SU-SSL: Maximize Performance in Unseen Classes and Maintain Safeness in Seen Classes

Anonymous Author(s)
Affiliation
Address
email

Abstract

Semi-supervised learning (SSL) has received tremendous attention due to its ability to leverage unlabeled data. Existing SSL methods typically assume all unlabeled data are from seen classes, i.e., classes are observed in the labeled dataset. However, in real-world applications, unseen classes are commonly occurred, which severely degrade SSL performance on seen classes. Open-set SSL methods are designed to maintain safeness on seen classes, but they fail to classify unseen classes. Novel class discovery (NCD) methods aim to discover unseen classes automatically but it is unsafe for seen classes. In this paper, we develop a new SSL approach, called Safe Unseen classification Semi-Supervised Learning (SU-SSL), which can not only classify unseen classes automatically but also maintain safeness on seen classes. Our approach consists of two modules: *Unseen Class Classification* and *Adaptive Threshold*. Specifically, we first improve the SSL methods to discover unseen classes by proposing a new unseen class classification objective that can exploit pairwise similarity and eliminate potential noisy pairs, and then bridge the performance gap between seen and unseen classes by proposing an adaptive threshold based SSL objective. Extensive empirical evaluations show our approach achieves 37.7% improvement in unseen-class classification compared with SSL methods, and 26.3% improvement in seen class compared with NCD methods.

1 Introduction

Machine learning, especially deep learning, has achieved great success in various tasks by leveraging sufficient labeled training data [1]. However, for many practical tasks, it can be difficult to attain a large number of labeled samples due to the high cost of the data labeling process [2, 3], which limits the widespread adoption of machine learning techniques.

Semi-supervised learning (SSL)[4] provides a powerful framework for leveraging unlabeled data when labels are limited or expensive to obtain. There has been a rapid development of SSL methods in recent years, such as entropy minimization methods [5, 6], consistency regularization methods [7–10], and holistic methods[11–15]. It has been reported that in certain cases, such as image classification [11], SSL methods can achieve the performance of purely supervised learning even when a substantial portion of the labels in a given dataset has been discarded.

All of the positive results of SSL, however, are based on a basic assumption that there are labels for each of the classes that one wishes to learn, i.e., all training and testing data are from seen classes that are observed in the labeled dataset. Such an assumption is difficult to hold in many real-world applications. For example, in the product recognition task [16], thousands of new types of products (i.e., unseen classes) are introduced to the supermarkets every day, and it would be very expensive to label them all. We give an illustration of this problem in Figure 1.

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

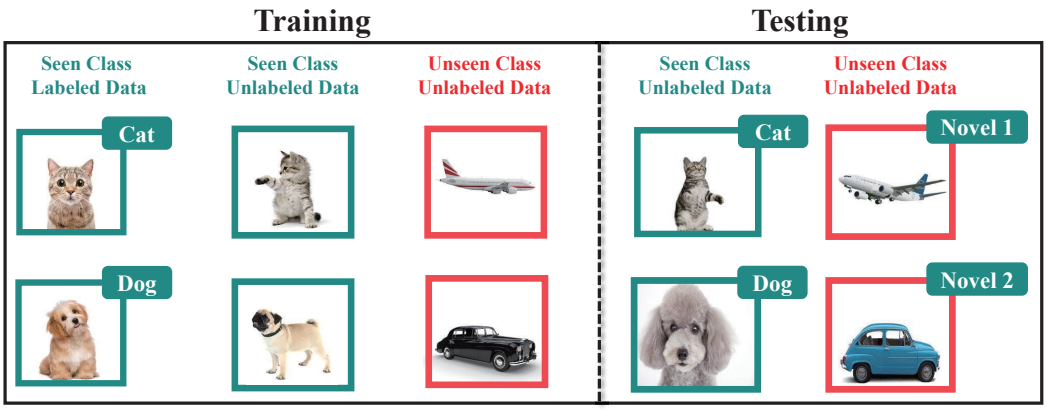


Figure 1: SSL with unseen classes. Training data includes labeled samples from a few seen classes as well as unlabeled samples from both seen and unseen classes. Testing data includes samples from both seen and unseen classes. The goal is to classify not only seen classes samples into accurate categories but also partition unseen classes samples into proper clusters.

It has been reported that SSL performance degrades severely on seen classes with unseen class unlabeled data [17, 18]. Open-set SSL methods [18–21] have been proposed to decrease the negative impact of unseen classes and maintain safeness on seen classes. However, these methods simply detect and drop out unseen classes, and fail to classify unseen classes. Novel class discovery (NCD) aims to discover unseen classes automatically. However, they ignore the classification task on seen classes, which leads to performance degradation in seen classes.

It is evident that both existing SSL methods and NCD methods could not tackle the problem concerned in this paper. This inspires us to consider answering the following question in this study:

Can we design an SSL algorithm that not only classifies unseen classes accurately but also maintains safeness in seen classes?

To this end, we propose a safe SSL algorithm for unseen class classification, SU-SSL, which consists of two key modules: unseen class classification and adaptive threshold. Specifically, we first propose a novel unseen class classification loss that can exploit pairwise similarity to classify similar sample pairs into the same class and eliminate noisy pairs based on a novel similarity filter. We then propose an adaptive threshold with distribution alignment to alleviate the issue that different learning paces between seen classes and unseen classes. We evaluate our approach on CIFAR-10, CIFAR-100, and ImageNet-100 datasets, and the results show that SU-SSL achieves 37.7% improvement in unseen classes compared with SSL methods, and 26.3% improvement in seen classes compared with NCD methods.

2 Related Work

Semi-Supervised Learning. SSL assumes all training and testing data are from seen classes, no matter labeled or unlabeled, and the goal is to classify unseen class unlabeled data into accurate categories. SSL has a long research history [22] and our paper is mainly related to deep SSL that introduces SSL techniques to deep neural networks and achieved significant advancement in recent years The mainstream of these can be broadly classified into entropy minimization methods [5, 6], consistency regularization methods [7–10], and holistic methods [11–15]. The existing SSL methods fail to address unseen classes compared with our works.

Open set Semi-Supervised Learning. Open set SSL relaxes the assumption of SSL and assumes that unlabeled training data could contain unseen classes. But they still assume testing data are all from seen classes, and the goal is to decrease the negative impact of unseen unlabeled training data to maintain safeness on seen classes. Many open set SSL methods have been proposed [18–20, 23, 21, 24–26], such as DS3L [18], which assign weights to unlabeled data based on a bi-level optimization, UASD [19], which filter unlabeled examples based on the prediction consistency, MTC [20], which adopts a multi-task curriculum learning framework to detect unseen classes

and classify seen classes simultaneously, OSSGAN [27], which propose a method quantifies the likelihood that a sample belongs to seen classes, T2T [24], which propose a novel cross-modal matching strategy to detect unseen classes. Although the above open set SSL methods are safe for seen classes, they can not classify unseen classes automatically.

Novel class discovery. NCD assumes training data consists of seen labeled and unseen unlabeled samples, and the goal is to classify both seen and unseen classes during the testing phase. The NCD problem is the first formally introduced in [28]. Recently, many NCD methods have been proposed based on a two-step training strategy [28–31, 16, 32, 33], i.e., a data embedding is learned on the labeled data using a metric learning technique, and then fine-tuned while learning the cluster assignments on the unlabeled data. These NCD methods can discover unseen classes automatically, however, they do not have the ability to classify seen classes accurately.

3 Preliminary and Background

In our study, the training data contains n labeled samples $\mathcal{D}_l = \{(\mathbf{x}_1, \mathbf{y}_1), \cdots, (\mathbf{x}_n, \mathbf{y}_n)\}$ and m unlabeled samples $\mathcal{D}_u = \{\mathbf{x}_{n+1}, \cdots, \mathbf{x}_{n+m}\}$, $\mathbf{x} \in \mathcal{X} \in \mathbb{R}^D, \mathbf{y} \in \mathcal{Y} = \{1, \cdots, C_L\}$ where D is the feature dimension and C_L is the number of seen classes. We use C_U to represent the total number of classes in unlabeled data, $C_L = C_U$ in previous SSL and $C_L \cap C_U = \emptyset$ in NCD. In our study, the number of seen classes $C_{seen} = C_L \cap C_U$ and the number of unseen classes $C_{unseen} = C_U \setminus C_L$. The goal is to learn a model $f(\mathbf{x};\theta) : \{\mathcal{X};\Theta\} \rightarrow \mathcal{Y}$ parameterized by $\theta \in \Theta$ from training data. Specifically, the $f(\mathbf{x};\theta)$ can be decomposed of an embedding model $g(\mathbf{x};\theta) : \mathbb{R}^D \rightarrow \mathbb{R}^N$ to learn a low-dimensional feature z and a classification model $h(\mathbf{z}) : \mathbb{R}^N \rightarrow \mathbb{R}^{C_{seen}+C_{unseen}}$.

The training loss of an SSL algorithm usually contains supervised loss $\mathcal{L}_{SUP}$ and semi-supervised loss $\mathcal{L}_{SSL}$ with a trade-off parameter $\lambda_u > 0$: $\mathcal{L}_{SUP} + \lambda_u \mathcal{L}_{SSL}$, where $\mathcal{L}_{SUP}$ is constructed on labeled data and $\mathcal{L}_{SSL}$ is constructed on both labeled and unlabeled data.

Typically, $\mathcal{L}_{SUP}$ applies the standard cross-entropy loss on labeled samples:

$$\mathcal{L}_{SUP} = \frac{1}{n} \sum_{i=1}^{n} H(\mathbf{y}_i, p(\mathbf{x})) \tag{1}$$

where $p(\mathbf{x}) = \text{Softmax}(f(\mathbf{x};\theta))$ is the predicted probabilities produced for the input $\mathbf{x}$, and $H(\cdot,\cdot)$ is the cross-entropy function.

Different constructions of the semi-supervised loss $\mathcal{L}_{SSL}$ lead to different SSL methods. Typically, there are two ways of constructing $\mathcal{L}_{SSL}$: one is to use pseudo-labels to formulate a "supervised loss" such as the cross-entropy loss, and another one is to optimize a regularization that does not depend on labels such as consistency regularization.

For example, FixMatch [11] adopts the pseudo-label loss which can be written as:

$$\mathcal{L}_{SSL} = \frac{1}{m} \sum_{i=n+1}^{n+m} I\left(\max\left(p(\text{aug}_w\left(\mathbf{x}\right))\right) \geq \tau\right) H\left(\widehat{\mathbf{y}}_i, p(\text{aug}_s\left(\mathbf{x}\right))\right) \tag{2}$$

where $\text{aug}_w(\mathbf{x})$ and $\text{aug}_s(\mathbf{x})$ indicate the weak and strong augmentation for an input $\mathbf{x}$, $\widehat{\mathbf{y}}_i = \arg\max p(\text{aug}_w\left(\mathbf{x}\right))$ is the pseudo-label, τ is the confidence threshold for pseudo-label selection, $I(\cdot)$ is the indicator function.

UDA [34] adopts the consistency regularization based which can be written as

$$\mathcal{L}_{SSL} = \frac{1}{m} \sum_{i=n+1}^{n+m} \|p(\text{aug}(\mathbf{x}_i)) - p(\text{aug}'(\mathbf{x}_i))\|_2^2 \tag{3}$$

where $\text{aug}(\mathbf{x}_i)$ and $\text{aug}'(\mathbf{x}_i)$ represents different augmentation strategies, such as rotations for images.

4 The SU-SSL Method

In this section, we propose an efficient SSL algorithm called SU-SSL to enable the previous SSL to classify unseen classes and maintain safeness on seen classes. The overall framework is illustrated

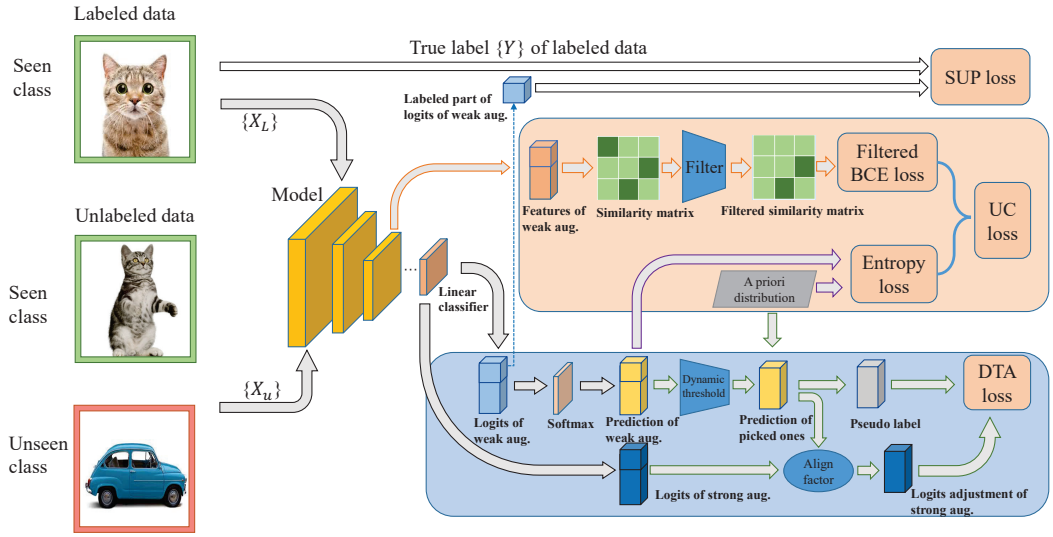


Figure 2: Framework of our proposed SU-SSL. The objective of SU-SSL can be decomposed into $\mathcal{L}_{SUP}$, $\mathcal{L}_{UC}$ and $\mathcal{L}_{DTA}$, $\mathcal{L}_{UC}$ is the module with the light yellow background box in the figure and $\mathcal{L}_{DTA}$ is the module with the light blue background box in the figure.

in Figure 2. SU-SSL consists of two main parts: a novel unseen class classification loss to discover unseen classes automatically and an adaptive threshold with distribution alignment to alleviate the different learning paces between seen classes and unseen classes. We first provide an overview of the overall objective. The concrete details of the objective are provided in the following contents.

4.1 SU-SSL: An Overview for Two Questions

Previous SSL methods do not have the ability to classify unseen classes, which leads to a large number of samples from unseen classes being misclassified into seen classes. To address this problem, two main challenges need to be considered.

The first one is how to automatically classify unseen classes during model training. We propose to cluster unlabeled samples using the pairwise objective [16], and then we adjust the results of the clustering using the known prior distribution. Specifically, we adopt the cosine similarity to find the most similar sample in a mini-batch for each sample as the positive pairs. To avoid the mismatched situation where a sample from seen classes and a sample from unseen classes are wrongly paired, we design a similarity-based filter to get rid of the appearance of such pairs.

The second is how to balance the difference in learning paces due to the difference in learning styles between seen classes and unseen classes in order to improve the classification accuracy of unseen classes while ensuring the classification accuracy of seen classes does not decrease. We propose a metric that measures the difference between the current model's learning of seen classes and unseen classes, and further, formulate this metric into an adaptive threshold with distribution alignment. Based on the adaptive threshold, the model can adjust the predicted probability adaptively to balance the learning differences between seen classes and unseen classes.

Moreover, the classification of unseen classes can be regarded as a clustering task since no labels are provided, therefore, reliable feature representations are essential. Benefiting from recent progress on self-supervised learning, we adopt a simple contrastive learning method SimCLR [35] to pre-train our backbone network on the whole dataset in an unsupervised fashion.

Overall, the objective of SU-SSL consists of three parts: (i) supervised loss $\mathcal{L}_{SUP}$ for seen labeled data; (ii) unseen classes discovery loss $\mathcal{L}_{UC}$ for seen and unseen unlabeled data classification; (iii) a novel semi-supervised $\mathcal{L}_{DTA}$ for assigning pseudo labels using the adaptive threshold to achieve better classification results on both seen and unseen classes unlabeled data:

$$\mathcal{L} = \mathcal{L}_{SUP} + \lambda_1 \mathcal{L}_{UC} + \lambda_2 \mathcal{L}_{DTA} \tag{4}$$

where λ_1 and λ_2 are trade-off hyper-parameters, which are all set to 1 in our study.

4.2 Unseen Classes Classification: $\mathcal{L}_{UC}$

To enable SSL the ability to classify unseen classes, we propose to use binary cross-entropy (BCE) loss which can utilize pairwise similarity to achieve the objective of unseen classes classification:

$$\begin{aligned}\mathcal{L}_{BCE} = -\frac{1}{(m+n)^2}\sum_{i=1}^{m+n}\sum_{j=1}^{m+n}\Big[s_{ij}\log p\left(\mathbf{x}_i\right)^\top p\left(\mathbf{x}_j\right) \\ + (1-s_{ij})\log\Big(1-p\left(\mathbf{x}_i\right)^\top p\left(\mathbf{x}_j\right)\Big)\Big]\end{aligned} \tag{5}$$

where s_{ij} is a measure of the degree of similarity between $\mathbf{x}_i$ and $\mathbf{x}_j$ (e.g., cosine similarity between $\mathbf{x}_i$ and $\mathbf{x}_j$). The first term of the equation is to pull two similar samples closer, while the latter term is to push the two dissimilar samples farther apart. BCE loss achieves clustering of unseen classes by pairing similar samples.

The BCE loss is commonly adopted in NCD studies, however, in our study, using BCE loss directly can not effectively classifying unseen classes because both seen classes and unseen classes appear in the unlabeled data. The main reasons are different learning pace between seen and unseen classes that results in two samples belonging to the same unseen class are likely to be pushed apart at the beginning of training due to their low similarity. To alleviate this issue, on one hand, we drop the push-apart strategy, on the other hand, we adopt the cosine similarity to find the most similar sample to be pulled together for each sample in a batch [36].

Meanwhile, to further improve the pair accuracy, we propose a filter strategy to address the wrong pairs that consist samples from seen and unseen classes. Specifically, for labeled data in a batch, we directly match pairs based on the ground truth label, for $\mathbf{x}$ in the unlabeled data, we first find the sample $\widetilde{\mathbf{x}}$ with the greatest similarity, then we have the following considerations: we compute the cosine similarity between the sample $\widetilde{\mathbf{x}}$ and the labeled data in the batch and sort them in descending order $\{\cos(g(\widetilde{\mathbf{x}}), g(\mathbf{x}_{label_1})), \cos(g(\widetilde{\mathbf{x}})), g(\mathbf{x}_{label_2})), ...\cos(g(\widetilde{\mathbf{x}}), g(\mathbf{x}_{label_n})\})$. We consider the fact that if x is a sample from unseen classes and $\widetilde{\mathbf{x}}$ is a sample from seen classes, then $\cos(g(\mathbf{x}), g(\widetilde{\mathbf{x}}))$ is likely to be less than some values in $\cos(g(\widetilde{\mathbf{x}}), g(\mathbf{x}_{label_i}))$. Based on this phenomenon, we set a threshold k, and only when sample $\mathbf{x}$ and sample $\widetilde{\mathbf{x}}$ satisfy $\cos(g(\mathbf{x}), g(\widetilde{\mathbf{x}})) \geq \cos(g(\widetilde{\mathbf{x}}), g(\mathbf{x}_{label_i}))$ we use this pair to calculate the BCE loss. The proposed filter-BCE loss ($\mathcal{L}_{FBCE}$) is formulated as follows:

$$\mathcal{L}_{FBCE} = -\frac{1}{n}\sum_{\mathbf{x}_i\in\mathcal{D}_l}\log\Big(p\left(\mathbf{x}_i\right)^\top p\left(\widetilde{\mathbf{x}}_i\right))\Big) - \frac{1}{m}\sum_{\mathbf{x}_j\in\mathcal{D}_u} I\left(\mathbf{x}_j\right)\log\Big(p\left(\mathbf{x}_j\right)^\top p\left(\widetilde{\mathbf{x}}_j\right))\Big) \tag{6}$$

$$I\left(\mathbf{x}_j\right) \equiv \cos(g(\mathbf{x}_j), g(\widetilde{\mathbf{x}}_j))) \geq \cos(g(\widetilde{\mathbf{x}}_j), g(\mathbf{x}_{label_k})) \tag{7}$$

To further prevent the model from classifying all unseen classes into one class and thus hindering the performance of unseen classes classification, we regularize the predictive distribution of the pseudo label using the known prior distribution:

$$\mathcal{L}_{ENT} = \mathrm{KL}\left(\frac{1}{m+n}\sum_{\mathbf{x}_i\in\mathcal{D}_l\cup\mathcal{D}_u} p(\mathbf{x}_i)\|\mathcal{P}(y)\right) \tag{8}$$

where $\mathcal{P}(y)$ represents the known prior distribution. The definition of $\mathcal{L}_{UC}$ is:

$$\mathcal{L}_{UC} = \mathcal{L}_{FBCE} + \mathcal{L}_{ENT} \tag{9}$$

4.3 SSL Loss with Adaptive Threshold: $\mathcal{L}_{DTA}$

To maintain safeness on seen classes, we propose to adapt the different learning statuses between seen and unseen classes. For the learning of seen classes in labeled data, the model can use cross-entropy loss to build up a mapping from features to labels directly. However, the model can only learn unseen classes from the pairwise objective. This results in the learning of unseen classes being slower than seen classes, so we come up with the idea of using pseudo labels to accelerate the learning of unseen classes. In previous SSL studies, e.g., Fixmatch [11], pseudo-labels are selected based on a fixed

confidence threshold but in our studies, it is unreasonable to use the same threshold for both seen classes and unseen classes due to the different learning paces, so we propose a dynamic threshold to solve this problem. We define a metric to measure the difference in the learning process between seen classes and unseen classes as follows. First, we calculate the maximum classification confidence and the corresponding pseudo-label for each sample:

$$\widehat{p_i} = \max\left(p(\text{aug}_w\left(\mathbf{x}\right))\right) \tag{10}$$

Then we define the metric $\mathbf{U}$ as follows:

$$\mathbf{U} = \left(\frac{1}{N_{seen}} \sum_{\mathbf{x}_i \in \mathcal{X}_{seen}} \widehat{p_i}\right) - \left(\frac{1}{N_{unseen}} \sum_{\mathbf{x}_j \in \mathcal{X}_{unseen}} \widehat{p_j}\right) \tag{11}$$

where N_{seen} and N_{unseen} are the total number of samples which are classified to seen classes and unseen classes, respectively. $\mathcal{X}_{seen}$ refers to samples with pseudo label belongs seen classes and $\mathcal{X}_{unseen}$ refers to samples with pseudo label belongs to unseen classes.

$\mathbf{U}$ can effectively assess the degree of learning difference between seen classes and unseen classes. We applied $\mathbf{U}$ to the threshold selection for unseen classes, where for seen classes we assume that the threshold is τ, then for unseen classes, we heuristically set the threshold to $\tau - \alpha\mathbf{U}$ where α is the trade-off parameters.

The above operation ensures that more samples predicted as unseen classes can be used in the model learning process. To further exploit the pseudo label, we refer to the distribution alignment in [37]. It should be noted that although $\mathcal{L}_{ENT}$ already takes into account distribution alignment, that is for all samples. Here we consider distribution alignment based on samples with confidence above the threshold (e.g., samples classified to be seen classes with confidence above τ and samples classified to be unseen classes with confidence above $\tau - \alpha\mathbf{U}$), because these samples are used for training, and a severe imbalance exists among them.

Our goal is to have the distribution of these above-threshold samples converge to a known prior distribution to better mining unseen classes, so we add distribution alignment as a fine-tuning for logits to $\mathcal{L}_{DTA}$. In Fixmtach, the main part of the unsupervised loss is $\mathcal{L}_{CE}\left(\widehat{\mathbf{y}}_i, f(\text{aug}_s\left(\mathbf{x}\right); \theta)\right)$, on which we want to add distribution alignment as an adjustment term to logit, we define:

$$\begin{aligned} \mathcal{P}(X_{select}) = &\sum_{\mathbf{x}_i \in \mathcal{X}_{seen}} I\left(\widehat{\mathbf{p}}_i \geq \tau\right) p(\text{aug}_w\left(\mathbf{x}_i\right)) \\ &+ \sum_{\mathbf{x}_j \in \mathcal{X}_{unseen}} I\left(\widehat{\mathbf{p}}_j \geq \tau - \alpha\mathbf{U}\right) p(\text{aug}_w\left(\mathbf{x}_j\right)) \end{aligned} \tag{12}$$

We then require $\mathcal{P}(X_{select})$ and prior distribution $\mathcal{P}(y)$ to be aligned:

$$F_{ali} = \log \mathcal{P}(X_{select})/\mathcal{P}(y) \tag{13}$$

The adjustment factor F_{ali} aims to align the distribution of selected data to a prior distribution, and we give the definition of $\mathcal{L}_{DTA}$:

$$\mathcal{L}_{DTA} = \sum_{\mathbf{x}_i \in \mathcal{X}_{seen} \cup \mathcal{X}_{unseen}} I\left(\widehat{\mathbf{p}}_i \geq \tau_i\right) H\left(\widehat{\mathbf{y}}_i, f(\text{aug}_s\left(\mathbf{x}_i\right); \theta) + F_{ali}\right) \tag{14}$$

where τ_i is τ for $\mathbf{x}_i$ belongs to $\mathcal{X}_{seen}$ and τ_i is $\tau - \alpha\mathbf{U}$ for $\mathbf{x}_i$ belongs to $\mathcal{X}_{unseen}$. It is important to note that dynamically adjusted thresholds and logit adjustment with distribution alignment factor are complementary: dynamic threshold adjustment is designed for preventing fixed high thresholds from hindering the learning of unseen classes. Thus, we dynamically adjust the threshold so that more samples of unseen classes could be learned by the model, which is beneficial to logit adjustment with distribution alignment. Distribution alignment as a factor for logit adjustment will also make the model less inclined towards seen classes and thus facilitate the learning of unseen classes.

5 Experiments

In this section, we give a comprehensive evaluation of SU-SSL. Experimental results and detailed analysis are reported to demonstrate the effectiveness of our proposal.

5.1 Experimental Setup

Datasets. We evaluate SU-SSL and compared methods on three benchmark datasets CIFAR-10, CIFAR-100 [38] and ImageNet [39]. Specifically, for the ImageNet dataset, 100 classes are sub-sampled following [40, 36]. We first divide classes into 50% seen and 50% unseen classes, then select 50% of seen classes as the labeled data, and the rest as unlabeled data.

Compared Methods. We compare SU-SSL with representative SSL, open-set SSL, and NCD methods. The SSL and open-set SSL methods are extended to be applicable to unseen classes in the following way: Samples are divided into known classes and unknown classes. We report their performance on seen classes and apply K-means clustering to unseen class samples to obtain clustering results. For SSL, the FixMatch [11] is adopted due to its empirical success and estimate unseen classes based on softmax confidence scores. For open-set SSL, we adopt two representative methods DS3L [18] which tries to assign lower weights to unseen classes unlabeled data, and CGDL [23] which automatically rejects unseen class samples. NCD methods are extended to classify seen classes by using the Hungarian algorithm [41] to match some of the discovered classes with classes in the labeled data. Specifically, two NCD methods are employed: DTC [28] and RankStats [16], which have been reported to achieve the state-of-the-art performance on NCD tasks. Moreover, we also compare SU-SSL with ORCA [36] methods, which also consider both seen classes and unseen classes classification.

All compared methods are implemented based on the pre-trained model using the contrastive learning algorithm SimCLR [35]. The only exception is DTC which has its own specialized pretraining procedure on labeled data [28].

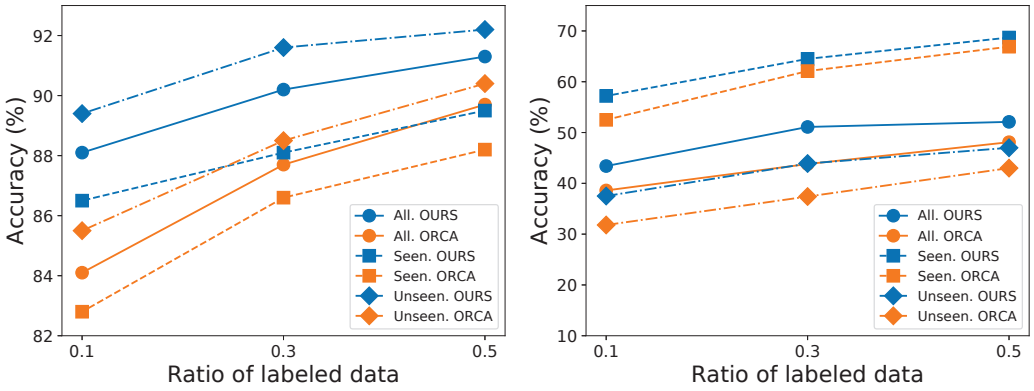
Table 1: Classification accuracy of compared methods on seen, unseen and all classes. The underline indicates the performance is worse than the baseline SSL methods.

Classes	**Dataset**	**SSL**	**Open-set SSL**		**NCD**			
		Fixmatch	**DS3L**	**CGDL**	**DTC**	**RankStats**	**ORCA**	**OURS**
Seen	CIFAR-10	71.5	77.6	72.3	<u>53.9</u>	86.6	88.2	**89.5**
	CIFAR-100	39.6	55.1	49.3	<u>31.3</u>	<u>36.4</u>	66.9	**68.7**
	ImageNet-100	65.8	71.2	67.3	<u>25.6</u>	<u>47.3</u>	89.1	**91.0**
	Average	59.0	68.0	63.0	36.9	56.8	81.4	**83.1**
Unseen	CIFAR-10	50.4	<u>45.3</u>	<u>44.6</u>	<u>39.5</u>	81.0	90.4	**92.2**
	CIFAR-100	23.5	23.7	<u>22.5</u>	<u>22.9</u>	28.4	43.0	**47.0**
	ImageNet-100	36.7	<u>32.5</u>	<u>33.8</u>	<u>20.8</u>	<u>28.7</u>	72.1	**75.5**
	Average	36.9	33.9	33.6	27.7	46.0	68.5	**71.6**
All	CIFAR-10	49.5	<u>40.2</u>	<u>39.7</u>	<u>38.3</u>	82.9	89.7	**91.3**
	CIFAR-100	20.3	24.0	23.5	<u>18.3</u>	23.1	48.1	**52.1**
	ImageNet-100	34.9	<u>30.8</u>	<u>31.9</u>	<u>21.3</u>	40.3	77.8	**79.6**
	Average	34.9	31.7	31.7	26.0	48.8	71.9	**74.3**

5.2 Main Results

The mean classification accuracy on CIFAR-10, CIFAR-100, and ImageNet-100 dataset are provided in Table 1. From the results, it can be observed that open-set SSL methods can address the performance degradation problem on seen classes, but can not classify unseen classes accurately. NCD methods, e.g., RankStats can improve the unseen classification performance, but suffer unsafe problem on seen classes, DTC performs even worse than SSL on unseen class classification. On the contrary, our proposal SU-SSL can not only classify unseen classes accurately but also maintain safeness on seen classes. For example, SU-SSLachieves a 24.1% improvement on seen classes and 34.7% on unseen classes compared with the FixMatch methods. Compared with ORCA, SU-SSL also achieves a significant performance improvement in both seen and unseen classes.

To further demonstrate the effectiveness of our proposal with varying label sizes, we evaluate the performance of SU-SSL and ORCA with different numbers of labeled data, as shown in Figure 3.



(a) Classification accuracy on CIFAR-10. (b) Classification accuracy on CIFAR-100.

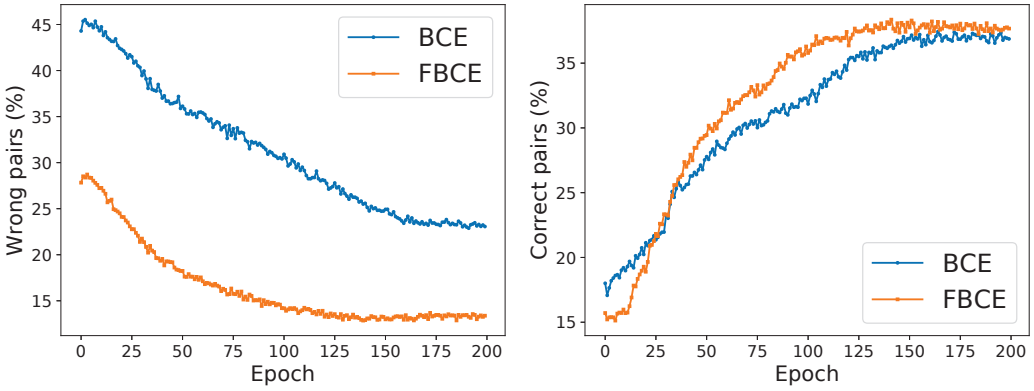
Figure 3: Performance of SU-SSL and ORCA with different numbers of labeled data.

The shows show that the performances of SU-SSL are always better than ORCA in all cases with a significant margin. Moreover, the results also demonstrate that our proposal SU-SSL is robust with label size, for example, even with only 10% of labeled samples, the unseen-class accuracy of our method can still reach more than 89.4% and 37.5%.

5.3 Detail Analysis

In this subsection, detailed analyses are shown to help understand the superiority of our proposal, including analyses of the two modules: $\mathcal{L}_{UC}$ and $\mathcal{L}_{DTA}$, and parameter sensitivity analysis.

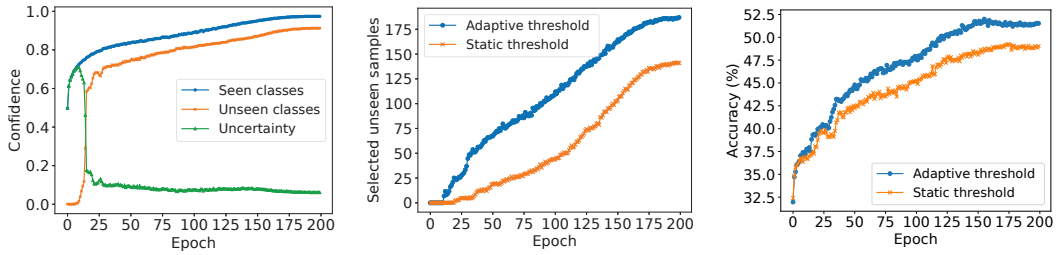
Analysis of Unseen-class Classification Loss: $\mathcal{L}_{UC}$. We first show the effectiveness of the proposed FBCE loss by proposing a basic model (BM) with $C_{seen} + C_{unseen}$ classification heads and optimize the model by minimizing the simple BCE loss and our proposed FBCE loss separately. The comparison results are reported in Figure 4, including misclassified pairs and correct classified pairs. From the results, we can see that the proposed FBCE loss can effectively decrease the ratio of misclassified seen-unseen pairs. Then, the ablation studied of $\mathcal{L}_{UC}$ are reported in Table 2. The results show that the full $\mathcal{L}_{UC}$ achieves the best performance on both seen and unseen classification.



(a) Ratio of seen-unseen pairs (b) Correctly selected unseen-unseen pairs

Figure 4: (a) Ratio of seen-unseen pairs changes during training; (b) Number of unseen-unseen pairs chosen correctly in a batch. Both the above results are implemented on CIFAR-100.

Analysis on Dynamic Threshold: $\mathcal{L}_{DTA}$. We first compare the performance between our proposed adaptive threshold and the static threshold (e.g., 0.95 in FixMatch), the results are reported in Figure 5. From Figure 5a, we can see that the prediction confidence between seen and unseen classes are significantly different, thus, it is not proper to adopt a static threshold. Figure 5b and Figure 5c show



(a) Learning difference (b) Selected unseen classes samples (c) Pseudo-label accuracy of unseen

Figure 5: (a) learning difference between seen classes and unseen classes during training; (b) the number of pseudo-labels for unseen classes; (c) the accuracy of pseudo-labels for unseen classes. All of the above results are implemented on CIFAR-100.

that the proposed adaptive threshold can improve the accuracy of pseudo-label assignment in the training process significantly. Results in Table 3 give a more clear ablation study to demonstrate the effectiveness of our proposed $\mathcal{L}_{DTA}$ loss.

Table 2: Analysis of $\mathcal{L}_{UC}$: classification accuracy on CIFAR-100.

Method	Seen	Unseen	all
FixMatch	39.6	23.5	20.3
BM + $\mathcal{L}_{BCE}$	72.8	28.3	31.5
BM + $\mathcal{L}_{BCE}$ + $\mathcal{L}_{ENT}$	67.2	44.2	50.7
BM + $\mathcal{L}_{UC}$ (UC Model)	68.2	44.3	50.2

Table 3: Analysis of $\mathcal{L}_{DTA}$: classification accuracy on CIFAR-100.

Method	Seen	Unseen	all
UC Model	68.2	44.3	50.2
UC Model + DA	68.6	44.8	50.3
UC Model + DT	69.3	45.6	49.6
UC Model + DTA	68.7	47.0	52.1

5.4 Parameter Sensitivity Analysis

Evaluating different k used in $\mathcal{L}_{FBCE}$. The intention of $\mathcal{L}_{FBCE}$ is to filter mismatched pairs containing samples from seen classes and unseen classes. We show the performance with different k in Figure 6, and the results show that we $k = 2$ the proposal achieves the best performance on all classes, and the performance does not degrade significantly with k changes. This demonstrates that our proposal is quite robust with the selection of k.

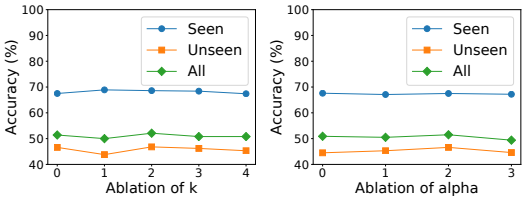


Figure 6: Classification accuracy on CIFAR-100.

Evaluating different α used in $\mathcal{L}_{DTA}$. We further analysis the impact of the dynamic threshold coefficient α. It can be shown that our method is not sensitive to the selection of α and when $\alpha = 2$ our proposal achieves the best result.

6 Conclusion

In this paper, we tackle an important problem of SSL, that is, SSL with the ability to classify unseen classes. We propose a novel SU-SSL approach that consists of two important modules: unseen discovery and adaptive re-balance. We propose a novel unseen class classification objective that can exploit pairwise similarity and eliminate noisy pairs, and a novel semi-supervised objective that adopt an adaptive threshold with distribution alignment to improve the performance on both seen and unseen classes. Extensive experiments clearly show the effectiveness of our proposal.

How to classify unseen classes with no labeled data is an important problem in SSL. Our work puts a promising scheme in this direction. One limitation of our scheme is it does not have theoretical guarantees. We will put efforts into this direction in future work, such as giving generalization risk analysis on unseen classes.

References

[1] Yann LeCun, Bengio Yoshua, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

[2] Zhi-Hua Zhou. A brief introduction to weakly supervised learning. *National Science Review*, 5 (1):44–53, 2017.

[3] Yu-Feng Li, Lan-Zhe Guo, and Zhi-Hua Zhou. Towards safe weakly supervised learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(1):334–346, 2019.

[4] Xiaojin Zhu and Andrew B Goldberg. Introduction to semi-supervised learning. *Synthesis lectures on artificial intelligence and machine learning*, 3(1):1–130, 2009.

[5] Dong-Hyun Lee. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, volume 3, page 896, 2013.

[6] Yves Grandvalet and Yoshua Bengio. Semi-supervised learning by entropy minimization. In *Advances in Neural Information Processing Systems*, pages 529–536, 2004.

[7] Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, and Shin Ishii. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(8):1979–1993, 2019.

[8] Mehdi Sajjadi, Mehran Javanmardi, and Tolga Tasdizen. Regularization with stochastic transformations and perturbations for deep semi-supervised learning. In *Advances in Neural Information Processing Systems*, pages 1163–1171, 2016.

[9] Samuli Laine and Timo Aila. Temporal ensembling for semi-supervised learning. In *Proceedings of the 5th International Conference on Learning Representations*, 2017.

[10] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *Advances in Neural Information Processing Systems*, pages 1195–1204, 2017.

[11] Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin A Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and ChunLiang Li. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. In *Advances in Neural Information Processing Systems*, pages 596–608, 2020.

[12] David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin A Raffel. Mixmatch: A holistic approach to semi-supervised learning. In *Advances in Neural Information Processing Systems*, pages 5050–5060, 2019.

[13] David Berthelot, Nicholas Carlini, Ekin D Cubuk, Alex Kurakin, Kihyuk Sohn, Han Zhang, and Colin Raffel. Remixmatch: Semi-supervised learning with distribution alignment and augmentation anchoring. In *Proceedings of the 8th International Conference on Learning Representations*, 2020.

[14] Yi Xu, Lei Shang, Jinxing Ye, Qi Qian, YuFeng Li, Baigui Sun, Hao Li, and Rong Jin. Dash: Semi-supervised learning with dynamic thresholding. In *Proceedings of the 38th International Conference on Machine Learning*, pages 11525–11536, 2021.

[15] Bowen Zhang, Yidong Wang, Wenxin Hou, Hao Wu, Jindong Wang, Manabu Okumura, and Takahiro Shinozaki. Flexmatch: Boosting semi-supervised learning with curriculum pseudo labeling. In *Advances in Neural Information Processing Systems*, pages 18408–18419, 2021.

[16] Kai Han, Sylvestre-Alvise Rebuffi, Sebastien Ehrhardt, Andrea Vedaldi, and Andrew Zisserman. Automatically discovering and learning new visual categories with ranking statistics. In *Proceedings of the 8th International Conference on Learning Representations*, 2020.

[17] Avital Oliver, Augustus Odena, Colin A Raffel, Ekin Dogus Cubuk, and Ian Goodfellow. Realistic evaluation of deep semi-supervised learning algorithms. In *Advances in neural information processing systems*, pages 3239–3250, 2018.

[18] LanZhe Guo, ZhenYu Zhang, Yuan Jiang, YuFeng Li, and ZhiHua Zhou. Safe deep semi-supervised learning for unseen-class unlabeled data. In *Proceedings of the 37th International Conference on Machine Learning*, pages 3897–3906, 2020.

[19] Yanbei Chen, Xiatian Zhu, Wei Li, and Shaogang Gong. Semi-supervised learning under class distribution mismatch. In *The 34th AAAI Conference on Artificial Intelligence*, pages 3569–3576, 2020.

[20] Qing Yu, Daiki Ikami, Go Irie, and Kiyoharu Aizawa. Multi-task curriculum framework for open-set semi-supervised learning. In *Proceedings of the European Conference on Computer Vision*, pages 438–454, 2020.

[21] Kuniaki Saito, Donghyun Kim, and Kate Saenko. Openmatch: Open-set semi-supervised learning with open-set consistency regularization. In *Advances in Neural Information Processing Systems*, pages 25956–25967, 2021.

[22] Olivier Chapelle, Bernhard Scholkopf, and Alexander Zien. *Semi-supervised learning*. MIT Press, 2006.

[23] Xin Sun, Zhenning Yang, Chi Zhang, Keck-Voon Ling, and Guohao Peng. Conditional gaussian distribution learning for open set recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13480–13489, 2020.

[24] Junkai Huang, Chaowei Fang, Weikai Chen, Zhenhua Chai, Xiaolin Wei, Pengxu Wei, Liang Lin, and Guanbin Li. Trash to treasure: Harvesting ood data with cross-modal matching for open-set semi-supervised learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8310–8319, 2021.

[25] Alex Yuxuan Peng, Yun Sing Koh, Patricia Riddle, and Bernhard Pfahringer. Investigating the effect of novel classes in semi-supervised learning. In *Proceedings of the 11th Asian Conference on Machine Learning*, pages 615–630, 2019.

[26] Zhuo Huang, Chao Xue, Bo Han, Jian Yang, and Chen Gong. Universal semi-supervised learning. In *Advances in Neural Information Processing Systems*, 2021.

[27] Kai Katsumata, Duc Minh Vo, and Hideki Nakayama. OSSGAN: open-set semi-supervised image generation. *CoRR*, abs/2204.14249, 2022.

[28] Kai Han, Andrea Vedaldi, and Andrew Zisserman. Learning to discover novel visual categories via deep transfer clustering. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8401–8409, 2019.

[29] Yen-Chang Hsu, Zhaoyang Lv, and Zsolt Kira. Learning to cluster in order to transfer across domains and tasks. In *Proceedings of the 6th International Conference on Learning Representations*, 2018.

[30] Yen-Chang Hsu, Zhaoyang Lv, Joel Schlosser, Phillip Odom, and Zsolt Kira. Multi-class classification without multi-class labels. In *Proceedings of the 7th International Conference on Learning Representations*, 2019.

[31] Enrico Fini, Enver Sangineto, Stéphane Lathuilière, Zhun Zhong, Moin Nabi, and Elisa Ricci. A unified objective for novel class discovery. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9264–9272, 2021.

[32] Zhun Zhong, Linchao Zhu, Zhiming Luo, Shaozi Li, Yi Yang, and Nicu Sebe. Openmix: Reviving known knowledge for discovering novel visual categories in an open world. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9462–9470, 2021.

[33] Zhun Zhong, Enrico Fini, Subhankar Roy, Zhiming Luo, Elisa Ricci, and Nicu Sebe. Neighborhood contrastive learning for novel class discovery. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 10867–10875, 2021.

[34] Qizhe Xie, Zihang Dai, Eduard H. Hovy, Thang Luong, and Quoc Le. Unsupervised data augmentation for consistency training. In *Advances in Neural Information Processing Systems*, pages 6256–6268, 2020.

[35] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning*, pages 1597–1607, 2020.

[36] Kaidi Cao, Maria Brbic, and Jure Leskovec. Open-world semi-supervised learning. In *Proceedings of the 10th International Conference on Learning Representations*, 2022.

[37] Justin Lazarow, Kihyuk Sohn, ChunLiang Li, Zizhao Zhang, ChenYu Lee, and Tomas Pfister. Unifying distribution alignment as a loss for imbalanced semi-supervised learning. 2022.

[38] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. *Tech Report*, 2009.

[39] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

[40] Wouter Van Gansbeke, Simon Vandenhende, Stamatios Georgoulis, Marc Proesmans, and Luc Van Gool. Scan: Learning to classify images without labels. In *Proceedings of the European Conference on Computer Vision*, pages 268–285, 2020.

[41] Harold W Kuhn. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1-2):83–97, 1955.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See Section 5, we performed the parametric sensitivity analysis.
   (c) Did you discuss any potential negative societal impacts of your work? [N/A]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [No] We will release the code later.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Section 5 and supplementary materials.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] See Section 5, we report all the results over three runs.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See supplementary materials.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] See Section 5.
   (b) Did you mention the license of the assets? [Yes] See Section 5.
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]

(d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [Yes] See Section 5.

(e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]

5. If you used crowdsourcing or conducted research with human subjects...

(a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]

(b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]