Debiasing, calibrating, and improving Semi-supervised Learning performance via simple Ensemble Projector

Khanh-Binh Nguyen
Sungkyunkwan University
South Korea
n.k.binh00@gmail.com

Abstract

Recent studies on semi-supervised learning (SSL) have achieved great success. Despite their promising performance, current state-of-the-art methods tend toward increasingly complex designs at the cost of introducing more network components and additional training procedures. In this paper, we propose a simple method named Ensemble Projectors Aided for Semi-supervised Learning (EPASS), which focuses mainly on improving the learned embeddings to boost the performance of the existing contrastive joint-training semi-supervised learning frameworks. Unlike standard methods, where the learned embeddings from one projector are stored in memory banks to be used with contrastive learning, EPASS stores the ensemble embeddings from multiple projectors in memory banks. As a result, EPASS improves generalization, strengthens feature representation, and boosts performance. For instance, EPASS improves strong baselines for semi-supervised learning by 39.47%/31.39%/24.70% top-1 error rate, while using only 100k/1%/10% of labeled data for SimMatch, and achieves 40.24%/32.64%/25.90% top-1 error rate for CoMatch on the ImageNet dataset. These improvements are consistent across methods, network architectures, and datasets, proving the general effectiveness of the proposed methods. Code is available at https://github.com/beandkay/EPASS.

1. Introduction

Deep learning has shown remarkable success in a variety of visual tasks such as image classification [19], speech recognition [1], and natural language processing [34]. This success benefits from the availability of large-scale annotated datasets [20, 21, 27, 31, 32]. Large amounts of annotations are expensive or time-consuming in real-world domains such as medical imaging, banking, and finance. Learning without annotations or with a small number of annotations has become an essential problem in computer vision, as demonstrated by [5, 6, 10–12, 17, 18, 23, 24, 26, 35, 36, 42, 46, 51].

Contrastive self-supervised learning (CSL) is based on instance discrimination, which attracts positive samples while repelling negative ones to learn the representation [10, 18, 40]. Inspired by CSL, contrastive joint-training SSL methods such as CoMatch [26] and SimMatch [51] leverage the idea of a memory bank and momentum encoder from MoCo [18] to support representational learning. In the current mainstream contrastive joint-training SSL methods, a multi-layer perceptron (MLP) is added after the encoder to obtain a low-dimensional embedding. Training loss and accuracy evaluation are both performed on this embedding. The previously learned embeddings from a low-dimensional projector are stored in a memory bank. These embeddings are later used in the contrastive learning phase to aid the learning process and improve the exponential moving average (EMA) teacher [36]. Although previous approaches demonstrate their novelty with state-of-the-art benchmarks across many datasets, there are still concerns that need to be considered. For instance, conventional methods such as CoMatch [26] and SimMatch [51] are based
on the assumption that the learned embeddings are correct, regardless of confirmation bias. This assumption is directly adopted from CSL; however, in a joint-training scheme, the easy-to-learn representation could easily dominate the hard-to-learn representation, leading to biased distributions and embeddings. This would become even worse when confirmation bias happens and the embeddings are driven away by the incorrect pseudo-labels. As a result, the embeddings stored in the memory bank are also affected, causing the confirmation bias issue and the erroneous EMA teacher.

The confirmation bias could be seen in Figure 1, where CoMatch only has 80.56% correctness for pseudo-labels and SimMatch has 90.61% correctness for pseudo-labels. When the embedding bias happens at the instance level and the confirmation bias happens at the semantic level, they degrade the performance of the EMA teacher. As a result, the well-learned embeddings at the instance level could be driven away by the confirmation bias at the semantic level during backward propagation, and vice versa.

<table>
<thead>
<tr>
<th>Method</th>
<th>WRN-28-2</th>
<th>WRN-28-8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original [6, 7, 35, 38, 43, 47]</td>
<td>1.4 M</td>
<td>23.4 M</td>
</tr>
<tr>
<td>Chen et al. [9]</td>
<td>3.7 M (+ 2.3)</td>
<td>19.9 M (*, + 3.5)</td>
</tr>
<tr>
<td>CoMatch [26]</td>
<td>1.5 M</td>
<td>23.71 M</td>
</tr>
<tr>
<td>SimMatch [51]</td>
<td>1.5 M</td>
<td>23.74 M</td>
</tr>
<tr>
<td>CoMatch [26] + EPASS (3 projs)</td>
<td>1.54 M (+ 0.04)</td>
<td>24.30 M (+ 0.59)</td>
</tr>
<tr>
<td>SimMatch [51] + EPASS (3 projs)</td>
<td>1.56 M (+ 0.06)</td>
<td>24.39 M (+ 0.63)</td>
</tr>
</tbody>
</table>

Table 1. Comparison with multi-head co-training. ‘*’ indicates different architecture as Chen et al. [9] modified the number of channels of the final block from 512 to 256.

Figure 2. Training phase for contrastive joint-training SSL without/with the proposed EPASS. 2a represents the conventional training phase without EPASS [26, 51]. Unlike 2a, in 2b, instead of using only one projector to learn the embeddings, EPASS uses multiple projectors to ensemble the embeddings, which is less biased and more generalized.

To address these limitations, we propose Ensemble Projectors Aided for Semi-supervised Learning (EPASS), a plug-and-play module to strengthen the EMA teacher as well as to improve the generalization of the learned embeddings, as illustrated in Figure 2. Adding a projector helps mitigate the overfitting problem, and the generated features are more distinguishable for classification [26, 51]. Chen et al. [13] proves the strengths of ensemble projectors in teacher-student frameworks via knowledge distillation. Therefore, we leverage those strengths with SSL, especially contrastive joint-training frameworks. Although there has been study about ensemble for SSL [9], they only discover it in the classification head, thus resulting in a large number of parameter overheads as shown in Table 1. Unlike [9], we specifically enrich the learned embeddings from the model by employing multiple projectors rather than only one, as it is common in conventional methods. Using ensemble projectors in contrastive learning, where multiple projectors are used instead of a single one, may improve the performance and robustness of the learned representations. By using multiple projectors, the model can learn different feature representations from different perspectives, which can be combined to produce more informative representations of the data. Additionally, using ensemble projectors can help to improve the generalization performance of the model, by reducing the risk of overfitting to the specific characteristics of a single projector.

Using ensemble projectors can also increase the robustness of the model against variations in the data distribution, as the multiple projectors can learn different features that are less sensitive to changes in the data distribution. This can be especially useful in situations where the data distribution is not well-defined or changes over time. Therefore, the embeddings of the model would be the ensemble ones, which are less biased and more robust than conventional methods. Our comprehensive results show that such a simple ensemble design brings a sweet spot between model performance and efficiency.

By incorporating the ensemble projectors in a contrastive-based SSL fashion, the proposed EPASS makes better use of embeddings to aid contrastive learning as well as to improve the classification performance simultaneously. In addition, ensemble multiple projectors introduce a relatively smaller number of parameters compared with ensemble multiple classification heads. Extensive experiments justify the effectiveness of EPASS, which produces a less biased feature space. Specifically, EPASS achieves a state-of-the-art performance with 39.47%/31.39%/24.70% top-1 error rate, while using only 100k/1%/10% of labeled data for SimMatch; and achieves 40.24%/32.64%/25.90% top-1 error rate for CoMatch on ImageNet dataset.

The contributions of this paper are summarized as follows:

- We hypothesize that the conventional contrastive joint-training SSL frameworks are sub-optimal since the multi-objective learning could harm the learned em-
beddings when confirmation bias occurs.

- We propose EPASS, a simple plug-and-play module that improves a generalization of the model by imposing the ensemble of multiple projectors, which encourages the model to produce less biased embeddings.

- To the best of our knowledge, this is the first work to enhance the performance of contrastive joint-training SSL methods by considering the embedding bias.

- Extensive experiments on many benchmark datasets demonstrate that EPASS consistently improves the performance of contrastive joint-training methods.

2. Related Work

2.1. Semi-supervised Learning

Semi-supervised learning is an essential method to leverage a large amount of unlabeled data to enhance the training process. Pseudo-label (PSL) [24] is the pioneer of nowadays popular methods, including self-training-based or consistency-based SSL approaches. In the pseudo-label-based methods, the model is first trained on a small amount of labeled data. Then, the model is used to make predictions for unlabeled data. The unlabeled data and their corresponding pseudo-labels are then used to train the model simultaneously with labeled data, forming the self-training-based methods [2, 3, 24, 28, 36, 42, 49]. Consistency-based methods [5, 6, 26, 29, 35, 47, 51] use a high threshold to determine the reliable predictions from weakly augmented samples. Then, they will be used as pseudo-labels for strongly augmented examples, and the low-confidence predictions will be discarded. However, those approaches suffer from confirmation bias [2] since they overfit the incorrect pseudo-labels during training. Moreover, methods using the high confirmation bias only use a small amount of labels during training. FlexMatch [47] introduces a Curriculum Pseudo Labeling (CPL) method based on the Curriculum Learning (CL) [4]. CPL configures a dynamic threshold for each class after each iteration, thus letting the model learn better for either hard-to-learn or easy-to-learn classes.

2.2. Contrastive joint-training SSL

Li et al. proposes CoMatch [26], which combines two contrastive representations on unlabeled data. However, CoMatch is extremely sensitive to the hyperparameter setting. Especially during training, CoMatch requires a large memory bank to store the embedded features. Recently, Zheng et al. [51] published work that takes semantic similarity and instance similarity into account during training. It shows that forcing consistency at both the semantic level and the instance level can bring an improvement, thus achieving state-of-the-art benchmarks. Along this line of work, [44, 50] also leverage the benefit of Class-aware Contrastive loss to the training process of SSL.

Previous methods might fail to provide the correct embeddings due to confirmation bias. Conventionally, confirmation bias does not exist in CSL; however, it occurs in contrastive joint-training SSL by the use of a threshold. It leads to the degradation of the classifier and the projector, thus providing incorrect predictions and embeddings. Our EPASS exploits the ensemble strategy for multiple projectors, imposing consistency and improving generalization for the learned embeddings, thus enhancing the correctness of model predictions.

3. Method

3.1. Preliminaries

We first define notations used in the following sections. For semi-supervised image classification problem, let \( X = \{ (x_b, y_b) : b \in \{1, \ldots, B\} \} \) be a batch of \( B \) labeled examples, where \( x_b \) is training examples and \( y_b \) is one-hot labels, and \( U = \{ u_b : b \in \{ 1, \ldots, \mu B \} \} \) be a batch of \( \mu B \) unlabeled examples where \( \mu \) is a hyper-parameter determining the relative sizes of \( X \) and \( U \). For labeled samples, we apply weak augmentation \( (A_w) \) to obtain the weakly augmented samples. Then, an encoder \( f (\cdot) \) and a fully-connected classifier \( h (\cdot) \) are applied to get the distribution over classes as \( p (y \mid x) = h (f (x)) \). The supervised cross-entropy loss for labeled samples is defined as:

\[
\mathcal{L}_s = \frac{1}{B} \sum_{b=1}^{B} \mathcal{H} (y_b, p_b) \tag{1}
\]

where \( \mathcal{H} \) is a standard cross-entropy loss function.

Conventionally, CoMatch and SimMatch apply a weak \( (A_w) \) and strong \( (A_s) \) augmentation on unlabeled samples, then use the trained encoder and fully-connected classifier to get the predictions as \( p_b^w = p (y \mid A_w (u_b)) \) and \( p_b^s = p (y \mid A_s (u_b)) \). Following CoMatch [26] and SimMatch [51], the predictions that surpassing confidence threshold \( \tau \) would be directly used as pseudo-labels to compute the unsupervised classification loss as:

\[
\mathcal{L}_u = \frac{1}{\mu B} \sum_{b=1}^{\mu B} \mathbb{1} (\max (p_b^w) \geq \tau) \mathcal{H} (p_b^w, p_b^s) \tag{2}
\]
where $\hat{y}_b^w = DA(p_b^w)$ is the pseudo-label for input $A_w(u_b)$ and $DA$ is the distribution alignment strategy [26, 51] to balance the pseudo-labels distribution.

Besides, a non-linear projector head $g(\cdot)$ is used to map the representation from encoder $f(\cdot)$ into a low-dimensional embeddings space $z = g \circ f$. The embeddings then are used to compute contrastive loss, which we simplify as the cross-entropy between the two normalized graphs:

$$L_c = \frac{1}{\mu B} \sum_{b=1}^{\mu B} H(q_b^w, q_b^s)$$

where $q = \phi(L_2 - \text{norm}(z))$ is the result after the transformation $\phi(\cdot)$ of CoMatch or SimMatch on the $L_2$ normalized vector. In CoMatch, $\phi$ is building a pseudo-label graph to guide the representation learning as described in Section 3.2.1. For SimMatch, $\phi$ is calculating the similarities between $z^w$ and $i-th$ instance by using a similarity function $\text{sim}(\cdot)$, which represents the dot product between $L_2$ normalized vectors $\text{sim}(u, v) = u^T v / \|u\|\|v\|$. As a result, we have:

$$q_i^w = \frac{\exp(\text{sim}(z_k^w, z_i)/T)}{\sum_{k=1}^{K} \exp(\text{sim}(z_k^w, z^k)/T)}$$
$$q_i^s = \frac{\exp(\text{sim}(z_k^s, z_i)/T)}{\sum_{k=1}^{K} \exp(\text{sim}(z_k^s, z^k)/T)}$$

The momentum embeddings stored in the memory bank and the EMA model are then defined as:

$$z_t \leftarrow mz_{t-1} + (1-m)z_t; \quad \theta_t \leftarrow m\theta_{t-1} + (1-m)\theta_t$$

where $z$ is the embeddings, $\theta$ is the model’s parameters, $t$ is the iteration, and $m$ is the momentum parameter. The overall training objective is:

$$L = L_s + \lambda_u L_u + \lambda_c L_c$$

### 3.2. EPASS

We propose a simple yet effective method to boost the performance of the conventional contrastive-based SSL that maximizes the correctness of the embeddings from different projections by using the ensemble technique.

Unlike conventional methods such as CoMatch and SimMatch, which assume that the learned embeddings from one projector are absolutely correct, we propose using the ensemble embeddings from multiple projectors to mitigate the bias. While there could be diverse options to combine multiple embeddings (e.g., concatenation, summation), we empirically found that simply averaging the selected embeddings works reasonably well and is computationally efficient. As each projector is randomly initialized, it provides a different view of inputs, which benefits the generalization of the model. This intuition is similar to that of multi-view learning. However, since we generate views with multiple projectors instead of creating multiple augmented samples, we introduce far less overhead to the pipeline. The ensemble of multiple projectors helps mitigate the bias in the early stages of training. In the joint-training scheme, the correct learned embeddings help improve the performance of the classification head and vice versa, thus reducing the confirmation bias effect. The embeddings stored in the memory bank by Equation 6 therefore are updated as:

$$z_t \leftarrow mz_{t-1} + (1-m)z_t; \quad \bar{z}_t = \frac{\sum_{p=1}^{P} z_{t,p}}{P}$$

where $P$ is the number of projectors.

#### 3.2.1 Application

**SimMatch:** Using our ensemble embeddings, we re-define instance similarity in SimMatch [51] and rewrite the Equation 4 as:

$$\bar{q}_i^w = \frac{\exp(\text{sim}(\bar{z}_k^w, \bar{z}_i)/T)}{\sum_{k=1}^{K} \exp(\text{sim}(\bar{z}_h^w, \bar{z}_k)/T)}$$

where $T$ is the temperature parameter controlling the sharpness of the distribution, $K$ is the number of weakly augmented embeddings, and $i$ represents the $i-th$ instance. Similarly, we can compute $\bar{q}_j^s$ by calculating the similarities between the strongly augmented embeddings $\bar{z}^s$ and $\bar{z}_i$.

$$\bar{q}_j^s = \frac{\exp(\text{sim}(\bar{z}_k^s, \bar{z}_i)/T)}{\sum_{k=1}^{K} \exp(\text{sim}(\bar{z}_h^s, \bar{z}_k)/T)}$$

The Equation 3 then is rewritten as:

$$L_c = \frac{1}{\mu B} \sum_{b=1}^{\mu B} H(\bar{q}_b^w, \bar{q}_b^s)$$

**CoMatch:** In CoMatch, the embeddings are used to construct a pseudo-label graph that defines the similarity of samples in the label space. Specifically, the instance similarity is also calculated as Equation 8 for weakly augmented samples. Then, a similarity matrix $W^q$ is constructed as:

$$W_{b_j}^q = \begin{cases} 1 & \text{if } b = j \\ \bar{q}_b \cdot \bar{q}_j & \text{if } b \neq j \text{ and } \bar{q}_b \cdot \bar{q}_j \geq \tau_c \\ 0 & \text{otherwise} \end{cases}$$

where $\tau_c$ indicates the similarity threshold. Also, an embedding graph $W^z$ is derived as:

$$W_{b_j}^z = \begin{cases} \exp(\bar{z}_b \cdot \bar{z}_j/t) & \text{if } b = j \\ \exp(\bar{z}_b \cdot \bar{z}_j/t) & \text{if } b \neq j \end{cases}$$
where $z_b = g \circ f (A_s (u_b))$ and $z_b' = g \circ f (A'_s (u_b))$. The Equation 3 then is rewritten as:

$$L_c = \frac{1}{\mu B} \sum_{b=1}^{\mu B} \mathcal{H} \left( \hat{W}_b^q, \hat{W}_b^s \right) \quad (13)$$

where $\hat{W}_{bj} = \frac{W_{bj}}{\sum_j W_{bj}}$, $\mathcal{H} \left( \hat{W}_b^q, \hat{W}_b^s \right)$ can be decomposed into:

$$\mathcal{H} \left( \hat{W}_b^q, \hat{W}_b^s \right) = - \hat{W}_{bb}^q \log \left( \frac{\exp (\bar{z}_b \cdot \bar{z}_b')/T)}{\sum_j \hat{W}_{bj}^q} \right)$$

$$- \sum_{j=1, j \neq b}^{\mu B} \hat{W}_{bj}^q \log \left( \frac{\exp (\bar{z}_b \cdot \bar{z}_j)/T)}{\sum_{j=1}^{\mu B} \hat{W}_{bj}^q} \right)$$

4. Experiments

4.1. Implementation Details

We evaluate EPASS on common benchmarks: CIFAR-10/100 [22], SVHN [30], STL-10 [14], and ImageNet [15]. We conduct experiments with varying amounts of labeled data, using previous work [5, 6, 26, 29, 35, 41, 43, 47, 51].

For a fair comparison, we train and evaluate all methods using the unified code base USB [37] with the same backbones and hyperparameters. We use Wide ResNet-28-2 [45] for CIFAR-10, Wide ResNet-28-8 for CIFAR-100, Wide ResNet-37-2 [52] for STL-10, and ResNet-50 [19] for ImageNet. We use SGD with a momentum of 0.9 as an optimizer. The initial learning rate is 0.03 with a cosine learning rate decay schedule of $\eta = \eta_0 \cos \left( \frac{\pi k}{K} \right)$, where $\eta_0$ is the initial learning rate and $k(K)$ is the total training step. We set $K = 2^{20}$ for all datasets. During the testing phase, we employ an exponential moving average with a momentum of 0.999 on the training model to perform inference for all algorithms. The batch size for labeled data is 64, with the exception of ImageNet, which has a batch size of 128. The same weight decay value, pre-defined threshold $\tau$, unlabeled batch ratio $\mu$ and loss weights are used for Pseudo-Label [24], II model [33], Mean Teacher [36], VAT [29], MixMatch [6], ReMixMatch [5], UDA [41], FixMatch [35], FlexMatch [47], CoMatch [26], SimMatch [51], AdaMatch [7], and FreeMatch [38].

We use the same parameters as in [37, 43] for Dash method. For other methods, we follow the original settings reported in their studies. In Appendix ??, you can find a comprehensive description of the hyperparameters used. To ensure the robustness, we train each algorithm three times with different random seeds. Consistent with [47], we report the lowest error rates achieved among all checkpoints.

4.2. CIFAR-10/100, STL-10, SVHN

The best error rate of each method is evaluated by averaging the results obtained from three runs with different random seeds. The results are presented in Table 2, where we report the classification error rates on the CIFAR-10/100, STL-10, and SVHN datasets. EPASS is shown to improve the performance of SimMatch and CoMatch significantly on all datasets. For instance, even though EPASS does not achieve state-of-the-art results in CIFAR-10/100, it still boosts the performance of conventional SimMatch and CoMatch. It should be noted that CIFAR-10/100 are small datasets where prior works have already achieved high performance, leaving little room for improvement. Moreover, ReMixMatch performs well on CIFAR-100 (2500) and CIFAR-100 (10000) due to the mixup technique and the self-supervised learning part. Additionally, on the SVHN and STL-10 datasets, SimMatch and CoMatch with EPASS surpass all prior state-of-the-art results by a significant margin, achieving a new state-of-the-art performance. These results demonstrate the effectiveness of EPASS in mitigating bias, particularly on imbalanced datasets such as SVHN and STL-10, where overfitting is a common issue.

4.3. ImageNet

EPASS is evaluated on the ImageNet ILSVRC-2012 dataset to demonstrate its effectiveness on large-scale datasets. In order to assess the performance of EPASS, we sample 100k/1%/10% of labeled images in a class-balanced manner, where the number of samples per class is 10, 13, or 128, respectively. The remaining images in each class are left unlabeled. Our experiments are conducted using a fixed random seed, and the results are found to be robust across different runs.

As presented in Table 3, EPASS outperforms the state-of-the-art methods, achieving a top-1 error rate of 39.47%/31.39%/24.70% for SimMatch and a top-1 error rate of 40.24%/32.64%/25.90% for CoMatch, respectively. The results clearly demonstrate the effectiveness of EPASS in improving the performance of SSL methods on large-scale datasets like ImageNet.

5. Ablation Study

5.1. ImageNet convergence speed

The convergence speed of the proposed EPASS is extremely noticeable through our extensive experiments. When training on ImageNet, we observe that EPASS achieves over 50% of accuracy in the first few iterations, indicating that the model is able to quickly learn meaningful representations from the unlabeled data. This is likely due to the fact that EPASS encourages the model to focus on the most informative and diverse instances during training, which helps the model learn more quickly and effectively. Additionally, we find that the accuracy of SimMatch and CoMatch with EPASS is consistently increasing with iterations, outperforming conventional SimMatch and CoMatch with the same training epochs. This suggests that the use
Table 2. Error rate on CIFAR-10/100, SVHN, and STL-10 datasets on 3 different folds. Bold indicates best result and Underline indicates the second best result.

<table>
<thead>
<tr>
<th>Method</th>
<th>Top-1</th>
<th>Top-5</th>
<th>Top-1</th>
<th>Top-5</th>
<th>Top-1</th>
<th>Top-5</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>100k</td>
<td>1%</td>
<td>10%</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FlexMatch [47]</td>
<td>41.85</td>
<td>19.48</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>CoMatch [26]</td>
<td>42.17</td>
<td>19.64</td>
<td>34.00</td>
<td>13.60</td>
<td>26.30</td>
<td>8.60</td>
</tr>
<tr>
<td>SimMatch [51]</td>
<td>41.15</td>
<td>19.23</td>
<td>32.80</td>
<td>12.90</td>
<td>25.60</td>
<td>8.40</td>
</tr>
<tr>
<td>FreeMatch [38]</td>
<td>40.57</td>
<td>18.77</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>SoftMatch [5]</td>
<td>40.52</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>[26] + EPASS</td>
<td>40.24</td>
<td>18.40</td>
<td>32.64</td>
<td>12.71</td>
<td>25.90</td>
<td>8.48</td>
</tr>
</tbody>
</table>

Table 3. ImageNet error rate results. Bold indicates best result and Underline indicates the second best result.

5.2. Calibration of SSL

Chen et al. [9] propose a method for addressing confirmation bias from the calibration perspective. To evaluate the effectiveness of EPASS in this regard, we measure the calibration of CoMatch and SimMatch on the ImageNet dataset with and without EPASS, using 10% labeled data. Several common calibration indicators, including Expected Calibration Error (ECE), confidence histogram, and reliability diagram, are utilized in this study.

Figure 4 illustrates that when EPASS is used with 10% of labels, the ECE value of the model decreases. Moreover, under the 1% label scheme, CoMatch and SimMatch without EPASS are significantly overconfident and overfitted due to confirmation bias. However, when EPASS is employed, it helps to reduce the ECE by a large margin and also mitigate the overconfidence of the model. Notably, models with EPASS have average accuracy and average confidence that are approximately equal, whereas the average confidence of models without EPASS is usually higher than the accuracy.

It is worth mentioning that since CoMatch does not impose the interaction between semantic and instance similarity like SimMatch, the effect of introducing EPASS to CoMatch for calibration is not as significant as that for SimMatch. Additionally, the model with EPASS becomes underfit and may benefit from additional training.

5.3. Number of projectors

This section studies the effectiveness of the proposed projectors ensemble method and how different ensemble strategies affect performance. In this experiment, we study the effect of different numbers of projectors on performance. The top-1 classification accuracy of the proposed EPASS with different numbers of projectors is shown in Ta-

https://github.com/hollance/reliability-diagrams
Table 4. Top-1 accuracy (%) on ImageNet 10% using different # projectors.

<table>
<thead>
<tr>
<th>Method</th>
<th># projectors</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>CoMatch [26] + EPASS</td>
<td>73.6</td>
<td>73.8</td>
<td>74.1</td>
<td>73.9</td>
<td></td>
</tr>
<tr>
<td>SimMatch [51] + EPASS</td>
<td>74.4</td>
<td>74.8</td>
<td>75.3</td>
<td>75.2</td>
<td></td>
</tr>
</tbody>
</table>

Table 5. Top-1 accuracy (%) on ImageNet 10% using different ensemble strategies.

<table>
<thead>
<tr>
<th>Method</th>
<th>Ensemble strategy</th>
<th>Concatenate</th>
<th>Sum</th>
<th>Mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>CoMatch [26] + EPASS</td>
<td></td>
<td>74.0</td>
<td>73.9</td>
<td>74.1</td>
</tr>
<tr>
<td>SimMatch [51] + EPASS</td>
<td></td>
<td>75.1</td>
<td>74.8</td>
<td>75.3</td>
</tr>
</tbody>
</table>

5.4. Imbalanced SSL

<table>
<thead>
<tr>
<th>Dataset</th>
<th>CIFAR-10-LT</th>
<th>CIFAR-100-LT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Imbalance</td>
<td>λ = 50</td>
<td>λ = 150</td>
</tr>
<tr>
<td>FixMatch [35]</td>
<td>18.5±0.48</td>
<td>31.2±1.08</td>
</tr>
<tr>
<td>FlexMatch [47]</td>
<td>17.8±0.24</td>
<td>29.5±0.47</td>
</tr>
<tr>
<td>FreeMatch [38]</td>
<td>17.7±0.33</td>
<td>28.9±0.04</td>
</tr>
<tr>
<td>SoftMatch [8]</td>
<td>16.6±0.29</td>
<td>27.4±0.46</td>
</tr>
<tr>
<td>CoMatch [26]</td>
<td>16.3±0.24</td>
<td>30.1±0.31</td>
</tr>
<tr>
<td>SimMatch [51]</td>
<td>20.3±0.31</td>
<td>28.7±0.48</td>
</tr>
<tr>
<td>[26] + EPASS</td>
<td>16.1±0.22</td>
<td>29.6±0.41</td>
</tr>
<tr>
<td>[51] + EPASS</td>
<td>18.2±0.54</td>
<td>28.4±0.40</td>
</tr>
</tbody>
</table>

Table 6. Error rates (%) of imbalanced SSL using 3 different random seeds. **Bold** indicates best result and **Underline** indicates the second best result.

To provide additional evidence of the effectiveness of EPASS, we assess its performance in the imbalanced semi-supervised learning scenario [16, 25, 39], where both the labeled and unlabeled data are imbalanced. Our experi-
Table 7. Error rate on CIFAR-10/100, SVHN, and STL-10 datasets on 3 different folds. **Bold** indicates best result and **Underline** indicates second best result.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>CIFAR-100</th>
<th>STL-10</th>
<th>Euro-SAT</th>
<th>TissueMNIST</th>
<th>Semi-Aves</th>
</tr>
</thead>
<tbody>
<tr>
<td>Label Amount</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>200</td>
<td>30.75±0.10</td>
<td>19.94±0.32</td>
<td>39.22±2.87</td>
<td>23.59±2.97</td>
<td>11.15±1.20</td>
</tr>
<tr>
<td>400</td>
<td>37.43±0.58</td>
<td>26.17±0.24</td>
<td>48.98±1.41</td>
<td>25.56±1.30</td>
<td>29.86±2.89</td>
</tr>
<tr>
<td><strong>30.85±1.42</strong></td>
<td><strong>18.80±0.99</strong></td>
<td><strong>30.61±3.47</strong></td>
<td><strong>18.33±1.98</strong></td>
<td><strong>4.53±1.60</strong></td>
<td><strong>4.10±0.37</strong></td>
</tr>
<tr>
<td>FixMatch [35]</td>
<td>30.45±0.65</td>
<td>19.48±0.93</td>
<td>42.06±3.94</td>
<td>24.05±1.79</td>
<td>12.48±2.57</td>
</tr>
<tr>
<td>FlexMatch [47]</td>
<td>27.08±0.90</td>
<td>17.67±0.66</td>
<td>37.58±2.97</td>
<td>23.40±1.50</td>
<td>7.07±2.32</td>
</tr>
<tr>
<td>Dash [43]</td>
<td>30.19±1.34</td>
<td>18.90±0.420</td>
<td>43.34±1.46</td>
<td>25.90±0.35</td>
<td>9.44±0.75</td>
</tr>
<tr>
<td>CoMatch [26]</td>
<td>35.68±0.54</td>
<td>26.10±0.09</td>
<td>29.70±1.17</td>
<td>21.46±1.34</td>
<td>5.25±0.49</td>
</tr>
<tr>
<td>SimMatch [51]</td>
<td>23.26±1.25</td>
<td>16.82±0.40</td>
<td>34.12±1.63</td>
<td>22.97±2.04</td>
<td>6.88±1.77</td>
</tr>
<tr>
<td>AdaMatch [7]</td>
<td>21.27±1.04</td>
<td>17.01±0.55</td>
<td>36.25±1.89</td>
<td>23.30±0.73</td>
<td>5.70±0.37</td>
</tr>
<tr>
<td><strong>26</strong> + EPASS</td>
<td><strong>35.10±0.55</strong></td>
<td><strong>25.53±0.50</strong></td>
<td><strong>29.56±2.50</strong></td>
<td><strong>21.14±0.31</strong></td>
<td><strong>3.41±0.24</strong></td>
</tr>
<tr>
<td><strong>51</strong> + EPASS</td>
<td><strong>22.52±0.83</strong></td>
<td><strong>16.78±0.59</strong></td>
<td><strong>30.03±0.71</strong></td>
<td><strong>22.65±1.94</strong></td>
<td><strong>5.35±0.81</strong></td>
</tr>
<tr>
<td>Fully-Supervised</td>
<td>8.90±0.12</td>
<td>-</td>
<td>0.85±0.06</td>
<td>33.91±0.03</td>
<td>-</td>
</tr>
</tbody>
</table>

In this section, we evaluate the effectiveness of EPASS within the context of the USB [37] framework, adhering strictly to the USB settings for CV tasks that utilize pre-trained Vision Transformers (ViT). For a detailed overview of hyperparameters used in these experiments, please refer to Appendix ??.
References


[27] Dhruv Mahajan, Ross Girshick, Vignesh Ramanathan, Kaiming He, Manohar Paluri, Yixuan Li, Ashwin Bharambe, and Laurens van der Maaten. Exploring the limits of weakly supervised pretraining, 2018. 1


[38] Yidong Wang, Hao Chen, Qiang Heng, Wenxin Hou, Yue Fan, Zhen Wu, Jindong Wang, Marios Savvides, Takahiro Shoinzaki, Bhiksha Raj, Bernt Schiele, and Xing Xie. Freematch: Self-adaptive thresholding for semi-supervised learning. 2023. 2, 5, 6, 7


