


## Article

# GBVSSL: Contrastive Semi-Supervised Learning Based on Generalized Bias-Variance Decomposition

Shu Li <sup>1,\*</sup>, Lixin Han <sup>1</sup>, Yang Wang <sup>2</sup>  and Jun Zhu <sup>3</sup><sup>1</sup> School of Computer and Information, Hohai University, Nanjing 211100, China; lixinhan2002@hotmail.com<sup>2</sup> School of Computer and Information, Anqing Normal University, Anqing 246133, China; wangy@aqnu.edu.cn<sup>3</sup> School of Computer and Software, Nanjing Vocational University of Industry Technology, Nanjing 210023, China; zj\_zijin@163.com

\* Correspondence: shul93@hhu.edu.cn

**Abstract:** Mainstream semi-supervised learning (SSL) techniques, such as pseudo-labeling and contrastive learning, exhibit strong generalization abilities but lack theoretical understanding. Furthermore, pseudo-labeling lacks the label enhancement from high-quality neighbors, while contrastive learning ignores the supervisory guidance provided by genuine labels. To this end, we first introduce a generalized bias-variance decomposition framework to investigate them. Then, this research inspires us to propose two new techniques to refine them: neighbor-enhanced pseudo-labeling, which enhances confidence-based pseudo-labels by incorporating aggregated predictions from high-quality neighbors; label-enhanced contrastive learning, which enhances feature representation by combining enhanced pseudo-labels and ground-truth labels to construct a reliable and complete symmetric adjacency graph. Finally, we combine these two new techniques to develop an excellent SSL method called GBVSSL. GBVSSL significantly surpasses previous state-of-the-art SSL approaches in standard benchmarks, such as CIFAR-10/100, SVHN, and STL-10. On CIFAR-100 with 400, 2500, and 10,000 labeled samples, GBVSSL outperforms FlexMatch by 3.46%, 2.72%, and 2.89%, respectively. On the real-world dataset Semi-iNat 2021, GBVSSL improves the Top-1 accuracy over CCSSL by 4.38%. Moreover, GBVSSL exhibits faster convergence and enhances unbalanced SSL. Extensive ablation and qualitative studies demonstrate the effectiveness and impact of each component of GBVSSL.

**Keywords:** bias-variance decomposition; semi-supervised learning; pseudo-labeling; contrastive learning



**Citation:** Li, S.; Han, L.; Wang, Y.; Zhu, J. GBVSSL: Contrastive Semi-Supervised Learning Based on Generalized Bias-Variance Decomposition. *Symmetry* **2024**, *16*, 724. <https://doi.org/10.3390/sym16060724>

Academic Editor: Christos Volos

Received: 11 May 2024

Revised: 6 June 2024

Accepted: 7 June 2024

Published: 11 June 2024



**Copyright:** © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

In recent years, semi-supervised learning [1–8] (SSL) based on various pseudo-labeling [2,3,9] and contrastive learning approaches [10–12] has demonstrated tremendous potential, thereby attracting widespread attention in the academic community. Since these semi-supervised learning (SSL) methods fit model predictions with true labels on scarce labeled data and efficiently utilize a large amount of unlabeled data for self-training, they significantly enhance the models' generalization performance and effectively reduce reliance on costly manual labeling.

Mainstream pseudo-labeling methods [2,3,13] use their models to assign reliable pseudo-labels to unlabeled samples for self-training. For example, UDA [13] and FixMatch [2] select pseudo-labels with confidence scores exceeding a fixed threshold (e.g., 0.95) for training, but they discard a considerable number of uncertain yet correct pseudo-labels. To compensate for this deficiency, methods such as Dash [14] and Adamatch [15] propose dynamic growing thresholds, while approaches like Flexmatch [3] and Adsh [16] adopt category-aware adaptive thresholds to acquire more pseudo-labels and consequently enhance the performance of FixMatch. However, these types of adaptive threshold methods may accept more incorrect pseudo-labels, thereby misleading the model. Therefore, confidence threshold-based pseudo-labeling mechanisms suffer from the dilemma of balancing

utilization and accuracy. Furthermore, existing pseudo-labeling relies on classifiers trained on labeled datasets to make individual predictions for unlabeled instances, overlooking the label enhancement from high-quality neighbors and the propagation of label supervision. This oversight can lead to imprecise pseudo-labels, consequently causing confirmation bias.

On the other hand, related research integrating contrastive learning [11,12] with SSL technology [1,6,8] has made remarkable contributions to the field of semi-supervised learning, significantly enhancing the performance of existing models. Early self-supervised contrastive learning [10,12] aims to separate perturbed versions of different instances in a pre-training task. This category-agnostic feature learning approach inevitably leads to the dispersion of sample features within the same class, severely violating the clustering property of SSL [1,6]. To address this issue, relevant researchers propose class-aware contrastive learning [1]. This approach mainly incorporates a neighborhood graph constructed by pseudo-labels from unlabeled samples into the contrastive loss function, aiming to effectively maintain the similarity of features within the same class and the distinctiveness of features between different classes, thus better adapting to semi-supervised learning tasks. However, existing class-aware contrastive learning still has some limitations. For instance, inaccurate pseudo-labels can lead to the misallocation of positive and negative pairs by the class-aware mechanism, thereby forming an adjacency graph with numerous erroneous connections. Such an adjacency graph will guide the contrastive constraints to push features from the same class apart and pull features from different classes closer, inevitably compromising the effectiveness of representation learning. Furthermore, it fails to utilize reliable labeled supervision to effectively guide the learning process of unlabeled data, leading to suboptimal feature representations.

In addition to the aforementioned deficiencies, pseudo-labeling and contrastive learning commonly lack theoretical foundations, and their theoretical connections remain unclear. To address these issues, we propose a generalized bias-variance decomposition framework to study and understand these two techniques, inspired by the observation that the bias-variance decomposition of cross-entropy is closely related to the SSL approach with respect to structure and generalization ability. This study initially inspires us to propose neighbor-enhanced pseudo-labeling and label-enhanced contrastive learning to address the limitations of existing techniques. Subsequently, it motivates us to unify them into a single loss function, resulting in the introduction of a novel SSL method named GBVSSL. Specifically, GBVSSL constructs two representations for each instance: one is the class probability output by the classifier, and the other is the low-dimensional embedding output by the projection head. The two representations interact and co-evolve through neighbor-enhanced pseudo-labeling and label-enhanced contrastive learning. Bias analysis encourages neighbor-enhanced pseudo-labeling to improve the individual predictions of target samples by leveraging aggregated predictions from high-quality neighbors, thereby enhancing the accuracy of corresponding pseudo-labels. Variance analysis guides label-enhanced contrastive learning to once again utilize aggregated predictions to enhance the pseudo-labels of unlabeled samples, while combining label supervision to construct a reliable adjacency graph, thereby enhancing feature representation. In essence, GBVSSL effectively reduces the generalization error of the model by simultaneously minimizing both bias and variance. The main advantage lies in effectively propagating label supervision to unlabeled data, enhancing the quality of pseudo-labels and feature representation. GBVSSL demonstrates outstanding generalization advantage on standard benchmarks, such as CIFAR-10/100, SVHN, and STL-10. Especially for CIFAR-100 with 400, 2500, and 10,000 labeled samples, GBVSSL achieves error rates of 37.27%, 23.45%, and 18.86%, outperforming the well-known method FlexMatch by 3.46%, 2.72%, and 2.89%, respectively. Furthermore, on the real-world dataset Semi-iNat 2021, GBVSSL improves the Top-1 accuracy over CCSSL [1] by 4.38%.

Our main contributions can be summarized in four aspects:

1. We introduce a generalized bias-variance decomposition framework for studying and understanding pseudo-labeling and contrastive learning;

2. We propose neighbor-enhanced pseudo-labeling and label-enhanced contrastive learning to improve pseudo-label and feature representation, respectively.
3. We present a novel SSL algorithm, GBVSSL, which combines neighbor-enhanced pseudo-labeling and label-enhanced contrastive learning.
4. Extensive experiments demonstrate that GBVSSL outperforms previous state-of-the-art methods on multiple SSL benchmarks and establishes a new performance benchmark on the real-world dataset Semi-INet 2021.

## 2. Related Work

### 2.1. Confidence-Based Pseudo-Labeling

In mainstream semi-supervised learning methods, confidence-based pseudo-labeling [2,6,17] serves as a pivotal component capable of substantially enhancing their performance. Currently, these methods [3–5,13,14,18] focus on improving the quality of pseudo-labels. For example, FixMatch [2] and UDA [13] directly employ a fixed high-confidence threshold to ensure pseudo-label accuracy. However, recent studies have indicated that this approach exhibits poor performance in low-data settings, specifically showing overfitting issues, particularly on easily learnable minority classes. To tackle such issues, Flexmatch [3] and Dash [14] propose category- or instance-based adaptive threshold methods. FreeMatch [18] adopts a confidence threshold method that is adaptive to the training state of the model. In addition, ReMixMatch [4] utilizes a sliding average of pseudo-labels, while SoftMatch [5] weights the samples using a truncated Gaussian function, both of which are effective in adjusting for bias. Another promising direction for research is to optimize the feature representation through additional training on auxiliary tasks, aiming to refine the predictions of the classifier. For instance, SimMatch [8] and CoMatch [6] enhance pseudo-labels by performing instance similarity matching and graph-based contrastive learning, respectively. Different from existing methods, we propose a neighbor-enhanced pseudo-labeling approach, which enhances confidence-based pseudo-labels by leveraging aggregated predictions generated through label propagation on a carefully constructed neighbor graph.

### 2.2. Contrastive Learning-Based SSL

Recently, some studies have achieved state-of-the-art performance by effectively integrating contrastive learning [10–12] and SSL techniques [1,2,6–8]. Among these methods, naive self-supervised contrastive learning [10], which employs category-agnostic feature learning, may result in the dispersion of sample features within the same class, thereby contradicting the clustering property of semi-supervised learning [1,6]. To address this contradiction, researchers have naturally proposed leveraging the category-relevant prior information provided by pseudo-labels to enhance contrastive learning. For example, FixMatch [2] employs pseudo-labels exceeding a high confidence threshold for supervised contrastive learning, while CoMatch [6] utilizes memory-smoothed pseudo-labels for graph-based contrastive learning. Similarly, ConMatch [7] introduces pseudo-labels as the supervisory information for its contrastive loss. SimMatch [8] enhances FixMatch [2] by combining the semantic consistency loss guided by pseudo-labeling with the sample similarity loss. The most relevant to our work, CCSL [1], improves contrastive learning through weighted clustering of samples guided by pseudo-labels. Despite the impressive results of these studies, their performance is constrained by a few issues, mainly including instance pair mismatches due to incorrect pseudo-labeling, and underutilization of label supervision information. To address these issues, this study proposes label-enhanced contrastive learning, which guides and optimizes feature representation by combining enhanced pseudo-labels from unlabeled data and label supervision to construct reliable adjacency graphs.

### 3. Generalized Bias-Variance Decomposition

Bias-variance decomposition is a key method for studying the generalization performance of machine learning models, and therefore serves as the theoretical cornerstone of this research. Following the assumptions and definitions of [19], let us consider a classification model  $f$  on a given dataset  $D$  with  $c$  classes. Let  $y \in R^c$  be the one-hot encoding of the ground-truth label, and  $\hat{y} = \hat{f}(x, D) \in R^c$  be the model's output prediction. The expression  $\bar{y} = \bar{f}(x, D)$  is defined as the average of the normalized logarithmic probabilities  $\hat{y}$ .

$$\bar{y} = \frac{1}{Z} \exp[E_D \log \hat{y}] \quad (1)$$

where  $Z$  is the normalization constant. In other words,  $\bar{y}$  represents the expectation of the model. We directly delve into the generalized bias-variance decomposition framework of cross-entropy proposed by [19,20] for the classification model  $f$ .

$$\begin{aligned} \mathcal{L}_{\text{general}} &= E_{x,D}[-y \log \hat{y}] \\ &= \delta^2 + D_{KL}(y \parallel \bar{y}) + E_D[D_{KL}(\bar{y} \parallel \hat{y})] \\ &= H(y) + H(y, E[\bar{f}(x, D)]) - H(y) \\ &\quad + D_{KL}(E[\bar{f}(x, D)] \parallel \hat{f}(x, D)) \\ &= \underbrace{H(y, \bar{f}(x, D))}_{\text{bias}} + \underbrace{D_{KL}(\bar{f}(x, D) \parallel \hat{f}(x, D))}_{\text{variance}} \\ &\leq H(y, \hat{f}(x, D)) + D_{KL}(\bar{f}(x, D) \parallel \hat{f}(x, D)) \end{aligned} \quad (2)$$

where  $\delta^2$ ,  $D_{KL}(y \parallel \bar{y})$ ,  $E_D[D_{KL}(\bar{y} \parallel \hat{y})]$  are the noise, bias, and variance terms defined by [19,20], respectively.  $D_{KL}$  represents the Kullback–Leibler divergence. We keep the variance term unchanged while gradually simplifying the bias to  $H(y, \bar{f}(x, D))$ . To ensure the generalization performance of model  $f$ , it is crucial to simultaneously minimize both the bias and variance in Equation (2). A significant challenge lies in providing a solution for the unknown expectation function  $\bar{f}(x, D)$ .

#### 3.1. Analysis and Implementation Scheme for Minimizing the Bias $H(y, \bar{f}(x, D))$

According to Equation (2), we understand that existing SSL methods employ the standard cross-entropy loss  $H(y, \bar{f}(x, D))$  as the supervised loss for labeled data and the pseudo-labeling loss for unlabeled data, primarily to minimize model bias. Equation (2) indicates that  $H(y, \bar{f}(x, D))$  is a lower bound on  $H(y, \hat{f}(x, D))$ ; thus, minimizing  $H(y, \bar{f}(x, D))$  theoretically enables more effective reduction of generalization error. In other words, the output expectation  $\bar{f}(x, D)$  is closer to the ground-truth label  $y$  than the prediction  $\hat{f}(x, D)$  in practice, thereby enhancing prediction accuracy. This bias analysis inspires us to incorporate the output expectation of samples to enhance the accuracy of pseudo-labeling in SSL. To introduce the implementation scheme of bias, we define the bias of an arbitrary sample point  $(x_i, y_i)$  as follows:

$$H(y_i, \bar{f}(x_i, D)) = H\left(y^m, \frac{1}{n} \sum_{i=1}^n \sum_{k=1}^c p_i^k \cdot y^k\right) \quad (3)$$

where  $n$  and  $c$  denote the number of samples and classes in dataset  $D$ , respectively. Assume the label  $y_i$  of sample  $x_i$  belongs to the  $m$ th class, and  $y^m$  represents the one-hot encoding of the  $m$ th class. Here, we first assume that the expectation  $\bar{f}(x_i, D) = \frac{1}{n} \sum_{j=1}^n \hat{y}_j$ ,  $\hat{y}_j$  is the model's prediction for  $x_j$ , and  $p_j^k$  is the assignment probability of  $x_j$  on the  $k$ th class one-hot vector  $y^k$ ,  $\hat{y}_j = \sum_{k=1}^c p_j^k \cdot y^k$ . The minimization of Equation (3) is equivalent to maximize the consistency between  $\frac{1}{n} \sum_{j=1}^n p_j^m \cdot y^m$  and  $y^m$ . Ideally, if sample  $x_j$  belongs to the  $m$ th class, then  $p_j^m$  should approach 1, and it should contribute to the calculation of the

expectation of  $x_j$ . If  $x_j$  does not belong to the  $m$ th class,  $p_j^m$  should approach 0, and such samples should be excluded from Equation (3). Therefore, we further define the output expectation  $\bar{f}(x_i, D)$  of sample  $x_i$  as the average of the output predictions  $\hat{y}_j$  of all samples  $x_j$  belonging to the same class as  $x_i$ . The message-passing mechanism of GNNs [21], akin to this definition, inspires us to calculate the expectation  $\bar{f}(x_i, D)$  using its efficient matrix computation form. The implementation is as follows: We begin by obtaining the feature matrix  $F$  and the prediction matrix  $\hat{Y}$  output by the model  $f$  on the dataset  $D$ . Then, we construct the normalized adjacency matrix  $A$  based on the features  $F$  and predictions  $\hat{Y}$  on  $D$ . Finally, we naturally arrive at the matrix computation form of the expectation:

$$\bar{f}(x_i, D) = A_i \hat{Y} \quad (4)$$

where  $A_i$  comprises all the neighbors and their weights that belong to the same class as the target sample  $x_i$ . Ensuring accurate estimates of both  $A$  and  $\hat{Y}$  is crucial for the quality of the expectation in Equation (4).

For existing pseudo-labeling methods [2,6,17] in semi-supervised learning, pseudo-labels are selected from high-confidence predictions of individual target samples, but their accuracy needs improvement. Additionally, this “hard” labeling approach discards a considerable number of uncertain yet correct pseudo-labels, resulting in low utilization of unlabeled data. The bias analysis motivates us to propose neighbor-enhanced pseudo-labeling, which fully leverages the expected predictions of reliable neighbors (potential samples of the same class) to generate high-quality pseudo-labels for unlabeled data. For the neighbors in  $A$ , we should prioritize labeled samples with high feature similarity, and then select unlabeled samples with high-confidence predictions and high feature similarity, and  $\hat{Y}$  should consist of high-confidence predictions on unlabeled samples and ground-truth labels on labeled samples. Furthermore, transductive inference [22] motivates us to further generalize the Equation (4) to multiple rounds of label propagation for  $\hat{Y}$  on  $A$ , aiming to achieve accurate predictions that tend to converge. On the other hand, we reasonably assume that as the model training progresses, the sharpened low-confidence predictions gradually converge towards the true labels. Therefore, we utilize them as “soft” pseudo-labels for pseudo-labeling loss, aiming to enhance the utilization of unlabeled data and reduce prediction bias on low-confidence unlabeled data.

### 3.2. Analysis and Implementation Scheme for Minimizing the Variance $D_{KL}(\bar{f}(x, D) \parallel \hat{f}(x, D))$

This study directly converts the complex  $D_{KL}(\bar{f}(x, D) \parallel \hat{f}(x, D))$  into the problem  $\left| \bar{f}(x, D) - \hat{f}(x, D) \right|^2$  of minimizing redundancy loss, as they both aim to minimize the difference between  $\bar{f}(x, D)$  and  $\hat{f}(x, D)$ . Notably, we redefine the expectation as the sum of the probability distributions predicted by the model across all classes.

$$\bar{f}(x_i, D) = E_y \left[ \hat{f}(x_i, D) \right] = \sum_{k=1}^c p_i^k \cdot y^k \quad (5)$$

We reasonably assume that the output prediction corresponds to the probability output of model  $f$  on the true class of  $x_i$ , as it typically represents the maximum probability output.

$$\hat{f}(x_i, D) = p_i^t \cdot y^t \quad (6)$$

where  $y^t$  represents the one-hot encoding of the true label for  $x_i$ , while  $p_i^t$  is the output probability of the model for sample  $x_i$  on the label  $y^t$ . This definition and assumption



constitute the pivotal conditions for addressing the redundancy minimization problem at hand. Thus, the redundancy loss over the entire dataset  $D$  is as follows:

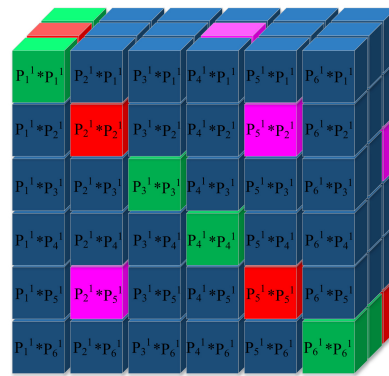
$$\begin{aligned}\mathcal{L}_{\text{redundancy}}^{V_1} &= E_D E_x \left[ \left| \hat{f}(x, D) - E_y[\hat{f}(x, D)] \right|^2 \right] \\ &= \sum_i^n \left[ \left( \sum_{k=1}^c 1[y^k \neq y_i^t] p_i^k \cdot y^k \right)^T \left( \sum_{k=1}^c 1[y^k \neq y_i^t] p_i^k \cdot y^k \right) \right]\end{aligned}\quad (7)$$

The redundancy minimization represented by Equation (7) aims to minimize the correlation among features from different classes. This aligns with the concepts explored in the related study of Barlow Twins [23] and warrants further discussion in future research. We transpose the dimensions  $n$  and  $c$  in Equation (7), resulting in an alternative redundancy minimization paradigm represented by Equation (8). This section will delve into the analysis of Equation (8), as it introduces a variance minimization scheme closely related to the contrastive learning technique emphasized in this study.

$$\mathcal{L}_{\text{redundancy}}^{V_2} = \sum_{k=1}^c \left[ \left( \sum_{i=1}^n 1[y^k \neq y_i^t] p_i^k \cdot y^k \right)^T \left( \sum_{i=1}^n 1[y^k \neq y_i^t] p_i^k \cdot y^k \right) \right]\quad (8)$$

We take the sample set  $D = [(x_2^1, y_2^1), (x_5^1, y_5^1)], [(x_1^2, y_1^2), (x_4^2, y_4^2)], [(x_3^3, y_3^3), (x_6^3, y_6^3)]$  as an example and plot  $V_2$  in Figure 1 to visualize Equation (8). Specifically,  $D$  consists of  $n = 6$  samples, divided into  $c = 3$  categories, where  $(x_i^j, y_i^j) \in D$  represents the  $i$ th sample and its  $j$ th class label.  $V_2$  has a shape of  $R^{c \times n \times n}$  and symmetrically contains elements of four colors: red, green, pink, and dark blue. Equation (8) excludes the calculations of red  $p_i^t * p_i^t$  and pink  $p_i^t * p_j^t$  elements using the indicator  $1[y^k \neq y_i^t]$ , aiming to minimize the green  $p_i^k * p_i^k$  and dark blue  $p_i^k * p_j^k$  elements. Since minimizing the cross-entropy  $H(y, \hat{f}(x, D))$  will maximize  $p_i^t$  and  $p_j^t$ , the red  $p_i^t * p_i^t$  and pink  $p_i^t * p_j^t$  elements should be maximized. Numerical analysis of the four aforementioned elements suggests that to ensure the output predictions of  $\hat{f}(x, D)$  for any sample  $x_i$  are consistent with its label  $y_i^t$ , the red  $p_i^t * p_i^t$  should be maximized; to ensure the output predictions of  $\hat{f}(x, D)$  are consistent within the same class of samples, where  $y_i^t$  and  $y_j^t$  represent the samples  $x_i$  and  $x_j$  belonging to the same category, the pink  $p_i^t * p_j^t$  should be maximized. On the other hand, to minimize the output probability  $p_i^k$  of the non-true label  $y_i^k$  for sample  $x_i$  by  $\hat{f}(x, D)$ , the green  $p_i^k * p_i^k$  should be minimized. Additionally, to ensure that  $\hat{f}(x, D)$  effectively distinguishes samples from different categories and provides distinct predictions, where  $y_i^k$  and  $y_j^k$  denote the labels of samples  $x_i$  and  $x_j$  not belonging to the same category, the dark blue  $p_i^k * p_j^k$  should be minimized. For a more detailed analysis, please refer to our published GCL [24] model. Note that  $*$  represents the multiplication sign.

When extending the output prediction  $\hat{f}(x_i, D)$  to the feature representation  $z_i$  in the embedding space, the redundancy minimization paradigm described by Equation (8) aims to maximize the similarity among intra-class sample features and minimize the similarity among inter-class sample features. Essentially, it ensures consistency between the similarity matrix and the adjacency matrix of dataset  $D$ . We can achieve this objective by directly minimizing the cross-entropy  $H(A, S)$ , where  $A$  represents the accurate adjacency matrix and  $S$  represents the normalized similarity matrix, e.g.,  $S_{ij} = \frac{\exp(z_i \cdot z_j)}{\sum_{j \in A(i)} \exp(z_i \cdot z_j)}$ . This approach is actually a contrastive learning method based on an adjacency graph. We naturally conclude that contrastive learning achieves excellent generalization performance by effectively reducing the model's variance.



**Figure 1.** The symmetric  $V_2$  is used to visualize Equation (8). Minimizing the elements represented by the green  $p_i^k * p_i^k$  and the dark blue  $p_i^k * p_j^k$  in tensor  $V_2$  is equivalent to achieving the minimum variance as described in Equation (8) across the dataset  $D$ .

In semi-supervised learning,  $S$  can be easily computed from the features matrix  $F$ , with the primary challenge being the unknown  $A$ . Currently, contrastive learning in SSL primarily leverages high-confidence pseudo-labels to construct the adjacency matrix  $A$ , which guides the representation learning of unlabeled data and thereby achieves some performance improvement. These works strongly support variance analysis, but they struggle to ensure the reliability of pseudo-labels and are susceptible to confirmation bias. Furthermore, our variance analysis is conducted on the entire dataset, so the contrastive learning represented by  $H(A, S)$  should encompass both labeled and unlabeled data, whereas existing contrastive learning methods are primarily utilized for unlabeled data, without considering reliable labeled data, resulting in the incompleteness of  $A$ .

This variance analysis prompts us to propose label-enhanced contrastive learning, which reuses aggregated predictions to obtain more accurate pseudo-labels, and then combines the enhanced pseudo-labels with the ground-truth labels to construct a reliable and complete adjacency matrix  $A$ . Label-enhanced contrastive learning enhances the feature representation of both labeled and unlabeled data by incorporating  $A$  as a contrastive constraint for feature learning, thereby addressing the limitations of existing SSL-based contrastive learning methods.

In conclusion, we understand that pseudo-labeling and contrastive learning correspond to the minimization of bias and variance, respectively. They can be unified into a generalized bias-variance loss function to jointly reduce the model's generalization error.

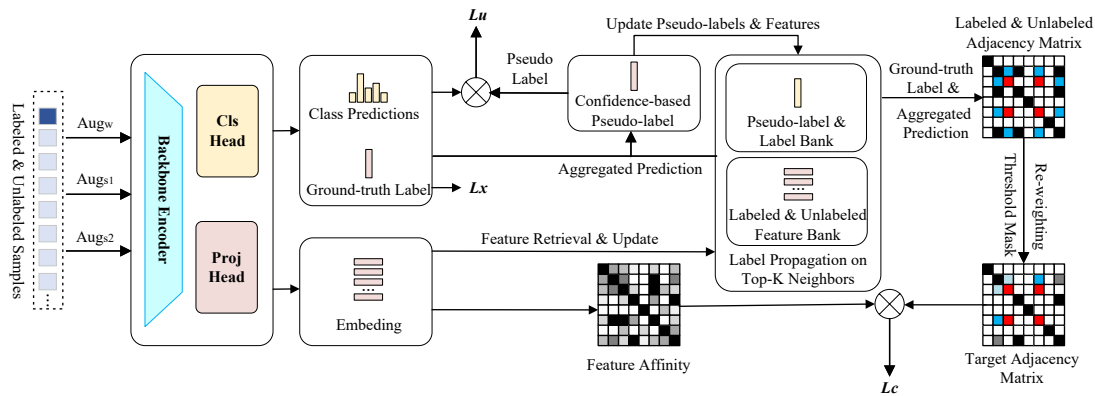
## 4. The Proposed GBVSSL Method

### 4.1. Problem Formulation

A semi-supervised classification task is typically set up with  $X$  and  $U$ , where  $X = \{(x_i, y_i) : i \in (1, \dots, B)\}$  denotes a batch of labeled image data,  $x_i$  denotes the  $i$ th labeled sample, and  $y_i$  represents the one-hot label for  $x_i$ . While  $U = \{u_i : i \in (1, \dots, \mu B)\}$  represents the same batch of unlabeled data, the coefficient  $\mu$  determines the ratio of unlabeled data to labeled data. The semi-supervised classification model learns simultaneously on both labeled data  $X$  and unlabeled data  $U$ , but only has access to a small number of labels from  $X$  for classification training. The objective is to achieve accurate predictions for the labels of  $U$ .

### 4.2. Framework

Inspired by the research findings of the general bias-variance decomposition, this study proposes a new SSL method called GBVSSL. GBVSSL mainly consists of three modules: conventional cross-entropy on labeled data, neighbor-enhanced pseudo-labeling, and label-enhanced contrastive learning, with the latter two primarily applied to unlabeled data. We describe the various modules of GBVSSL depicted in Figure 2 in a left-to-right sequential order, elucidating the process from input to output in detail.



**Figure 2.** The framework and pipeline of GBVSSL. The weakly augmented views with labels are directly used for supervised learning; The weakly augmented views of unlabeled data are used together with their reliable Top-K neighbors found in the memory bank for label propagation, thereby obtaining aggregated predictions and their confidence-based pseudo-labels, which are then used for pseudo-labeling along with the corresponding predictions from strongly augmented views; Combining the aggregated predictions of weakly augmented views from unlabeled views with the ground-truth labels from labeled views, a target adjacency matrix is constructed to guide the contrastive learning of features from strongly augmented views of both labeled and unlabeled data in the same batch.

For all input images, we use a weak augmentation  $Aug_w(\cdot)$  and two types of strong augmentation  $Aug_s(\cdot)$  to generate multiple views for them. For labeled sample  $x_i$ , we use its weak augmentation for supervised learning. For unlabeled sample  $u_i$ , a weak augmentation used for estimating pseudo-labels and a strong augmentation used for training predictions are simultaneously fed into neighbor-enhanced pseudo-labeling. Meanwhile, two types of strong augmentations for  $x_i$  and  $u_i$  are employed for label-enhanced contrastive learning. Weak augmentation with slight variations leads to less loss of information, enabling more accurate predictions and making them suitable as pseudo-labels, thereby aiding the model in capturing the inherent structure of the data more accurately. Conversely, strong augmentation enhances the model's adaptability to significant variations, thereby improving the model's feature representation capability and generalization performance.

For the input augmented views, we utilize the image encoder  $E(\cdot)$  to extract their feature representations  $r = E(Aug(x))$ , with the choice of encoder being independent.

The label-enhanced contrastive learning module maps the high-dimensional feature  $r$  of the image to the low-dimensional feature space  $z$  using the projection head  $proj(\cdot)$  for contrastive learning. The implementation ensures the desired feature distribution by minimizing the cross-entropy between the feature affinity matrix and the target adjacency matrix. It should be emphasized that both matrices, similar to  $V_2$ , possess symmetry. By leveraging this consistent symmetry, these two matrices effectively enhance the model's adaptability to significant variations in different augmented views of the same sample, thereby improving the model's generalization performance.

The standard cross-entropy trains the classifier  $P_{cls}(\cdot)$  to output classification predictions during inference. The neighbor-enhanced pseudo-labeling minimizes the cross-entropy between the predictions of unlabeled strong augmentations and their pseudo-labels derived from the aggregated predictions of the corresponding weak augmentations.

Next, we will sequentially detail the three loss modules of GBVSSL under the generalized bias-variance decomposition framework.

#### 4.3. Bias Minimization

##### 4.3.1. Labeled Cross-Entropy

To reduce the prediction bias of the model on each labeled image  $x_i$ , we perform supervised learning by minimizing the standard cross-entropy [2] between the one-hot



ground-truth label  $y_i$  and the prediction  $\hat{y}_i^w$  of weak augmentation, as illustrated in Equation (9):

$$L_x = \frac{1}{B} \sum_{i=1}^B y_i \log \hat{y}_i^w \quad (9)$$

where the coefficient  $B$  denotes the batch size.

#### 4.3.2. Neighbor-Enhanced Pseudo-Labeling

To reduce the prediction bias of the model on unlabeled data, the neighbor-enhanced pseudo-labeling loss  $L_u$  still follows the standard cross-entropy form [25] based on the confidence threshold.

$$L_u = \frac{1}{\mu B} \sum_{i=1}^{\mu B} (1(\max(\hat{y}_i^w) \geq \tau) H(y_i^w, P_{cls}(Aug_s(u_i))) + 1(\max(\hat{y}_i^w) < \tau) H(\tilde{y}_i^w, P_{cls}(Aug_s(u_i)))) \quad (10)$$

$L_u$  consists of two parts: the first part is the cross-entropy loss between the enhanced pseudo-labels based on aggregated predictions and the strongly augmented predictions; the second part is the cross-entropy loss between the sharpened predictions of weak augmentation and the strongly augmented predictions. Where  $H$  still represents the standard cross-entropy,  $\hat{y}_i^w = P_{cls}(Aug_w(u_i))$  represents the model's predictions for the weak augmentations of unlabeled samples  $u_i$ ,  $\tau$  denotes the confidence threshold, and the pseudo-labels  $y_i^w$  must satisfy the high confidence threshold  $\max(\hat{y}_i^w) > \tau$ .  $\tilde{y}_i^w = \frac{\exp(\hat{y}_i^w / \delta)}{\sum_k \exp(\hat{y}_k^w / \delta)}$  represents the sharpened predictions, and  $\delta$  is the sharpening coefficient. The model selects pseudo-labels  $y_i^w$  from weakly augmented samples with less information loss and more accurate predictions. Meanwhile, the model utilizes the classification predictions of strongly augmented samples  $\hat{y}_i^s = P_{cls}(Aug_s(u_i))$  to improve generalization.

Notably, bias analysis inspires us to improve the accuracy of pseudo-labels  $y_i^w$  by utilizing aggregated predictions  $\bar{y}_i^w$  from reliable neighbors (potential samples of the same class) of  $u_i$ . We initially employ the model to acquire the weakly augmented feature representation  $z_i^w$  and the prediction  $\hat{y}_i^w$ , then retrieve the weakly augmented neighbors of  $z_i$  and their labels. The candidate neighbors primarily come from two categories: all weakly augmented labeled samples and weakly augmented pseudo-labeled samples with predictions exceeding the confidence threshold (0.99 in this paper). Drawing on [6,8,12], these two types of data are maintained in reality by four memory banks: the labeled feature bank  $Z_l$ , the label bank  $Y_l$ , the pseudo-labeled feature stack  $Z_u$ , and the pseudo-label stack  $Y_u$ . With the exception of the label  $Y_l$ , the features and pseudo-labels are dynamically updated as the model undergoes training, and their quality is enhanced as the model performance improves. To exclude the interference from different classes of samples with low similarity, we select the top  $K_l$  labeled samples and the top  $K_u$  pseudo-labeled samples that are most similar to the sample  $z_i^w$ , and concatenate these samples to obtain the feature queue  $Z = [z_i^w, z_1^w, z_2^w, \dots, z_K^w]$  and the labeled queue  $Y = [\hat{y}_i^w, y_1, y_2, \dots, y_K]$ , where  $K = K_l + K_u$  and  $K_l$  equals  $K_u$  here. Next, we calculate the similarity matrix  $A$  on the  $K$ -nearest neighbor features  $Z$  as follows:

$$A_{ij} = \frac{\exp(z_i^w \cdot z_j / t)}{\sum_{k=1}^K \exp(z_i^w \cdot z_k / t)} \quad (11)$$

where  $t$  represents the temperature coefficient. The shape of matrix  $A$  is  $R^{((1+K)(1+K))}$ . Before applying softmax normalization to matrix  $A$ , we set all diagonal elements to zero (i.e.,  $A_{ij} = 0$  when  $i = j$ ) to eliminate the impact of self-loops. Lastly, we calculate the

label propagation of the label queue  $Y$  on the  $K$ -nearest neighbor graph represented by matrix  $A$ .

$$Y_\varphi = (\alpha \cdot A)^{\varphi-1}Y + (1 - \alpha) \sum_{k=1}^{\varphi-1} (\alpha \cdot A)^k Y \quad (12)$$

where  $Y$  represents the initial state  $[\hat{y}_i^{wv}, y_1, y_2, \dots, y_K]$ ,  $\alpha$  is a parameter ranging from 0 to 1 that assigns weight to the initial labels and aggregated labels, while  $\varphi$  represents the iterations of label propagation, and multiple iterations of label propagation tend to converge. Finally, the aggregated prediction  $\hat{y}_i^{wv}$  of the unlabeled sample  $u_i$  can be directly obtained from the first row vector of  $Y_\varphi$  (i.e.,  $Y_\varphi[0]$ ), and then used for confidence-based pseudo-label  $y_i^{wv}$ . Finally, the cross-entropy between the pseudo-label  $y_i^{wv}$  and the strong augmentation prediction  $\hat{y}_i^s = P_{cls}(Aug_s(u_i))$  is minimized to improve the model performance, as shown in Equation (10).

It is worth emphasizing that if  $\alpha = 1$  and  $\varphi = 1$ , then  $Y_1 = AY$ . Assuming  $A$  includes self-loops,  $Y_1$  is equivalent to the Equation (4) provided in the previous bias minimization analysis. In other words, Equation (4) corresponds to the special case where  $Y_\varphi$  propagates only once. However, label propagation  $Y_\varphi$  offers greater flexibility and scalability, and achieves more accurate and stable pseudo-labels through multiple iterations of label propagation. Differing from existing pseudo-labeling methods, neighbor-enhanced pseudo-labeling aggregates the labels of reliable labeled neighbors and the high-confidence pseudo-labels of unlabeled neighbors through a limited number of label propagation iterations to enhance the predictions of current samples.

#### 4.4. Variance Minimization

The label-enhanced contrastive loss is inspired by variance analysis, aiming to minimize the cross-entropy  $H(A, S)$  between the similarity matrix  $S$  and the reliable, complete adjacency matrix  $A$  on the same batch of labeled and unlabeled samples.

##### 4.4.1. Similarity Matrix $S$

Similar to [1,12], we randomly sample from the same batch of labeled images of size  $B$  and unlabeled images of size  $\mu B$ , respectively. Figure 2 illustrates the pipeline by which we obtain labeled feature views  $z_x^{s1}$  and  $z_x^{s2}$ , as well as unlabeled feature views  $z_u^{s1}$  and  $z_u^{s2}$ , from two types of strong augmentation,  $Aug_s(\cdot)$ . Subsequently, we apply the following metric method to measure pairwise sample similarity in the feature matrix  $Z = \{z_i : i = 1, \dots, 2(1 + \mu)B\}$ :

$$S_{ij} = \frac{\exp(z_i \cdot z_j / t)}{\sum_{j=1}^{2(1+\mu)B} \exp(z_i \cdot z_j / t)} \quad (13)$$

The similarity matrix  $S$  has a shape of  $R^{(2B+2\mu B) \times (2B+2\mu B)}$ , where  $t$  is the temperature factor. Unlike existing studies,  $Z$  consists of both labeled and unlabeled features.

##### 4.4.2. Target Adjacency Matrix $A$

The adjacency matrix  $A$  faces challenges in two aspects: data organization and numerical computation.

For the data organization of  $A$ , variance analysis inspires us to once again employ the label propagation algorithm from Section 4.3.2 to obtain aggregated predictions from high-quality neighbors, thereby enhancing the accuracy of pseudo-labels  $Y_u$  compared to existing pseudo-labeling methods based on individual predictions. Its advantage lies in effectively integrating the similarity and label information of high-quality neighbors to enhance the adjacency matrix  $A$ . In addition, we combine the enhanced pseudo-labels  $Y_u$  with the ground-truth labels  $Y_l$  from the same batch to jointly compute  $A$ . The computed

$A$  is more comprehensive and reliable compared to existing contrastive learning methods that rely solely on pseudo-labels.

For the numerical computation of  $A$ , it provides two key pieces of information: first, whether paired samples  $i$  and  $j$  belong to the same class ( $A_{ij} > 0$ ) or different classes ( $A_{ij} = 0$ ); second, the intimacy of the same-class relationship (i.e., the magnitude of the weight  $A_{ij}$ ). Therefore,  $A$  primarily consists of two major computational steps: clustering assignment and weight estimation. Take the example of estimating  $A_{ij}$  for any two samples  $z_i, z_j$ , their corresponding labels or predictions can be indexed from  $Y_l, Y_u$ . Firstly, we compute the maximum assignment probabilities  $p_i$  and  $p_j$ , along with their corresponding one-hot labels  $y_i$  and  $y_j$ . Subsequently, we utilize this information to calculate the clustering relationship  $c_{ij}$  and the correlation weight  $w_{ij}$  for  $z_i$  and  $z_j$ , respectively. Since the label  $Y_l$  itself represents a list of one-hot encoding labels, the assignment probability  $p_l$  can take a value within the range of  $[Y_\alpha, 1]$ .

In the clustering assignment step, we determine the clustering relationship  $c_{ij}$  between  $z_i$  and  $z_j$  by computing  $c_{ij} = y_i^T \cdot y_j$ , where  $c_{ij}$  is assigned a value of either 0 or 1.

In the weight estimation step, we further prioritize training on high-confidence positive pairs based on assignment probabilities, while mitigating the potential bias introduced by false positive pairs. The correlation weight  $w_{ij}$  between  $z_i$  and  $z_j$  is computed as  $w_{ij} = p_i \cdot p_j$ , and  $w_{ij}$  takes values from 0 to 1.

At this point,  $A_{ij} = c_{ij} \cdot w_{ij}$  can satisfy the aforementioned objective. However, we also draw on CCSSL [1] to supervise the quality of the pseudo-label, aiming to exclude samples with assignment probability  $p_i$  below a confidence threshold  $Y_\alpha$  from the training of positive pairs, so as to focus learning on clean data with high confidence. The specific approach involves appending an additional step of mask computation after the preceding two steps. For example, the mask calculation between  $z_i$  and  $z_j$  is given by  $m_{ij} = 1[p_i \geq Y_\alpha] \cdot 1[p_j \geq Y_\alpha]$ . Combining the aforementioned three steps, we compute  $A_{ij} = c_{ij} \cdot w_{ij} \cdot m_{ij}$ . It is worth emphasizing that we set  $c_{ii} = 0$  to avoid self-loop, while we set the clustering relationship between different views of the same sample as  $c_{ij} = 1$ . Intuitively,  $A_{ij}$  is a sparse matrix with the following numerical distribution.

$$A_{ij} = \begin{cases} p_i \cdot p_j, & \text{if } i \neq j \text{ and the one-hot labels } y_i \text{ and } y_j \\ & \text{are the same, with both probabilities } p_i \\ & \text{and } p_j \text{ higher than the threshold } Y_\alpha. \\ 0, & \text{otherwise} \end{cases} \quad (14)$$

When dealing with batch data in practice, the aforementioned steps yield the clustering relationship matrix  $C$ , the correlation weight matrix  $W$ , and the mask matrix  $M$ , respectively. Therefore, the overall  $A$  is calculated as:

$$A = C \cdot W \cdot M \quad (15)$$

As the target adjacency matrix in Figure 2,  $A$  provides reliable neighbor information to effectively guide contrastive learning in bringing similar features closer while pushing dissimilar features farther apart, thereby achieving desirable feature representations.

#### 4.4.3. Label-Enhanced Contrastive Loss

The proposed label-enhanced contrastive learning aims to minimize the cross-entropy between the similarity matrix  $S$  and the adjacency matrix  $A$ .

$$L_c = H(A, S) = \sum_{i=1}^{2(1+\mu)B} \frac{1}{|P(i)|} L_{c,i} \quad (16)$$

Whereas  $H$  still represents the cross-entropy,  $P(i)$  denotes the indices of views belonging to the same class as sample  $i$ , and the cardinality  $|P(i)|$  represents the total number of positive sample pairs. The specific form of  $L_{c,i}$  is as follows:

$$L_{c,i} = - \sum_{p \in P(i)} A_{ip} \cdot \log \left( \frac{\exp(z_i \cdot z_p / \tau)}{\sum_{j=1}^{2(1+\mu)B} \exp(z_i \cdot z_j / \tau)} \right) \quad (17)$$

Unlike existing work,  $L_c$  incorporates ground-truth labels with enhanced pseudo-labels to compute a relatively accurate and complete adjacency matrix  $A$  for enhancing feature representation.

#### 4.5. Overall Loss Function

The total loss  $L$  is the weighted sum of the supervised loss  $L_x$ , the neighbor-enhanced pseudo-labeling loss  $L_u$ , and the label-enhanced contrastive loss  $L_c$ , as shown in Equation (18).

$$L = L_x + \lambda_u L_u + \lambda_c L_c \quad (18)$$

Similar to CCSSL [1], the weight coefficients  $\lambda_u$  and  $\lambda_c$  are used to balance the bias term represented by  $L_u$  and the variance term represented by  $L_c$ , respectively.

## 5. Experimental Results

### 5.1. Experimental Setup

#### 5.1.1. Common SSL Datasets

We experimentally evaluate our GBVSSL on mainstream SSL datasets: CIFAR-10/100 [26], SVHN [27], STL-10 [28]. The various numbers of labeled data settings for each dataset are presented in Table 1.

#### 5.1.2. Implementation Details

For a fair comparison, our model architecture employs WideResNet (WRN)-28-2 [29] for CIFAR-10, WRN-28-8 for CIFAR-100 and SVHN, and ResNet-18 network [30] for STL-10, in accordance with [6]. Meanwhile, we incorporate a 2-layer MLP projection head after the backbone network for contrastive learning in a 64-dimensional embedding space [31]. We utilize an SGD optimizer with momentum 0.9 and Nesterov momentum [32] for all experiments, along with a scheduler that implements cosine learning rate decay [33]. We set the initial learning rate to 0.03 and adjust the training epoch to 512 instead of the 1024 epoch used in [34] to showcase our faster convergence efficiency. All experiments are conducted on a single GPU. The weak and strong data augmentations utilized in the experiments are primarily inherited from the USB [35] codebase. Regarding hyperparameters, we set the default values as follows:  $\lambda_u = 1$ , batch size  $B = 64$ , the ratio of unlabeled data  $\mu = 7$ , label propagation coefficient  $\varphi = 3$ ,  $\alpha = 0.1$ , temperature coefficient  $t = 0.07$ , confidence threshold  $\tau = 0.95$ , and the number of neighbors  $K = 128$ . Regarding specific parameters, we assign  $\lambda_c = 0.2$  and  $Y_\alpha = 0$  for the simple CIFAR-10 and set  $\lambda_c = 1.0$  and  $Y_\alpha = 0$  for the relatively complex STL-10, CIFAR-100, and SVHN. In addition, we maintain four memory banks that store labeled sample features, ground-truth labels, pseudo-labeled sample features, and pseudo-labels, providing high-quality neighbors for pseudo-label estimation.

#### 5.1.3. Baseline Methods

We explore the state-of-the-art semi-supervised learning methods, including mainstream methods based on consistency regularization, such as MixMatch [36], FixMatch [2], and FlexMatch [3]. To validate the performance improvements of the proposed GBVSSL method, we also compare it with previous SSL methods based on contrastive learning, including CoMatch [6], CCSSL [1], and SimMatch [8], etc. Since CCSSL is the most similar to our GBVSSL, it is used as the benchmark for comparison in nearly all experiments.

**Table 1.** Error rate for CIFAR-10/100, SVHN, and STL-10 datasets on 3 different labeled data settings. **Bold** highlights optimal outcome and underline denotes the second-best performance.

Dataset	CIFAR-10			CIFAR-100			SVHN			STL-10		
	Label Amount	40	250	4000	400	2500	10,000	40	250	1000	40	250
IT Model [37]	74.34 ± 1.76	46.24 ± 1.29	13.13 ± 0.59	86.96 ± 0.80	58.80 ± 0.66	36.65 ± 0.00	67.48 ± 0.95	13.30 ± 1.12	7.16 ± 0.11	74.31 ± 0.85	55.13 ± 1.50	32.78 ± 0.40
Pseudo Label [9]	74.61 ± 0.26	46.49 ± 2.20	15.08 ± 0.19	87.45 ± 0.85	57.74 ± 0.28	36.55 ± 0.24	64.61 ± 5.60	15.59 ± 0.95	9.40 ± 0.32	74.68 ± 0.99	55.45 ± 2.43	32.64 ± 0.71
VAT [38]	74.66 ± 2.12	41.03 ± 1.79	10.51 ± 0.12	85.20 ± 1.40	46.84 ± 0.79	32.14 ± 0.19	74.75 ± 3.38	4.33 ± 0.12	4.11 ± 0.20	74.74 ± 0.38	56.42 ± 1.97	37.95 ± 1.12
Mean Teacher [39]	70.09 ± 1.60	37.46 ± 3.30	8.10 ± 0.21	81.11 ± 1.44	45.17 ± 1.06	31.75 ± 0.23	36.09 ± 3.98	3.45 ± 0.03	3.27 ± 0.05	71.72 ± 1.45	56.49 ± 2.75	33.90 ± 1.37
MixMatch [36]	38.84 ± 8.36	20.96 ± 2.45	10.25 ± 0.01	80.58 ± 3.38	47.88 ± 0.21	33.22 ± 0.06	26.61 ± 13.10	4.48 ± 0.35	5.01 ± 0.12	52.32 ± 0.91	36.34 ± 0.84	25.01 ± 0.43
ReMixMatch [4]	8.13 ± 0.58	6.34 ± 0.22	4.65 ± 0.09	41.60 ± 1.48	25.72 ± 0.07	20.04 ± 0.13	16.43 ± 13.77	5.65 ± 0.35	5.36 ± 0.58	27.87 ± 3.85	11.14 ± 0.52	6.44 ± 0.15
FixMatch [2]	12.66 ± 4.49	4.95 ± 0.10	4.26 ± 0.01	45.38 ± 2.07	27.71 ± 0.42	22.06 ± 0.10	<u>3.37 ± 1.01</u>	<b>1.97 ± 0.01</b>	2.02 ± 0.03	38.19 ± 4.76	8.64 ± 0.84	5.82 ± 0.06
FlexMatch [3]	5.29 ± 0.29	4.97 ± 0.07	4.24 ± 0.06	40.73 ± 1.44	26.17 ± 0.18	21.75 ± 0.15	8.19 ± 3.20	6.59 ± 2.29	6.72 ± 0.30	29.12 ± 5.04	9.85 ± 1.35	6.08 ± 0.34
Dash [14]	9.29 ± 3.28	5.16 ± 0.28	4.36 ± 0.10	47.49 ± 1.05	27.47 ± 0.38	21.89 ± 0.16	5.26 ± 2.02	2.01 ± 0.01	2.08 ± 0.09	42.00 ± 4.94	10.50 ± 1.37	6.30 ± 0.49
MPL [40]	6.93 ± 0.17	5.76 ± 0.24	4.55 ± 0.04	46.26 ± 1.84	27.71 ± 0.19	21.74 ± 0.09	-	-	-	35.76 ± 4.83	9.90 ± 0.96	6.66 ± 0.00
RelationMatch [41]	6.87 ± 0.12	<b>4.85 ± 0.04</b>	4.22 ± 0.06	45.79 ± 0.59	27.90 ± 0.15	22.18 ± 0.13	-	-	-	33.42 ± 3.92	9.55 ± 0.87	6.08 ± 0.29
CoMatch [6]	6.51 ± 1.18	5.35 ± 0.14	4.27 ± 0.12	53.41 ± 2.36	29.78 ± 0.11	22.11 ± 0.22	8.20 ± 5.32	2.16 ± 0.04	<u>2.01 ± 0.04</u>	<b>13.74 ± 4.20</b>	<u>7.63 ± 0.94</u>	5.71 ± 0.08
CCSSL [1]	9.71 ± 2.78	5.14 ± 0.55	4.46 ± 0.20	38.81 ± 1.65	<u>24.30 ± 0.63</u>	<u>19.32 ± 0.16</u>	7.85 ± 3.6	2.12 ± 0.04	<u>2.03 ± 0.03</u>	17.55 ± 4.2	8.43 ± 1.1	5.77 ± 0.82
SimMatch [8]	5.38 ± 0.01	5.36 ± 0.08	4.41 ± 0.07	39.32 ± 0.72	26.21 ± 0.37	21.50 ± 0.11	7.60 ± 2.11	2.48 ± 0.61	2.05 ± 0.05	16.98 ± 4.24	8.27 ± 0.40	5.74 ± 0.31
AdaMatch [15]	<u>5.09 ± 0.21</u>	5.13 ± 0.05	4.36 ± 0.05	38.08 ± 1.35	26.66 ± 0.33	21.99 ± 0.15	6.14 ± 5.35	2.13 ± 0.04	2.02 ± 0.05	19.95 ± 5.17	8.59 ± 0.43	6.01 ± 0.02
FreeMatch [18]	<b>4.90 ± 0.12</b>	<u>4.88 ± 0.09</u>	<u>4.16 ± 0.06</u>	39.52 ± 0.01	26.22 ± 0.08	21.81 ± 0.17	10.43 ± 0.82	8.23 ± 3.22	7.56 ± 0.25	28.50 ± 5.41	9.29 ± 1.24	5.81 ± 0.32
SoftMatch [5]	5.11 ± 0.14	4.96 ± 0.09	<u>4.27 ± 0.05</u>	<u>37.60 ± 0.24</u>	26.39 ± 0.38	21.86 ± 0.16	<b>2.46 ± 0.24</b>	2.15 ± 0.07	2.09 ± 0.06	22.23 ± 3.82	9.18 ± 0.68	5.79 ± 0.15
GBVSSL	6.15 ± 0.25	4.92 ± 0.05	<b>3.98 ± 0.10</b>	<b>37.27 ± 0.28</b>	<b>23.25 ± 0.33</b>	<b>18.86 ± 0.24</b>	5.21 ± 1.05	<u>1.99 ± 0.31</u>	<b>1.89 ± 0.04</b>	<u>15.74 ± 3.90</u>	<b>7.37 ± 0.86</b>	<b>5.63 ± 0.35</b>
Fully-Supervised		4.62 ± 0.05			19.30 ± 0.09			2.13 ± 0.02			None	



#### 5.1.4. Data Augmentation

For the data augmentation strategy for all experiments, we follow CCSSL [1] and employ one “weak” augmentation  $Aug_w$  and two “strong” augmentations  $Aug_{s1}$  and  $Aug_{s2}$ . For the “weak” augmentation  $Aug_w$ , we follow FixMatch’s standard cropping and flipping methods. As for the “strong” augmentation methods  $Aug_{s1}$  and  $Aug_{s2}$ , we adopt the augmentation strategies from RandAugment [11] and MoCo [12] (random color jittering and grayscale transformation) to ensure fairness in the comparison.

#### 5.2. Performance on Common SSL Datasets

We compare GBVSSL with 17 state-of-the-art semi-supervised learning methods, and the results are shown in Table 1. Overall, GBVSSL exhibits the best performance for most settings across all datasets. Specifically, GBVSSL first demonstrates comparable performance to other methods on CIFAR-10. Despite the performance increase being only  $-1.25\%$ / $-0.07\%$ / $0.18\%$  compared to the best SSL methods, GBVSSL achieves performance improvements of 3.56%, 0.22%, and 0.48% over CCSSL, significantly enhancing performance in the few-shot sample setting. For the relatively complex CIFAR-100 dataset, GBVSSL achieves the best performance in settings with 400, 2500, and 10,000 labels, exhibiting performance improvements of 1.54%, 0.85%, and 0.46% over CCSSL, respectively. GBVSSL exhibits a substantial performance enhancement in settings with fewer labeled samples, primarily attributed to the enhanced reliability of its pseudo-labels. For the imbalanced dataset SVHN, it is worth noting that FlexMatch performs poorly. This is because it registers too many incorrect pseudo-labels during the initial stages of training, which misleads the subsequent learning process. Nonetheless, GBVSSL still performs well on SVHN, demonstrating its ability to effectively utilize pseudo-labels, which helps alleviate overfitting issues on small-scale and imbalanced datasets. Lastly, the STL-10 dataset poses greater practical challenges due to the significant disparity in distribution between the unlabeled and labeled data. GBVSSL ranks second with 40 labeled samples, but achieves the best performance with 250 and 1000 labeled samples. Compared to CCSSL, GBVSSL demonstrates strong adaptability to the unlabeled data distribution, with performance gains of 1.81%, 1.06%, and 0.14%. In conclusion, the experimental results demonstrate that GBVSSL has strong generalization capabilities in semi-supervised learning, particularly suitable for handling imbalanced datasets prone to overfitting, such as SVHN and STL-10.

#### 5.3. Results on Semi-iNat 2021 [42]

Additionally, we evaluate the performance of GBVSSL on the Semi-iNat 2021 dataset. Semi-iNat 2021 is a highly complex real-world dataset, posing challenging obstacles for SSL, such as imbalanced class distribution, domain mismatch between labeled and unlabeled data, and the presence of out-of-distribution examples. It comprises 9721 labeled data, 313,248 unlabeled data, and 4050 validation data across 810 different categories. As a realistic dataset benchmark, Semi-iNat 2021 can objectively reflect the performance of various methods. The experiment follows the settings of CCSSL [1], resizes images to  $224 \times 224$ , employs a ResNet-50 backbone network, and is equipped with a projection head consisting of two layers and outputting 64 dimensions. To ensure fairness in comparison, we adopt the same parameter settings as CCSSL and HyperMatch [31], including  $\tau = 0.6$ ,  $B = 64$ ,  $\mu = 7$ ,  $\lambda_u = 1$ ,  $\lambda_c = 2$ , and  $t = 0.07$ . For the specific parameters of GBVSSL, we maintain  $\varphi = 3$ ,  $\alpha = 0.1$ ,  $K = 128$ , and  $Y_\alpha = 0.9$ . It is worth emphasizing that we follow CCSSL [1] and report the performance of GBVSSL under two experimental settings: training from scratch and training from a pre-trained MoCo [12] model.

The results are shown in Table 2. In the setting of training from scratch, GBVSSL achieves a Top-1 accuracy of 35.59%, surpassing the previous state-of-the-art method HyperMatch, with a Top-1 performance improvement of +4.38% compared to the optimal CCSSL setup (FixMatch + CCSSL). In the setting based on the MoCo pre-trained model, GBVSSL outperforms the HyperMatch approach with a Top-1 accuracy of 44.52%, showcasing an impressive improvement of +8.93% compared to the training-from-scratch setting.

GBVSSL outperforms CCSSL(FixMatch) with a Top-1 accuracy improvement of +3.24%, further highlighting its superiority in handling out-of-distribution data.

**Table 2.** Comparison of Top-1 and Top-5 accuracy for Semi-iNat 2021 dataset.

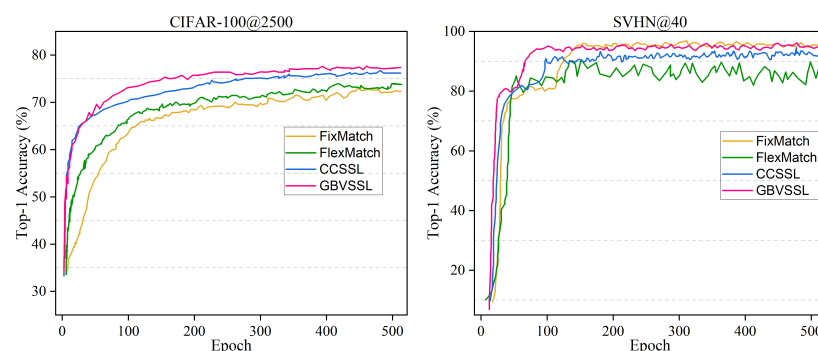
Method	Semi-iNat 2021			
	From Scratch		MoCo Pretrain	
	Top1	Top5	Top1	Top5
Supervised	19.09	35.85	34.96	57.11
MixMatch [36]	16.89	30.83	-	-
CoMatch [6]	20.94	38.96	38.94	61.85
FixMatch [2]	21.41	37.65	40.3	60.05
CCSSL(MixMatch) [1]	19.65	35.09	-	-
CCSSL(CoMatch) [1]	24.12	43.23	39.85	63.68
CCSSL(FixMatch) [1]	31.21	52.25	41.28	64.3
HyperMatch [31]	33.47	-	41.57	-
<b>GBVSSL</b>	<b>34.12</b>	<b>55.67</b>	<b>43.65</b>	<b>66.83</b>

#### 5.4. Qualitative Studies

We conduct qualitative analysis on GBVSSL and compare it with representative benchmark methods such as FixMatch, FlexMatch, and CCSSL to gain a deeper understanding of GBVSSL's distinctions and advantages relative to existing approaches.

##### 5.4.1. Convergence Speed

Figure 3 illustrates the fluctuation in GBVSSL's Top-1 accuracy on the CIFAR-100 and SVHN datasets throughout the training process, and compared to that of FixMatch, FlexMatch, and CCSSL. In comparison to FixMatch and FlexMatch, we observe that CCSSL and GBVSSL converge to higher performance levels at significantly faster rates. Compared to CCSSL, GBVSSL further accelerates the convergence speed. Additionally, GBVSSL outperforms CCSSL on CIFAR-100 with 2500 labels, while it performs comparably to FixMatch on SVHN with 40 labels. This is mainly attributed to the enhanced accuracy and robustness of pseudo-labels achieved through the utilization of aggregated predictions. Higher-quality pseudo-labels can effectively guide the model's training process and improve the learning quality, thereby resulting in faster convergence speed and superior performance.

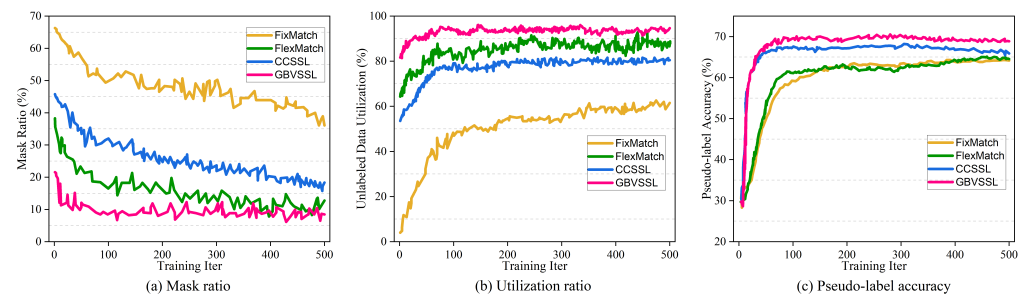


**Figure 3.** The Top-1 accuracy trends on CIFAR-100 with 2500 labels and SVHN with 40 labels demonstrate that GBVSSL surpasses other baselines with fewer training epochs.

##### 5.4.2. Mask Ratio, Data Utilization, and Pseudo-Label Accuracy

We compare the mask ratio, utilization rate, and pseudo-label accuracy of GBVSSL with those of FixMatch, FlexMatch, and CCSSL on the CIFAR-100 dataset with 400 labels, as

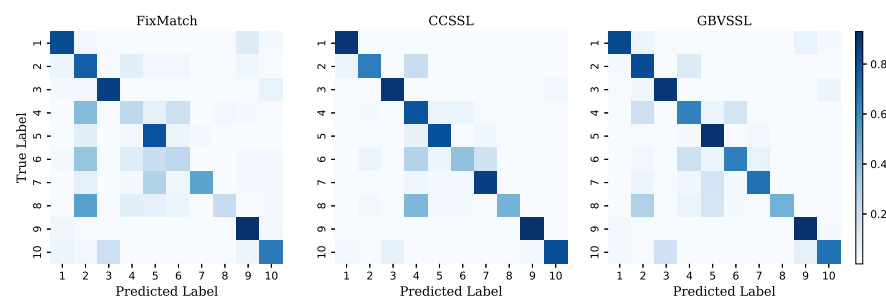
shown in Figure 4. It is worth emphasizing that the mask rate is defined as the proportion of pseudo-labeled samples that do not participate in model training at epoch  $t$  due to confidence masking, pseudo-boundary masking, or both. The utilization rate is defined as the proportion of unlabeled data effectively utilized by the model in epoch  $t$ . Pseudo-label accuracy is defined as the accuracy of pseudo-labels generated from unlabeled data during training. Typically, effective semi-supervised learning models are expected to fully leverage unlabeled data to enhance performance, demonstrating lower mask ratio and higher data utilization. Indeed, Figure 4a illustrates that GBVSSL significantly reduces the masked data ratio, while Figure 4b demonstrates that the data utilization of GBVSSL surpasses that of FixMatch, FlexMatch, and CCSSL. Moreover, we observe that GBVSSL demonstrates significantly higher pseudo-label accuracy on unlabeled data compared to FixMatch, FlexMatch, and CCSSL, as shown in Figure 4c. Experiments demonstrate that GBVSSL not only guarantees lower mask ratio and higher data utilization, but also maintains higher pseudo-label accuracy, effectively showcasing the superiority of our neighbor-enhanced pseudo-labeling approach.



**Figure 4.** The (a) mask ratio, (b) utilization ratio, (c) pseudo-label accuracy of unlabeled data on CIFAR-100 dataset with 400 labels.

### 5.4.3. Confusion Matrix

The rows and columns of the confusion matrix represent the true and predicted labels, respectively, illustrating the model's classification of the samples. As shown in Figure 5, we compare the confusion matrices of FixMatch, CCSSL, and GBVSSL on the STL-10 dataset with 40 labels. Among them, FixMatch overfits to minority classes and fails to recognize the 4th, 6th, and 8th classes. Compared to FixMatch, both CCSSL and GBVSSL effectively reduce confusion among similar classes and correctly identify the 4th class. GBVSSL performs comparably to CCSSL overall, with only a slight improvement in the recognition accuracy of the 6th class. The experiment intuitively demonstrates that GBVSSL possesses strong class discrimination capabilities, thereby enhancing the accuracy of pseudo-labels and improving classification performance.

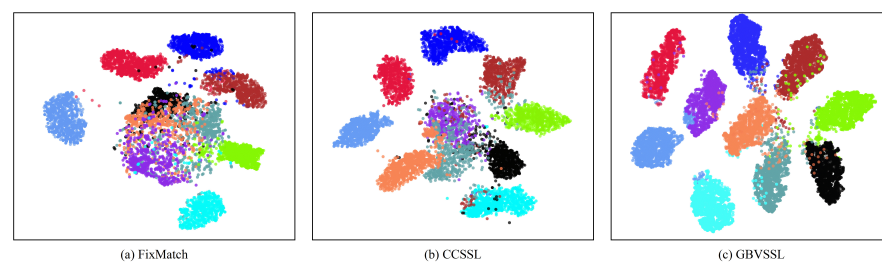


**Figure 5.** Confusion matrices for FixMatch, CCSSL, and GBVSSL on STL-10 dataset with 40 labels.

### 5.4.4. T-SNE Visualization

The label-enhanced contrastive module is essentially a graph contrastive module based on an adjacency matrix. It facilitates the model learn discriminative feature distributions, where the features of samples within the same class are encouraged to be as close as possible,

while the features of samples from different classes are encouraged to be as far apart as possible. To validate this claim, we present t-SNE visualizations of the features extracted by GBVSSL on the STL-10 dataset with 40 labels, and compare them with FixMatch and CCSSL, as shown in Figure 6. Certainly, the intra-class features of GBVSSL are tightly clustered together, while the inter-class boundaries are clearer and easier to separate. Compared to FixMatch and CCSSL, GBVSSL achieves a superior clustering distribution, thereby yielding more precise predictions and pseudo-labels, consequently enhancing the model's generalization performance. Furthermore, the strong cohesion of features within each cluster in GBVSSL suggests reduced bias and intra-class variance in classification predictions, thus validating to some extent the correctness and effectiveness of the bias and variance minimization scheme proposed in this study.



**Figure 6.** T-SNE visualization of features for FixMatch, CCSSL, and GBVSSL on the STL-10 dataset.

### 5.5. Ablation Experiments

#### 5.5.1. Contrastive & Class-Aware & Re-Weighting & Combination of Label and Pseudo-Label & Label Propagation

We investigate each technique for label-enhanced contrastive loss, with results depicted in Table 3. The label-enhanced contrastive loss inherits three foundational techniques: self-supervised contrastive loss, class-aware contrastive loss, and reweighting from FixMatch. It also introduces a hybrid contrastive strategy for labeled and unlabeled data, along with a refinement technique that enhances pseudo-labels using label propagation. For convenience, these five techniques are respectively referred to as ss-cl, ca-cl, re-weight, label-unlabel, and label prop. The experimental results demonstrate that all technical components are useful. It is worth noting that using ss-cl alone leads to a performance degradation on the intra-distribution dataset CIFAR-100. CCSSL successfully integrates the techniques of ca-cl and re-weight, effectively balancing the advantages of contrast and clustering, thereby enhancing the performance on these two datasets. GBVSSL effectively enhances performance by applying the label-unlabel technique to ca-cl, and then slightly improves performance by combining the re-weighting strategy. In the end, GBVSSL successfully improves the accuracy of pseudo-labels by leveraging label propagation (label prop) to obtain aggregated predictions of true labels or reliable pseudo-labels from high-similarity neighbors, leading to the best results.

**Table 3.** Performance evaluation of different combinations of technical components in label-enhanced contrastive learning.

ss-cl	ca-cl	Re-Weight	Label-Unlabel	Label Prop	Semi-iNat 2021	CIFAR-100@2500
					21.58	72.69
✓					27.86	72.45
✓		✓			29.66	72.82
	✓				30.62	75.71
	✓	✓			31.49	76.09
	✓		✓		32.09	76.32
	✓	✓	✓		32.91	76.57
	✓	✓	✓	✓	<b>34.14</b>	<b>76.93</b>

### 5.5.2. Ratio $\mu$ of Unlabeled Data

Table 4 presents the experimental results of GBVSSL on the CIFAR-100 dataset using various values of  $\mu$ . Since  $\mu$  directly determines the ratio of unlabeled data to labeled data within a batch, it indirectly influences the size and reliability of the adjacency matrix  $A$  in label-enhanced contrastive learning. Experimental observations indicate that GBVSSL achieves optimal performance in settings with a small number of unlabeled samples when label-enhanced contrastive learning fully trusts and utilizes both labels and pseudo-labels to construct the adjacency matrix. However, when we increase the impact of the confidence threshold by adjusting  $Y_\alpha$  to 0.4 (samples with  $Y_\alpha \leq 0.4$  only undergo self-supervised contrastive learning), GBVSSL achieves better results at a higher ratio  $\mu$  ( $\mu = 7$ ). Our analysis suggests that label-enhanced contrastive learning relies on reliable neighborhood relationships (represented by an adjacency matrix). When more unlabeled data are used, the size of the adjacency matrix increases, effectively enhancing the smoothness and robustness of distribution prediction. However, this also introduces additional challenges in dealing with noise. Therefore, GBVSSL achieves optimal performance at a larger ratio  $\mu$  by filtering noise through the threshold condition  $Y_\alpha$ .

**Table 4.** Experimenting with various ratios ( $\mu$ ) of unlabeled data using different confidence threshold ( $Y_{alpha}$ ) on the CIFAR-100 dataset with 10,000 labels. When fully trusting the pseudo-labels with  $Y_{alpha} = 0$ , smaller ratios ensure better performance due to noise in unlabeled samples. After setting  $Y_{alpha} = 0.4$  to filter out noise, larger ratio yields better performance.

Ratio ( $\mu$ )	CIFAR-100@10000	
	$Y_\alpha = 0$	$Y_\alpha = 0.4$
2	77.25	77.63
4	77.96	79.32
5	<b>79.46</b>	79.84
6	76.86	79.58
7	77.43	<b>81.15</b>

### 5.5.3. Memory Bank Setup

The memory bank setup directly determines the selection range and quality of neighboring samples, thereby indirectly impacting the reliability of aggregated predictions and their pseudo-labels. Here, we examine the performance of GBVSSL across three memory bank settings: pseudo-label memory bank, label memory bank, and a combination of both. The experimental results are elaborated in Table 5. We observe that Semi-iNat 2021 demonstrates superior performance on the pseudo-label memory bank compared to the labeled memory bank. We determine that unlabeled data are more beneficial for pseudo-label estimation in the out-of-distribution dataset Semi-iNat 2021. For the in-distribution dataset CIFAR-100, GBVSSL outperforms the labeled memory bank when utilizing the pseudo-labeled memory bank. This indicates that ground-truth labels are more effective in enhancing the pseudo-labels of the in-distribution dataset. In conclusion, the combination of both pseudo-labeled and labeled memory banks is more beneficial for GBVSSL to achieve optimal performance on both datasets compared to using either memory bank alone.

**Table 5.** Effect of different memory bank settings on performance at  $K = 128$ .

Bank Settings	Unlabeled	Labeled	Unlabeled & Labeled
Semi-iNat 2021	33.65	33.08	<b>34.12</b>
CIFAR-100@2500	76.32	76.69	<b>76.93</b>

### 5.5.4. Top-K Selections

To further enhance the quality of neighbors (reduce noise) and lessen the computational burden of label propagation, GBVSSL selects only the top  $K$  reliable neighbors for



each pseudo-labeled sample. We test the impact of different values of  $K$  on the performance of GBVSSL. The evidence presented in Table 6 indicates that GBVSSL exhibits robust performance across varying values of  $K$ , demonstrating its insensitivity to the parameter  $K$ . A moderate value of  $K$  (recommended 128) is preferable, as it exhibits a difference of no more than 0.57 between its best and worst performance. Notably, Table 6 shows the model's accuracy, unlike the error rate presented in Table 1. The relationship between these metrics is: Error Rate = 1 – Accuracy.

**Table 6.** Effect of the number of Top- $K$  neighbors. (We use both labeled and unlabeled memory banks).

$K$	8	16	32	64	128	256
Semi-iNat 2021	33.57	33.85	34.03	<b>34.14</b>	34.12	33.94
CIFAR-100@2500	76.66	76.54	76.86	76.82	<b>76.93</b>	76.75

### 5.5.5. Label Propagation Iterations $\phi$

We evaluate the impact of label propagation iterations  $\phi$  on the performance of GBVSSL. The results in Table 7 show that GBVSSL achieves better performance with label propagation ( $\phi > 0$ ) compared to without label propagation ( $\phi = 0$ ). Specifically, for the out-of-distribution dataset Semi-iNat 2021, label propagation contributes to a performance gain of +1.22% for the GBVSSL model compared to not using label propagation. Even for the in-distribution dataset CIFAR-100, label propagation still provides a performance advantage of +0.63% compared to its absence.

**Table 7.** Effect of the number of label propagation iterations  $\phi$  in Equation (12).

$\phi$	$Y_{\phi=0}$	$Y_{\phi=1}$	$Y_{\phi=2}$	$Y_{\phi=3}$
Semi-iNat 2021	32.92	33.57	<b>34.14</b>	33.85
CIFAR-100@2500	76.30	76.62	76.70	<b>76.93</b>

## 6. Limitations

This study is constrained by its inability to investigate stronger augmentation techniques and more advanced pipelines for semi-supervised learning. These limitations primarily arise from the lack of sufficient experimental resources required to determine the optimal model configuration.

## 7. Conclusions

We introduce a generalized bias-variance decomposition framework for delving into the theoretical mechanisms of mainstream semi-supervised learning techniques, such as pseudo-labeling and contrastive learning. The framework inspires us to propose neighbor-enhanced pseudo-labeling and label-enhanced contrastive learning, aiming to address the shortcomings of related techniques. Finally, we combine these two techniques to develop a new semi-supervised learning method, GBVSSL, which effectively enhances the pseudo-labels and feature representations. Extensive experiments validate the state-of-the-art performance of GBVSSL on multiple SSL benchmarks, as well as the effectiveness of each module. This work not only contributes to the understanding of semi-supervised learning but also provides promising directions for incorporating generalized bias-variance decomposition into future research.

**Author Contributions:** Conceptualization, S.L. and L.H.; methodology, S.L.; software, S.L. and Y.W.; validation, S.L., L.H. and J.Z.; formal analysis, S.L.; investigation, S.L.; resources, L.H. and Y.W.; data curation, S.L. and J.Z.; writing—original draft preparation, S.L.; writing—review and editing, S.L. and L.H.; visualization, S.L.; supervision, L.H.; project administration, L.H.; funding acquisition, Y.W. All authors have read and agreed to the published version of the manuscript.

**Funding:** The paper was partially supported by the Natural Science Foundation of Colleges and Universities in Anhui Province of China (No. KJ2021A0640).

**Data Availability Statement:** The raw data supporting the conclusion of this article will be made available by the authors on request.

**Acknowledgments:** The authors thank the anonymous reviewers for their valuable and constructive comments that greatly helped improve the quality and completeness of this paper.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## References

1. Yang, F.; Wu, K.; Zhang, S.; Jiang, G.; Liu, Y.; Zheng, F.; Zhang, W.; Wang, C.; Zeng, L. Class-aware contrastive semi-supervised learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 14421–14430.
2. Sohn, K.; Berthelot, D.; Carlini, N.; Zhang, Z.; Zhang, H.; Raffel, C.A.; Cubuk, E.D.; Kurakin, A.; Li, C.L. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 596–608.
3. Zhang, B.; Wang, Y.; Hou, W.; Wu, H.; Wang, J.; Okumura, M.; Shinozaki, T. Flexmatch: Boosting semi-supervised learning with curriculum pseudo labeling. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 18408–18419.
4. Berthelot, D.; Carlini, N.; Cubuk, E.D.; Kurakin, A.; Sohn, K.; Zhang, H.; Raffel, C. ReMixMatch: Semi-Supervised Learning with Distribution Alignment and Augmentation Anchoring. In Proceedings of the International Conference on Learning Representations, Online, 26 April–1 May 2020; pp. 1–13.
5. Chen, H.; Tao, R.; Fan, Y.; Wang, Y.; Wang, J.; Schiele, B.; Xie, X.; Raj, B.; Savvides, M. Softmatch: Addressing the quantity-quality trade-off in semi-supervised learning. *arXiv* **2023**, arXiv:2301.10921.
6. Li, J.; Xiong, C.; Hoi, S.C. Comatch: Semi-supervised learning with contrastive graph regularization. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, QC, Canada, 10–17 October 2021; pp. 9475–9484.
7. Kim, J.; Min, Y.; Kim, D.; Lee, G.; Seo, J.; Ryoo, K.; Kim, S. Conmatch: Semi-supervised learning with confidence-guided consistency regularization. In Proceedings of the European Conference on Computer Vision, Tel Aviv, Israel, 23–27 October 2022; pp. 674–690.
8. Zheng, M.; You, S.; Huang, L.; Wang, F.; Qian, C.; Xu, C. SimMatch: Semi-Supervised Learning with Similarity Matching. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022; pp. 14471–14481.
9. Lee, D.H. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In Proceedings of the Workshop on Challenges in Representation Learning, ICML, Atlanta, GA, USA, 16–21 June 2013; Volume 3, p. 896.
10. Wu, Z.; Xiong, Y.; Yu, S.X.; Lin, D. Unsupervised feature learning via non-parametric instance discrimination. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 3733–3742.
11. Chen, X.; Fan, H.; Girshick, R.; He, K. Improved baselines with momentum contrastive learning. *arXiv* **2020**, arXiv:2003.04297.
12. He, K.; Fan, H.; Wu, Y.; Xie, S.; Girshick, R. Momentum contrast for unsupervised visual representation learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 9729–9738.
13. Xie, Q.; Dai, Z.; Hovy, E.; Luong, T.; Le, Q. Unsupervised data augmentation for consistency training. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 6256–6268.
14. Xu, Y.; Shang, L.; Ye, J.; Qian, Q.; Li, Y.F.; Sun, B.; Li, H.; Jin, R. Dash: Semi-supervised learning with dynamic thresholding. In Proceedings of the International Conference on Machine Learning, Virtual Event, 18–24 July 2021; pp. 11525–11536.
15. Roelofs, B.; Berthelot, D.; Sohn, K.; Carlini, N.; Kurakin, A. AdaMatch: A Unified Approach to Semi-Supervised Learning and Domain Adaptation. In Proceedings of the International Conference on Learning Representations (ICLR), Virtual Event, 25–29 April 2022; pp. 1–50.
16. Guo, L.Z.; Li, Y.F. Class-imbalanced semi-supervised learning with adaptive thresholding. In Proceedings of the International Conference on Machine Learning, Baltimore, MD, USA, 17–23 July 2022; pp. 8082–8094.
17. Wang, X.; Wu, Z.; Lian, L.; Yu, S.X. Debaised learning from naturally imbalanced pseudo-labels. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 14647–14657.
18. Wang, Y.; Chen, H.; Heng, Q.; Hou, W.; Fan, Y.; Wu, Z.; Wang, J.; Savvides, M.; Shinozaki, T.; Raj, B.; et al. Freematch: Self-adaptive thresholding for semi-supervised learning. *arXiv* **2022**, arXiv:2205.07246.
19. Zhou, H.; Song, L.; Chen, J.; Zhou, Y.; Wang, G.; Yuan, J.; Zhang, Q. Rethinking soft labels for knowledge distillation: A bias-variance tradeoff perspective. In Proceedings of the International Conference on Learning Representations, Online, 3–7 May 2021; pp. 1–15.
20. Heskes, T. Bias/variance decompositions for likelihood-based estimators. *Neural Comput.* **1998**, *10*, 1425–1433. [[CrossRef](#)] [[PubMed](#)]
21. Kipf, T.N.; Welling, M. Semi-Supervised Classification with Graph Convolutional Networks. In Proceedings of the International Conference on Learning Representations (ICLR), Toulon, France, 24–26 April 2017.
22. Zhou, D.; Bousquet, O.; Lal, T.; Weston, J.; Schölkopf, B. Learning with local and global consistency. *NeurIPS* **2003**, *16*, 321–328.

23. Zbontar, J.; Jing, L.; Misra, I.; LeCun, Y.; Deny, S. Barlow twins: Self-supervised learning via redundancy reduction. In Proceedings of the International Conference on Machine Learning, Virtual Event, 18–24 July 2021; pp. 12310–12320.
24. Li, S.; Han, L.; Wang, Y.; Pu, Y.; Zhu, J.; Li, J. GCL: Contrastive learning instead of graph convolution for node classification. *Neurocomputing* **2023**, *551*, 126491. [[CrossRef](#)]
25. Cundy, C.; Ermon, S. Sequencematch: Imitation learning for autoregressive sequence modelling with backtracking. *Adv. Neural Inf. Process. Syst.* **2023**. [[CrossRef](#)]
26. Krizhevsky, A.; Hinton, G. *Learning Multiple Layers of Features from Tiny Images*; Technical Report, University of Toronto, Toronto, ON, Canada, 2009.
27. Netzer, Y.; Wang, T.; Coates, A.; Bissacco, A.; Wu, B.; Ng, A.Y. Reading digits in natural images with unsupervised feature learning. In Proceedings of the NIPS Workshop on Deep Learning and Unsupervised Feature Learning, Granada, Spain, 16 December 2011; Volume 2011, p. 7.
28. Coates, A.; Ng, A.; Lee, H. An analysis of single-layer networks in unsupervised feature learning. In Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, Fort Lauderdale, FL, USA, 11–13 April 2011; pp. 215–223.
29. Zagoruyko, S.; Komodakis, N. Wide residual networks. In Proceedings of the British Machine Vision Conference 2016, York, UK, 19–22 September 2016; pp. 87.1–87.12.
30. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
31. Zhou, B.; Lu, J.; Liu, K.; Xu, Y.; Cheng, Z.; Niu, Y. HyperMatch: Noise-tolerant semi-supervised learning via relaxed contrastive constraint. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 24017–24026.
32. Sutskever, I.; Martens, J.; Dahl, G.; Hinton, G. On the importance of initialization and momentum in deep learning. In Proceedings of the International Conference on Machine Learning, Atlanta, GA, USA, 16–21 June 2013; pp. 1139–1147.
33. Loshchilov, I.; Hutter, F. Sgdr: Stochastic gradient descent with warm restarts. In Proceedings of the International Conference on Learning Representations, Toulon, France, 24–26 April 2017; pp. 1–16.
34. Kalantidis, Y.; Sariyildiz, M.B.; Pion, N.; Weinzaepfel, P.; Larlus, D. Hard negative mixing for contrastive learning. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 21798–21809.
35. Wang, Y.; Chen, H.; Fan, Y.; Sun, W.; Tao, R.; Hou, W.; Wang, R.; Yang, L.; Zhou, Z.; Guo, L.Z.; et al. Usb: A unified semi-supervised learning benchmark for classification. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 3938–3961.
36. Berthelot, D.; Carlini, N.; Goodfellow, I.; Papernot, N.; Oliver, A.; Raffel, C.A. Mixmatch: A holistic approach to semi-supervised learning. *Adv. Neural Inf. Process. Syst.* **2019**, *32*, 1–14.
37. Rasmus, A.; Berglund, M.; Honkala, M.; Valpola, H.; Raiko, T. Semi-supervised learning with ladder networks. *Adv. Neural Inf. Process. Syst.* **2015**, *28*, 3546–3554.
38. Miyato, T.; Maeda, S.I.; Koyama, M.; Ishii, S. Virtual adversarial training: A regularization method for supervised and semi-supervised learning. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *41*, 1979–1993. [[CrossRef](#)] [[PubMed](#)]
39. Tarvainen, A.; Valpola, H. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 1195–1204.
40. Pham, H.; Dai, Z.; Xie, Q.; Le, Q.V. Meta pseudo labels. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 11557–11568.
41. Zhang, Y.; Yang, J.; Tan, Z.; Yuan, Y. Relationmatch: Matching in-batch relationships for semi-supervised learning. *arXiv* **2023**, arXiv:2305.10397.
42. Su, J.C.; Maji, S. The semi-supervised inaturalist challenge at the fgvc8 workshop. *arXiv* **2021**, arXiv:2106.01364.

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.