>Ours w/o CL</b>	<b>0.08/0.08</b>	98.7 $\pm$ 0.3	99.5 $\pm$ 0.7	98.8 $\pm$ 0.1	86.7 $\pm$ 1.2	86.4 $\pm$ 0.2	86.0 $\pm$ 0.5	86.4 $\pm$ 0.3	95.8 $\pm$ 1.1
<b>Ours</b>	<b>0.16/0.16</b>	<b>99.4<math>\pm</math>0.1</b>	<b>99.7<math>\pm</math>0.5</b>	99.2 $\pm$ 0.1	<b>87.2<math>\pm</math>1.2</b>	<b>87.0<math>\pm</math>0.6</b>	<b>86.5<math>\pm</math>1.5</b>	<b>87.3<math>\pm</math>0.4</b>	<b>97.1<math>\pm</math>0.2</b>

standard SGD with an initial learning rate of 0.3, and momentum set as 0.9. The hyperparameters prompt size  $p$ , candidate number  $N$ , and  $\lambda$  are empirically set to 40, 5, and 0.5, separately (see discussions in §4.3). The average result of three runs and its standard deviation are recorded.

**Datasets.** The proposed *OpenPrompt* is evaluated on three OSSSL benchmark image classification datasets, including CIFAR-10, CIFAR-100 Krizhevsky et al. (2009) and ImageNet-30 Deng et al. (2009). In the OSSSL task, the test set is assumed to contain both known (ID) and unknown (OOD) classes Saito et al. (2021); Yu et al. (2020). Specifically, for CIFAR10, we use the animal classes (six classes) as ID data and the other four classes as OOD data. For CIFAR100, we have two experimental settings: 80 classes as ID data (20 classes as OOD data) and 55 classes as ID data (45 classes as OOD data) Saito et al. (2021). For ImageNet-30, we pick the first 20 classes (in alphabetical order) as ID data and use the remaining 10 classes as OOD data Saito et al. (2021).

**Baselines.** We use MTC Yu et al. (2020) and OpenMatch Saito et al. (2021) with the source codes as the OSSSL baselines. Following Saito et al. (2021), we train two models, one using only labeled samples (Labeled Only) and the other one employing the FixMatch Sohn et al. (2020) method. Then, we add the outlier detector from OpenMatch to these two models for OOD detection. The hyper-parameters of all the methods are tuned by maximizing the AUROC on the validation set.

#### 4.1 RESULTS

**CIFAR10 and CIFAR100.** Following MTC Yu et al. (2020) and OpenMatch Saito et al. (2021), we use AUROC as the evaluation metric. The results are shown in Table 1, where the number of classes for ID and OOD and the number of labeled samples for each class are shown in each column. Since we employ the ResNet-28-2 He et al. (2016) network architecture for CIFAR10 and CIFAR100, and the ResNet-18 for ImageNet-30, the number of learnable parameters for each of the two basic network architectures are respectively recorded. As can be seen from this table, *OpenPrompt* achieves the best performance in most cases on both CIFAR10 and CIFAR100. For example, *OpenPrompt* increased the AUROC values from 86.4 to 87.3 in CIFAR100 at 100 labels. More importantly, the learnable parameters of our proposed *OpenPrompt* are far less than the existing methods, *e.g.*, 23.44M  $\rightarrow$  **0.16 M** compared with OpenMatch and 23.8M  $\rightarrow$  **0.16 M** compared with FixMatch Sohn et al. (2020), which are no more than 1%. This is because our method adapts a model on labeled data to unlabeled by modifying the data space while others modify the model space Bahng et al. (2022).

**ImageNet-30.** Here, we also evaluate the proposed *OpenPrompt* on the more challenging dataset ImageNet-30 Saito et al. (2021). As described in §4, we alphabetically select the first 20 classes as ID classes and the remaining 10 classes as OOD classes. As shown in the rightmost column of Table 1, *OpenPrompt* also has the highest AUROC accuracy on ImageNet-30. In particular, we see our proposed *OpenPrompt* leading to substantial performance gains (*i.e.*, 96.4  $\rightarrow$  **97.1**) with 10 % labels in ImageNet-30 compared to OpenMatch. All methods use the ResNet-18 network architecture on the ImageNet-30 dataset, and the learnable parameters are 33.2M, 32.66M, and 32.73M, respectively. However, our method freezes the entire network except for visual cues, and its

Table 2: AUROC evaluation of OOD detection. The higher values, the better performance. Supervised models indicate our proposed OpenMatch but are trained with fully labeled data.

(a) Training on CIFAR10, where there is 100 labeled data per class and unlabeled data.

Method	CIFAR10	SVHN	LSUN	ImageNet	CIFAR100	MEAN
Labeled Only	64.7 $\pm$ 1.0	83.6 $\pm$ 1.0	78.9 $\pm$ 0.9	80.5 $\pm$ 0.8	80.4 $\pm$ 0.5	80.8 $\pm$ 0.8
FixMatch Sohn et al. (2020)	60.4 $\pm$ 0.4	79.9 $\pm$ 1.0	67.7 $\pm$ 2.0	76.9 $\pm$ 1.1	71.3 $\pm$ 1.1	73.9 $\pm$ 1.3
MTC Yu et al. (2020)	98.2 $\pm$ 0.3	87.6 $\pm$ 0.5	82.8 $\pm$ 0.6	96.5 $\pm$ 0.1	90.0 $\pm$ 0.3	89.2 $\pm$ 0.4
OpenMatch Saito et al. (2021)	99.7 $\pm$ 0.1	93.0 $\pm$ 0.4	92.7 $\pm$ 0.3	<b>98.7<math>\pm</math>0.1</b>	95.8 $\pm$ 0.4	95.0 $\pm$ 0.3
<b>Ours</b>	<b>99.7<math>\pm</math>0.2</b>	<b>94.1<math>\pm</math>1.1</b>	<b>93.6<math>\pm</math>0.7</b>	97.4 $\pm$ 0.3	<b>96.2<math>\pm</math>0.5</b>	<b>95.4<math>\pm</math>1.3</b>
Supervised	89.4 $\pm$ 1.0	95.6 $\pm$ 0.5	89.5 $\pm$ 0.7	90.8 $\pm$ 0.4	90.4 $\pm$ 1.0	91.6 $\pm$ 0.6

(b) Training on ImageNet-30, where there is 10 % of labeled data and unlabeled data.

Method	ImageNet-30	LSUN	DTD	CUB	Flowers	Caltech	Dogs	MEAN
Labeled Only	80.3 $\pm$ 0.5	85.9 $\pm$ 1.4	75.4 $\pm$ 1.0	77.9 $\pm$ 0.8	69.0 $\pm$ 1.5	78.7 $\pm$ 0.8	84.8 $\pm$ 1.0	78.6 $\pm$ 1.1
FixMatch Sohn et al. (2020)	88.6 $\pm$ 0.5	85.7 $\pm$ 0.1	83.1 $\pm$ 2.5	81.0 $\pm$ 4.8	81.9 $\pm$ 1.1	83.1 $\pm$ 3.4	86.4 $\pm$ 3.2	83.0 $\pm$ 1.9
MTC Yu et al. (2020)	93.8 $\pm$ 0.8	78.0 $\pm$ 1.0	59.5 $\pm$ 1.5	72.2 $\pm$ 0.9	76.4 $\pm$ 2.1	80.9 $\pm$ 0.9	78.0 $\pm$ 0.8	74.2 $\pm$ 1.2
OpenMatch Saito et al. (2021)	96.3 $\pm$ 0.7	89.9 $\pm$ 1.9	84.4 $\pm$ 0.5	87.7 $\pm$ 1.0	80.8 $\pm$ 1.9	87.7 $\pm$ 0.9	<b>92.1<math>\pm</math>0.4</b>	87.1 $\pm$ 1.1
<b>Ours</b>	<b>97.1<math>\pm</math>0.5</b>	<b>90.7<math>\pm</math>1.7</b>	<b>85.3<math>\pm</math>0.1</b>	<b>89.2<math>\pm</math>0.3</b>	<b>83.2<math>\pm</math>1.5</b>	<b>88.6<math>\pm</math>1.0</b>	90.9 $\pm$ 0.2	<b>88.0<math>\pm</math>0.8</b>
Supervised	92.8 $\pm$ 0.8	94.4 $\pm$ 0.5	92.7 $\pm$ 0.4	91.5 $\pm$ 0.9	88.2 $\pm$ 1.0	89.9 $\pm$ 0.5	92.3 $\pm$ 0.8	91.3 $\pm$ 0.7

learnable parameters are much smaller than various SOTA baselines, which are just 0.16 M. This indicates that *OpenPrompt* achieves promising performance on complex and challenging datasets.

## 4.2 OOD DETECTION

Since our *OpenPrompt* could detect OOD samples from unlabeled data, we evaluated the performance of our prompt-driven joint space learning in separating ID samples from OOD samples in unlabeled data. Following Saito et al. (2021), we let the following datasets as OOD samples: SVHN Netzer et al. (2011), LSUN Yu et al. (2015), CIFAR100 and ImageNet for CIFAR10 experiments (see Table 2a), and LSUN, Dogs Khosla et al. (2011), CUB-200 Wah et al. (2011), Caltech Griffin et al. (2007), DTD Cimpoi et al. (2014) and Flowers Nilsback & Zisserman (2006) for ImageNet-30 (see Table 2b). The separation between the ID and OOD samples is evaluated by AUROC. We train a model utilizing all labeled examples of ID samples to show the gap from a supervised model. As can be seen in Table 2, *OpenPrompt* outperforms the various SOTA baselines by 1.1% on SVHN and 0.4% on average on CIFAR10. On ImageNet-30, *OpenPrompt* improved 2.4% on Flowers and 0.9% on average compared with OpenMatch. These results demonstrate that our proposed *OpenPrompt* is more sensitive to the representation gaps between ID and OOD samples, which leads to a more robust OSSL framework while they are exposed to unlabeled data containing outliers.

## 4.3 ANALYSIS

**Effectiveness of prompt-driven joint space.** To investigate the effectiveness of our proposed prompt-driven joint space learning mechanism, we built a model that uses a discriminator as an OOD detector to replace it (termed *w/o* joint space). We record the AUROC values on CIFAR10 dataset with 100 labeled data per class and unlabeled data in Fig. 4 (a), where the learning rate is decayed at epoch 400. As can be seen, our model converges faster and has higher accuracy (see the red curve). However, our method can still achieve satisfactory accuracy without the prompt-driven joint space mechanism (see the blue curve). This is likely because the visual prompt  $v_\phi$  inherited from the labeled training stage is sensitive to the variability in the data space as it only has knowledge of the ID samples. However, the results are less than ours because the detector is randomly initialized, which will lead to significant oscillations in the early stages of training (see 10-70 epochs of the blue curve).

**Effectiveness of prompt-wise contrastive representation.** Here, we verify our claim that prompt-wise contrastive representation can further shape the joint space. As shown in the last two rows of Table 1, we use **Ours** (*w/o* CL) to represent represents our model, but without prompt-wise contrastive learning (CL). As can be seen from this table, without the CL mechanism, the AUROC



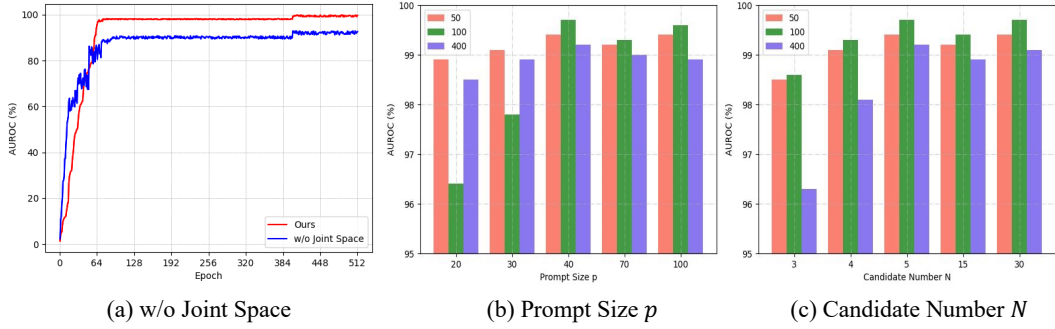


Figure 4: A set of ablative studies on CIFAR10. (a) The curve of the ablation study of joint space. (b) and (c) are the histograms of the different values of prompt size  $p$  and candidate number  $N$ , where three colors represent the different numbers of labeled samples *per class*.

results of **Ours** (w/o CL) will be lower than **Ours**. When we derive the ID-specific and OOD-specific prompt for the prompt-wise contrastive representation, our method provides the highest AUROC values. It indicates that the prompt-wise contrastive representation can push the OOD samples away from the ID ones in unlabeled data, thereby enhancing the discrimination of outliers.

**Prompt discussion.** We aim to find a balance between prompt size and model performance. Usually, it involves the template and size of the visual prompt. Following Bahng et al. (2022), we choose the padding as the prompt template. Here, we analyze how the prompt sizes  $p$  influence our method. As shown in

Table 3: OOD detection results under different values of  $\lambda$ .

$\lambda$	LSUN	DTD	CUB	Flowers	Caltech	Dogs	MEAN
0.1	88.1 $\pm$ 0.2	83.4 $\pm$ 0.7	86.5 $\pm$ 0.8	81.9 $\pm$ 1.1	87.3 $\pm$ 0.5	90.1 $\pm$ 0.1	86.2 $\pm$ 0.5
0.3	89.9 $\pm$ 0.5	84.7 $\pm$ 1.5	88.0 $\pm$ 0.3	82.5 $\pm$ 2.1	88.3 $\pm$ 2.4	90.6 $\pm$ 1.7	87.3 $\pm$ 1.4
<b>0.5</b>	<b>90.7<math>\pm</math>1.7</b>	<b>85.3<math>\pm</math>0.1</b>	<b>89.2<math>\pm</math>0.3</b>	<b>83.2<math>\pm</math>1.5</b>	<b>88.6<math>\pm</math>1.0</b>	<b>90.9<math>\pm</math>0.2</b>	<b>88.0<math>\pm</math>0.8</b>
0.8	90.5 $\pm$ 0.9	84.9 $\pm$ 1.3	89.1 $\pm$ 1.8	82.7 $\pm$ 2.9	87.9 $\pm$ 1.2	89.8 $\pm$ 3.4	87.5 $\pm$ 1.9
1.0	90.4 $\pm$ 2.2	84.2 $\pm$ 0.3	88.9 $\pm$ 0.8	82.6 $\pm$ 0.6	88.5 $\pm$ 1.1	90.0 $\pm$ 0.1	87.4 $\pm$ 0.9

Fig. 4 (a), our model achieves the best AUROC scores at  $p = 40$ . When  $p = 100$ , the AUROC scores similar to the results of  $p = 40$ . However, the number of learnable parameters will be increased from 0.16M to 0.35M resulting in lower efficiency.

**Ablation study on  $N$ .** As mentioned in §3.2, since the distribution of the unlabeled dataset is unknown in the pre-training stage, we should define multiple candidates for the initial binary classifier selection in the fine-tuning stage to find a better OOD center to construct the Apollonius circle. As can be seen in Fig. 4 (c), the greater the number of candidates, the better results. However, the performance gradually stabilized when  $N \geq 5$ . It should be noted that we should compute the  $\mathcal{A}_o$  values (see §3.2) for each candidate, thereby the greater the value of  $N$ , the greater the computational costs. Therefore, according to the result of Figure c, we set  $N = 5$  in our experiments.

**Ablation study on  $\lambda$ .** Here, we analyze the value of  $\lambda$  how to influence the Apollonius circle. As shown in Table 3, the larger value of  $\lambda$ , the greater AUROC performance of the OOD detection, i.e., 86.2 ( $\lambda = 0.1$ )  $\rightarrow$  **88.0** ( $\lambda = 0.5$ ). When  $\lambda > 0.5$ , the performances are no longer improving. Therefore, we set the  $\lambda = 0.5$  to build the Apollonius circle.

## 5 CONCLUSION

In this paper, we propose a new efficient framework for OSSSL (termed *OpenPrompt*). Based on visual prompting, *OpenPrompt* focuses on data space adaption instead of fine-tuning the whole model, which could achieve the state-of-the-art performance with less than 1% learnable parameters compared with other approaches. Our *OpenPrompt* can effectively detect OOD samples thanks to the prompt-driven joint space learning mechanism that enlarges the distribution gap between ID and OOD samples. Furthermore, we reuse the OOD samples through prompt-wise contrastive representation to explore the structural information of ID and OOD samples and further shape the joint space. We believe this paper could inspire a shift in the direction of future research toward efficient OSSSL.

## REFERENCES

- Donghyeon Baek, Youngmin Oh, and Bumsu Ham. Exploiting a joint embedding space for generalized zero-shot semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9536–9545, 2021.
- Hyojin Bahng, Ali Jahani, Swami Sankaranarayanan, and Phillip Isola. Exploring visual prompts for adapting large-scale models. *arXiv preprint arXiv:2203.17274*, 2022.
- David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin Raffel. Mixmatch: A holistic approach to semi-supervised learning. *Advances in neural information processing systems*, 32, 2019.
- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.
- Wei-Lun Chao, Soravit Changpinyo, Boqing Gong, and Fei Sha. An empirical study and analysis of generalized zero-shot learning for object recognition in the wild. In *European conference on computer vision*, pp. 52–68. Springer, 2016.
- Yanbei Chen, Xiatian Zhu, Wei Li, and Shaogang Gong. Semi-supervised learning under class distribution mismatch. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 3569–3576, 2020.
- Mircea Cimpoi, Subhansu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3606–3613, 2014.
- Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.
- Terrance DeVries and Graham W Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017.
- Geoff French, Samuli Laine, Timo Aila, Michal Mackiewicz, and Graham Finlayson. Semi-supervised semantic segmentation needs strong, varied perturbations. *British Machine Vision Conference*, 2019.
- Gregory Griffin, Alex Holub, and Pietro Perona. Caltech-256 object category dataset. 2007.
- Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.
- Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual prompt tuning. In *European Conference on Computer Vision (ECCV)*, 2022.
- Zhengbao Jiang, Frank F Xu, Jun Araki, and Graham Neubig. How can we know what language models know? *Transactions of the Association for Computational Linguistics*, 8:423–438, 2020.
- Aditya Khosla, Nityananda Jayadevaprakash, Bangpeng Yao, and Fei-Fei Li. Novel dataset for fine-grained image categorization: Stanford dogs. In *Proc. CVPR workshop on fine-grained visual categorization (FGVC)*, volume 2. Citeseer, 2011.
- BJ Kim. The circle of apollonius. *Mathematics Education Program J. Wilson, EMAT*, 6690, 2009.
- Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.
- Samuli Laine and Timo Aila. Temporal ensembling for semi-supervised learning. In *International Conference on Learning Representations*, 2017.

- Dong-Hyun Lee et al. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, volume 3, pp. 896, 2013.
- Junnan Li, Caiming Xiong, and Steven C.H. Hoi. Comatch: Semi-supervised learning with contrastive graph regularization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9475–9484, October 2021.
- Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. 2011.
- M-E Nilsback and Andrew Zisserman. A visual vocabulary for flower classification. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’06)*, volume 2, pp. 1447–1454. IEEE, 2006.
- Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pp. 8748–8763. PMLR, 2021.
- Antti Rasmus, Mathias Berglund, Mikko Honkala, Harri Valpola, and Tapani Raiko. Semi-supervised learning with ladder networks. *Advances in neural information processing systems*, 28, 2015.
- Kuniaki Saito, Donghyun Kim, and Kate Saenko. Openmatch: Open-set semi-supervised learning with open-set consistency regularization. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, 2021.
- Taylor Shin, Yasaman Razeghi, Robert L Logan IV, Eric Wallace, and Sameer Singh. Autoprompt: Eliciting knowledge from language models with automatically generated prompts. *arXiv preprint arXiv:2010.15980*, 2020.
- Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin A Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *Advances in neural information processing systems*, 33:596–608, 2020.
- Yi-Lin Sung, Jaemin Cho, and Mohit Bansal. Lst: Ladder side-tuning for parameter and memory efficient transfer learning. *arXiv preprint arXiv:2206.06522*, 2022.
- Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. *Advances in neural information processing systems*, 30, 2017.
- HA Thurston. On the definition of a tangent-line. *The American Mathematical Monthly*, 71(10): 1099–1103, 1964.
- Vikas Verma, Kenji Kawaguchi, Alex Lamb, Juho Kannala, Arno Solin, Yoshua Bengio, and David Lopez-Paz. Interpolation consistency training for semi-supervised learning. *Neural Networks*, 145: 90–106, 2022.
- Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011.
- Xiang Wang, Shiwei Zhang, Zhiwu Qing, Yuanjie Shao, Changxin Gao, and Nong Sang. Self-supervised learning for semi-supervised temporal action proposal. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1905–1914, June 2021.
- Yuchao Wang, Haochen Wang, Yujun Shen, Jingjing Fei, Wei Li, Guoqiang Jin, Liwei Wu, Rui Zhao, and Xinyi Le. Semi-supervised semantic segmentation using unreliable pseudo-labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4248–4257, 2022.

- Lihe Yang, Wei Zhuo, Lei Qi, Yinghuan Shi, and Yang Gao. St++: Make self-training work better for semi-supervised semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4268–4277, 2022.
- Fisher Yu, Ari Seff, Yinda Zhang, Shuran Song, Thomas Funkhouser, and Jianxiong Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015.
- Qing Yu, Daiki Ikami, Go Irie, and Kiyoharu Aizawa. Multi-task curriculum framework for open-set semi-supervised learning. In *European Conference on Computer Vision*, pp. 438–454. Springer, 2020.
- Jianlong Yuan, Yifan Liu, Chunhua Shen, Zhibin Wang, and Hao Li. A simple baseline for semi-supervised semantic segmentation with strong data augmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 8229–8238, 2021.
- Sangdo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 6023–6032, 2019.
- Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.
- Mingkai Zheng, Shan You, Lang Huang, Fei Wang, Chen Qian, and Chang Xu. Simmatch: Semi-supervised learning with similarity matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14471–14481, 2022.

## A APPENDIX

### A.1 DEFINATION OF APOLLONIUS CIRCLE

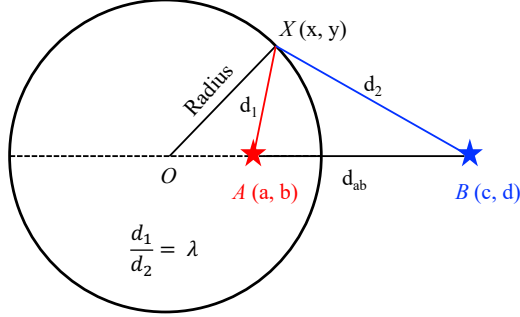


Figure 5: Apollonius circle in a two-dimensional Euclidean space.

Given two points  $A$  and  $B$ , Apollonius of Perga defines a circle as a set of points  $X$  which satisfies the equation

$$|XA| = \lambda |XB|, \quad (12)$$

where  $\lambda$  denotes a positive real number,  $|XA|$  and  $|XB|$  denotes the Euclidean distance from  $X$  to  $A$  and  $B$ , respectively. Assuming  $X$ ,  $A$  and  $B$  are three points in a two-dimensional Euclidean space (see Fig 5), we can define the radius of the Apollonius circle as follows:

$$\begin{aligned} \frac{|XA|}{|XB|} &= \lambda \\ \Rightarrow \frac{d_1}{d_2} &= \lambda \\ \Rightarrow \frac{\sqrt{(x-a)^2 + (y-b)^2}}{\sqrt{(x-c)^2 + (y-d)^2}} &= \lambda \\ \Rightarrow \left(x - \frac{a - (\lambda)^2 c}{1 - \lambda^2}\right)^2 + \left(y - \frac{b - (\lambda)^2 d}{1 - \lambda^2}\right)^2 &= \frac{(\lambda)^2}{(1 - (\lambda)^2)^2} ((a-c)^2 + (b-d)^2) \end{aligned} \quad (13)$$

From the last row in Eq 13, we can define the radius of the Apollonius circle as follows:

$$\begin{aligned} radius &= \frac{\lambda}{1 - (\lambda)^2} \sqrt{(a-c)^2 + (b-d)^2} \\ &= \frac{\lambda}{1 - \lambda^2} d_{12}, \end{aligned} \quad (14)$$

where  $d_{12}$  denotes the Euclidean distance between point  $A$  and point  $B$  (as shown in Fig 5).