Towards Cost-Effective Learning: A Synergy of Semi-Supervised and Active Learning

Tianxiang Yin[1,2,3] , Ningzhong Liu[1,3*], Han Sun[1,3]

[1] College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics
[2] School of Computer Science/School of Software, Luoyang Institute of Science and Technology
[3] MIIT Key Laboratory of Pattern Analysis and Machine Intelligence

{yintx,sunhan}@nuaa.edu.cn, lnz_nuaa@163.com

Abstract

Active learning (AL) and semi-supervised learning (SSL) both aim to reduce annotation costs: AL selectively annotates high-value samples from the unlabeled data, while SSL leverages abundant unlabeled data to improve model performance. Although these two appear intuitively compatible, directly combining them remains challenging due to fundamental differences in their frameworks. Current semi-supervised active learning (SSAL) methods often lack theoretical foundations and often design AL strategies tailored to a specific SSL algorithm rather than genuinely integrating the two fields. In this paper, we incorporate AL objectives into the overall risk formulation within the mainstream pseudo-label-based SSL framework, clarifying key differences between SSAL and traditional AL scenarios. To bridge these gaps, we propose a feature re-alignment module that aligns the features of unlabeled data under different augmentations by leveraging clustering and consistency constraints. Experimental results demonstrate that our module enables flexible combinations of SOTA methods from both AL and SSL, yielding more efficient algorithm performance.

1. Introduction

In recent years, deep learning has made remarkable progress in various domains. However, these advances depend heavily on a vast amount of high-quality labeled data. Despite the decreasing cost of acquiring unlabeled data, the annotation of these data still requires substantial human labor costs. Therefore, in practical applications, the high cost of data labeling remains a pressing issue that needs to be addressed. Active learning(AL)[1] and semi-supervised learning(SSL)[2] can effectively tackle this problem. AL selects valuable samples for labeling while ignoring those with little or no contribution, maximizing the informational value of labeled samples. On the other hand, SSL directly leverages a large amount of unlabeled data to explore the contained information. Given their complementary strengths, a natural strategy is to combine AL and SSL to achieve more cost-effective learning. One promising approach is to integrate the SSL process within an iterative AL framework: Initially, a small set of labeled samples and a large pool of unlabeled data are used to train the model using an SSL algorithm. After training is complete, an AL algorithm selects the most informative samples from the unlabeled dataset for labeling. These newly labeled samples are then added to the labeled set, and SSL is reapplied, thereby refining the model iteratively.

Previous studies have explored the performance of combining AL with SSL and found that simple integration approaches have not achieved the expected performance gains[3]. In recent years, some studies have tried to combine AL and SSL [4–6], but these efforts often focus on developing specific AL methods tailored to SSL algorithms, rather than genuinely integrating state-of-the-art techniques from both fields. In this paper, we propose a novel approach for flexibly and effectively merging AL and SSL.

In this work, we first analyze the overall risk loss in the current mainstream pseudo-label-based SSL framework. We then introduce AL objectives into this framework, providing a theoretical examination of the key differences in sample selection strategies between SSAL and AL. Through this analysis, we find that the upper bound of the risk loss in SSAL is influenced by two types of errors: *pseudo error* and *margin error*. These errors directly correspond to the two primary strategies in AL sample selection: diversity-based and uncertainty-based strategies.

However, directly applying these two types of selection strategies to SSAL presents two issues. The first challenge arises with diversity-based methods, which rely heavily on the clustering assumption [7], assuming that samples within

*Corresponding author is Ningzhong Liu

the same class exhibit high similarity. This assumption is valid only when the model has learned the features of each class adequately, enabling it to cluster similar samples. However, with limited labeled data, each class may have only a few labeled examples, making it difficult for the model to learn distinctive features. As a result, the clustering effect tends to be weak, leading to suboptimal performance when applying diversity-based selection in SSAL. The second issue pertains to the uncertainty-based strategy, which is ideally applied to strong augmented views of the samples. However, feature differences between the strong augmented views of the same sample can sometimes be substantial[8], complicating the accurate assessment of sample uncertainty.

To address the first issue of inadequate clustering, we draw on unsupervised learning insights to explicitly cluster unlabeled data[9]. During SSL training, we maintain two feature queues for strong and weak augmentations of all labeled data. Using the strong augmented features, we identify labeled data that are similar to the unlabeled data and use a similarity loss function to align their weak augmented features. To tackle the second issue of significant feature variations between strong augmented views, we use weak augmented features as proxies for strong augmented features in the uncertainty estimation. By enforcing constraints between the strong and weak augmented features, we ensure that weak augmented features can effectively approximate strong augmented features for more accurate uncertainty assessment. To conclude, our contributions are:

1. We provide a comprehensive theoretical analysis of risk loss in the mainstream SSL framework and integrate AL objectives. Our analysis identifies two key error terms for the upper bound of SSAL risk, which correspond to the two primary objectives in active learning.
2. Building upon our theoretical findings, we propose a feature re-alignment module that bridges the gap between AL and SSAL. This module can be seamlessly integrated into existing SSL frameworks that use strong-weak augmentations and can be effectively combined with various AL algorithms.
3. In the experimental section, we demonstrate the practical effectiveness of the proposed module by combining it with two representative SSL algorithms (FreeMatch and FixMatch) and multiple AL algorithms. Our extensive experiments validate that the module facilitates the efficient integration of SSL and AL methods, significantly improving performance.

2. Related Works

2.1. Active Learning

Currently, AL methods are generally classified into three categories: uncertainty-based, diversity-based, and hybrid methods. Uncertainty-based AL algorithms focus on selecting samples that are closest to the decision boundary and are the most uncertain. A number of these methods adopt Bayesian theory are grounded in the Expected Loss Reduction theory[10–14], which aims to reduce the expected risk loss. Several approaches measure uncertainty by leveraging information generated by the model, such as predicted probabilities[15, 16], predicted loss values[17, 18], generated gradients[19], network changes during training[20, 21], etc. Furthermore, some methods incorporate adversarial training[22–25] or utilize multiple classifiers[26, 27]. Diversity-based methods focus on selecting the most representative samples that are significantly different from the existing labeled data. CoreSet[28] is a key method in this category, with several improved versions[29–31]. Hybrid methods combine both uncertainty and diversity criteria to balance the benefits of both strategies. Methods like BADGE[32], BAIT[33], and AlphaMix[34] first select samples based on uncertainty and then apply diversity-based filtering to refine the set.

2.2. Semi-supervised Learning

Currently SSL methods mainly focus on the self-training paradigm, which generates pseudo-labels by leveraging the model's own high-confidence predictions. Pseudo-label[35] is a foundational framework in SSL due to its simplicity and efficiency. Following this, UDA[36] find that strong augmentation strategies can improve model performance efficiently, while FixMatch[37] unified the SSL paradigm by leveraging the information discrepancy between strongly and weakly augmented views to extract feature information from unlabeled samples. Subsequently, methods like FlexMatch[38], FreeMatch[39], AdaMatch[40], Dash[41], CoMatch[42], and SoftMatch[43] have refined pseudo-label quality by incorporating distribution alignment, dynamic thresholds, and other strategies. In addition, DeFixMatch[44] explore the theoretical underpinnings of unbiased training in FixMatch, while DebiasPL[45] and DST[46] investigate the class imbalance issues inherent in pseudo-labeling.

2.3. Semi-supervised Active Learning

SSAL merges the strengths of AL and SSL, aiming to improve model performance under conditions of limited annotation. However, previous attempts to combine AL and SSL have not consistently yielded the desired performance improvements[3]. This is primarily because simple combinations often fail to address the unique challenges posed by each method. As a result, contemporary research has focused more on tailoring specific AL strategies for SSL. For example, Consit[4] designs a consistency-based sample selection criterion and applies it to the Mixmatch SSL algorithm. Similarly, IDEAL[47] proposes an AL method

for SSL based on consistency principles. Although other studies[5, 6, 48] have developed corresponding AL methods for SSL, these approaches often fail to fully integrate the two paradigms. The ideal solution should seamlessly combine the advanced techniques from both AL and SSL to optimize performance.

In this paper, we aim to tackle this problem by proposing a novel solution based on theoretical analysis that bridges the gap between AL and SSAL, ultimately enhancing the efficiency and effectiveness of their integration.

3. Method

3.1. Preliminaries of SSL

In SSL framework, we have a labeled dataset $L = \{x_i^l, y_i^l\}_{i=1}^{N_l}$ and an unlabeled dataset $U = \{x_j^u\}_{j=1}^{N_u}$, where N_l and N_U represent the number of samples in the labeled and unlabeled datasets, respectively, and $N_l << N_u$. The supervised loss for labeled data is:

$$\mathcal{L}_l = \frac{1}{N_l} \sum_{i=1}^{N_l} \ell(p(y|\alpha(x_i^l)), y_i^l), \tag{1}$$

where ℓ represents the loss function, typically the cross-entropy loss; $p(.)$ denotes the model's output probability; and $\alpha(.)$ signifies weak data augmentation. For unlabeled data, current mainstream SSL methods employ a strong-weak data augmentation strategy. Specifically, for an unlabeled sample x_j^u, both a weak augmentation $\alpha(x_j^u)$ and a strong augmentation $\mathcal{A}(x_j^u)$ are applied. The pseudo-labels are generated based on the weakly augmented view and used with the strongly augmented view for training. The loss function for unlabeled data is formulated as follows:

$$\mathcal{L}_u = \frac{1}{N_u} \sum_{j=1}^{N_u} \mathbb{I}(\max(q(x_j^u)) \geq \tau) \ell(Q(x_j^u), \hat{q}(x_j^u)), \tag{2}$$

where $q(\cdot)$ and $Q(\cdot)$ are abbreviations for $p(y|\alpha(x_j^u))$ and $p(y|\mathcal{A}(x_j^u))$, respectively. Here, $\hat{q}(x_j^u)$ denotes the pseudo-label derived from $q(x_j^u)$, and τ is a filtering threshold.

Various SSL methods refine this basic formula, for instance, by employing a dynamic threshold $\tau_{dynamic}$, correcting prediction probabilities in the weak view, applying diverse strong augmentations, etc. However, the core concept remains rooted in Equation 2, leveraging the strong-weak augmentation strategy to compute the loss for unlabeled data.

3.2. Problem Formulation

In this work, we consider a multi-class classification problem on an input space $\mathcal{X}$ and a label space $\mathcal{Y} = \{1, 2, ..., K\}$ containing K classes. Both the labeled dataset L and the unlabeled dataset U are sampled from the same distribution $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$. Specifically, the labeled dataset L is represented as $L = \{x_i^l, y_i^l\}_{i=1}^{N_l}$, and the unlabeled dataset U is represented as $U = \{x_j^u, \hat{y}_j^u\}_{j=1}^{N_u}$, where $\hat{y}_j^u$ is the pseudo-label generated by the weak augmented view of x_j^u. These pseudo-labels may be correct or incorrect. We denote Θ_{al} and Θ_{ssl} as the AL and SSL algorithms, respectively. The goal of SSAL is to identify a subset $S \subset U$ through Θ_{al} for annotation, subsequently applying Θ_{ssl} to leverage the updated labeled and unlabeled datasets to maximize model performance. However, the inherent differences between Θ_{al} and Θ_{ssl} often lead to suboptimal integration outcomes. This article aims to bridge the gap between AL and SSAL, enhancing the effectiveness of combining Θ_{al} and Θ_{ssl}. In the following sections, we provide a theoretical analysis and a comprehensive description of our proposed method.

3.3. Theoretical Analysis

To clearly represent the true label situation of unlabeled samples, we define an idea $U^* = \{x_j^u, y_j^u\}_{j=1}^{N_u}$, where y_j^u represents the true label of the unlabeled sample x_j^u. We also define the dataset D, which combines all labeled and unlabeled samples, defined as $D = L \cup U = \{x_i^l, y_i^l\}_{i=1}^{N_l} \cup \{x_j^u, \hat{y}_j^u\}_{j=1}^{N_u}$. Additionally, we define an ideal dataset D^*, which contains all labeled samples and the true labels, represented by $D^* = L \cup U^* = \{x_i^l, y_i^l\}_{i=1}^{N_l} \cup \{x_j^u, y_j^u\}_{j=1}^{N_u}$. During the training process of the SSL algorithm, we train the model on the dataset D, and the resulting model is denoted as f_D. This model induces a labeling function h_D, which is defined as:

$$h_D = \arg\max_{k \in \mathcal{Y}} f_D^{(k)}(x), \tag{3}$$

where $f_D^{(k)}$ represents the k-th output value of the network.

We start by analyzing the risk of the SSL framework. On the overall distribution $\mathcal{Z}$, the model's risk can be decomposed into three parts: Generation Error, Training Error, and Semi-supervised Error. This decomposition can be expressed as follows[49]:

$$\begin{aligned}
&\mathbb{E}_{x,y\sim p_{\mathcal{Z}}} \ell(x, y; f_D) \\
\leq &\underbrace{|\, \mathbb{E}_{x,y\sim p_{\mathcal{Z}}} \ell(x, y; f_D) - \frac{1}{|D^*|} \sum_{(x,y)\in D^*} \ell(x, y; f_D) \,|}_{Generalization\ Error} \\
+ &\underbrace{|\, \frac{1}{|D|} \sum_{(x,y)\in D} \ell(x, y; f_D) \,|}_{Training\ Error} \\
+ &\underbrace{|\, \frac{1}{|D^*|} \sum_{(x,y)\in D^*} \ell(x, y; f_D) - \frac{1}{|D|} \sum_{(x,y)\in D} \ell(x, y; f_D) \,|}_{Semi-supervised\ Error}.
\end{aligned} \tag{4}$$

Previous research [50] has theoretically proved that there is an upper bound for the Generation Error of deep networks, and during the training process, deep networks can fit the given dataset well, making the Training Error relatively small. Therefore, this study focuses on Semi-supervised error in formula 4, which is mainly caused by the presence of incorrect pseudo-labels in unlabeled data.

Assuming that the loss values for labeled data L equal 0 after sufficient training, the Semi-supervised Error can be simplified as:

$$\sum_{(x,y)\in U^*} \ell(x, y; f_D) - \sum_{(x,\hat{y})\in U} \ell(x, \hat{y}; f_D). \tag{5}$$

We introduce the commonly used cross-entropy loss function, and denote $\mathcal{A}(x)$ and $\alpha(x)$ as x_s and x_w. The Equation 5 can be derived as:

$$\sum_{(x,y)\in U^*} \sum_{k\in\mathcal{Y}} \mathcal{P}(x_w, k; D) \cdot \mathcal{M}(x_s, k; D). \tag{6}$$

Here $\mathcal{P}(x_w, k; D) = |\mathbb{I}(y = k) - \mathbb{I}(h_D(x_w) = k)|$ is termed the *pseudo error*, and $\mathcal{M}(x_s, k; D) = f_D^{(max)}(x_s) - f_D^{(k)}(x_s)$ is referred to as the *margin error*. The aim of AL in the SSL framework is to find a subset $S \subset U$ that minimize:

$$\sum_{(x,y)\in(U^*/S)} \sum_{k\in\mathcal{Y}} \mathcal{P}(x_w, k; D \cup S) \cdot \mathcal{M}(x_s, k; D \cup S). \tag{7}$$

To simplify, we consider selecting a single sample $x \in U$, aiming to reduce both *pseudo error* and *margin error*. *These two errors align with the diversity-based and uncertainty-based sampling strategies in AL, respectively.* We analyze this below.

For the *pseudo error*, we assume that samples with higher similar features are more likely to belong to the same class. Let z_j^u represent the feature of an unlabeled sample x_j^u, and z_*^l denote the feature of its nearest labeled sample (x_*^l, y_*^l). If we use the y_*^l as the pseudo-label for x_j^u, the overall distribution of z_j^u is $\eta(z_j^u)$, and the empirical distribution is $\eta(z_*^l)$. The expected *pseudo error* for the unlabeled sample (x_j^u, y_j^u) can then be represented as:

$$\begin{aligned}
&\mathbb{E}_{k\in\mathcal{Y}}|\mathbb{I}(y_j^u = k) - \mathbb{I}(h_D(\alpha(x_j^u)) = k)| \\
=&\sum_{k\in\mathcal{Y}} P_{y_i^u\sim\eta_k(z_j^u)}(y_i^u = k)\mathbb{I}(h_D(\alpha(x_j^u)) \neq y_i^u) \\
\overset{(a)}{\leq}&\sum_{k\in\mathcal{Y}} P_{y_i^u\sim\eta_k(z_*^l)}(y_i^u = k)\mathbb{I}(h_D(\alpha(x_j^u)) \neq y_i^u) \\
&+ \sum_{k\in\mathcal{Y}} |\eta_k(z_j^u) - \eta_k(z_*^l)|\mathbb{I}(h_D(\alpha(x_j^u)) \neq y_i^u) \\
\propto&(1 - \mathrm{sim}(z_j^u, z_*^l)),
\end{aligned} \tag{8}$$

where $\mathrm{sim}(\cdot)$ represents the cosine similarity. The notation (a) in the above equation refers to the inference presented in the study [51]. Thus, the *pseudo error* inversely correlates with feature similarity, indicating that increased similarity reduces pseudo error. The object of AL for reducing *pseudo error* is to find

$$\begin{aligned}
x^* &= \arg\min_x \sum_{(x,y)\in U^*} \sum_{k\in\mathcal{Y}} \mathcal{P}(x_w, k; D \cup \{x, y\}) \\
&= \arg\max_{x\in U} \sum_{\substack{x'\in U^* \\ x'\neq x}} \mathrm{sim}(z, z').
\end{aligned} \tag{9}$$

Equation 9 implies finding the most representative sample in the unlabeled dataset, a goal that is consistent with the diversity-based approach.

For the *margin error*, it assesses the difference in network output between the predicted category and the true label under strong augmented. When an unlabeled sample's pseudo-label equals to its true label, *pseudo error* equals 0. So we only need to consider the *margin error* when the pseudo-label is incorrect, $\mathbb{I}(h_D(x_w) \neq y)$. To minimize Equation 7, we need to find the sample with the largest *margin error* in the U and add it to S. As stated in the previous section, SSL is making the strong augmented predictions progressively closer to the pseudo-labels, so that at the end of semi-supervised training, the predictions under the strong augmentation $h_D(x_s)$ will be consistent with the pseudo-labels $h_D(x_w)$. Thus, for an unlabeled data x with true label y and pseudo-label $\hat{y} = h_D(x_w)$, we can obtain the semi-supervised error when $\hat{y} \neq y$:

$$\begin{aligned}
&\sum_{k\in\mathcal{Y}} \mathcal{P}(x_w, k; D \cup S) \cdot \mathcal{M}(x_s, k; D \cup S) \\
=&f_D^{(max)}(x_s) - f_D^{(y)}(x_s) + f_D^{(max)}(x_s) - f_D^{(\hat{y})}(x_s) \\
=&f_D^{(\hat{y})}(x_s) - f_D^{(y)}(x_s).
\end{aligned} \tag{10}$$

This formula represents the difference between the predicted category and its true label category probability. A higher value suggests that the probabilities assigned to non-true labels are relatively higher, indicating that the sample contains more information. This sample selection approach has the same purpose as the uncertainty-based approach.

According to the above analysis, the diversity-based approaches can mitigate *pseudo error*, and the uncertainty-based approach can reduces *margin error*. Although sample selection strategy under SSAL is closely linked to AL, there are two key differences between them that make the AL and SSL algorithms unsatisfactory for integration. The first difference is the number of labeled samples. Although AL is trained with a relatively small number of labeled samples, there are usually several samples per class. In contrast, SSL typically utilizes only one or two labeled samples per class. This disparity leads to ineffective clustering, which results

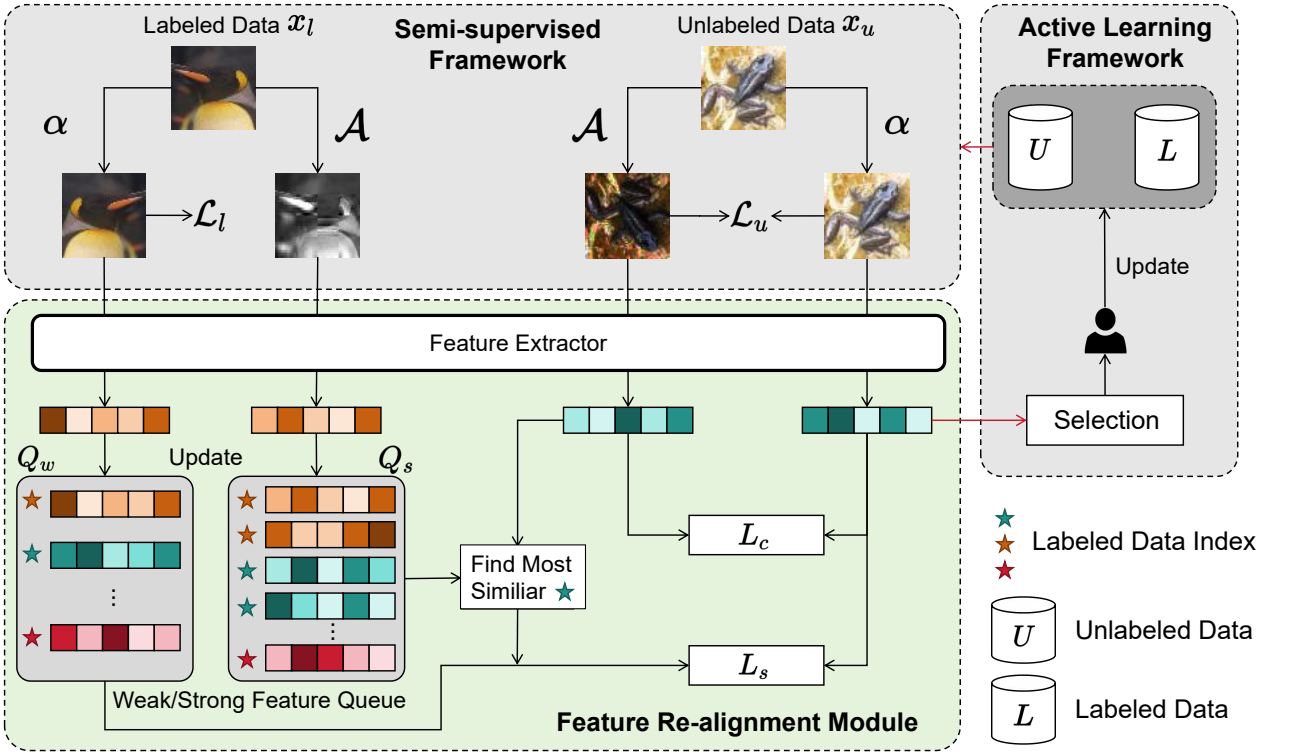


Figure 1. At the beginning of the algorithm, we use the initial labeled dataset L and unlabeled dataset U to train the model, guided by the SSL algorithm and the feature re-alignment module. In the feature realignment module, when calculating the loss $\mathcal{L}_u$ for the unlabeled data, we introduce two additional loss functions: the similarity loss function $\mathcal{L}_s$ and the contrastive loss function $\mathcal{L}_c$. After model training, samples are selected from the unlabeled dataset using various sample selection strategies under weak augmentations.

in suboptimal selection of representative samples when using the diversity-based approach. The second key difference lies in the augmentation views. AL methods typically perform sample selection under weak augmentations, while the analysis above indicates that SSAL requires uncertainty-based selection under strong augmentations. However, the variability of the same sample under different strong augmentations can be quite large. This variability makes it challenging to accurately assess sample informativeness using the uncertainty-based approach. To address these issues, we introduce the feature realignment module, which will be described in the following sections.

3.4. The Overall Framework

Figure 1 presents our proposed overall framework, which integrates the AL and SSL through a feature re-alignment module. The process is similar to the standard AL with a key distinction: instead of using a standard supervised training mode, we replace it with a SSL framework during model training. It is worth emphasizing that both the sample selection strategy within the AL framework and the learning strategy in the SSL framework are flexible, allowing the incorporation of advanced algorithms from each domain. This synergy enables the entire algorithm to leverage the best-performing techniques, resulting in improved overall performance. The next section will provide a detailed description of the feature realignment module.

3.5. Feature Re-alignment Module

As indicated by the earlier theoretical analysis, we have identified key differences in measuring the representativeness and uncertainty of samples between AL and SSAL. To address problems posed by these differences, we propose a feature realignment module that includes a similarity loss function $\mathcal{L}_s$ and a contrastive loss function $\mathcal{L}_c$.

Similarity Loss Function $\mathcal{L}_s$. Based on previous analysis, when evaluating sample representativeness, we expect the features of unlabeled data to exhibit strong clustering patterns. In SSL, labeled data is sparse, relying solely on supervised learning often fails to produce satisfactory clustering results. To address this, we introduce a similarity loss function that, under weak augmentation, explicitly performs clustering on unlabeled data. This helps improve the feature similarity between potential samples from the same class, enhancing the efficiency of sample representativeness measurement.

This method builds on insights from the unsupervised learning [9], which show that samples from the same class will exhibit high similarity under specific strong augmentations. Based on this, we identify the most similar labeled samples to each unlabeled sample under strong augmentation and compute the feature similarity loss using weak augmentation. By minimizing the similarity loss, we enhance feature similarity among samples from the same class, facilitating the reduction of *pseudo error* when selecting samples for annotation.

When calculating the labeled data loss $\mathcal{L}_l$ in the SSL framework, we create two feature queues for weak and strong augmentation views: the weak view feature queue $Q_w = \{q^l_{i,w}\}_{i=1}^{N_l}$ and the strong view feature queue $Q_s = \{q^l_{i,s,t}\}_{i=1}^{N_l}$, where $1 \leq t \leq T$ denotes the length of feature queue for each sample. At each training iteration, we replace old features in the weak view queue Q_w with newly generated weak augmented features, while for the strong view queue Q_s, we discard the earliest feature and add the latest strong augmented features. When calculating the similarity loss, we first extract the strong augmented view feature $z^u_{j,s}$ and the weak augmented view feature $z^u_{j,w}$ of unlabeled data x^u_j. Then computing the similarity between $z^u_{j,s}$ and each feature in the labeled data strong augmented view feature queue Q_s, selecting the most similar labeled sample $x^l_{i^*}$ based on:

$$i^* = \arg \max_{1 \leq i \leq N_l} \{\text{sim}(z^u_{j,s}, q^l_{i,s,t})\}_{1 \leq t \leq T}. \tag{11}$$

Next, we compute the similarity between $z^u_{j,w}$ and the corresponding weak augmented feature $q^l_{i^*,w}$ in Q_w, and use this as the similarity loss:

$$\mathcal{L}_s(x^u_j) = 1 - \text{sim}(z^u_{j,w}, q^l_{i^*,w}), \;\; i^* \text{ s.t. Equation 11.} \tag{12}$$

By introducing the similarity loss function $\mathcal{L}_s$, we can bring features of samples from the same class closer together, efficiently minimizing the *pseudo error* after sample selection based on representativeness criterion.

Contrastive Loss Function $\mathcal{L}_c$. When dealing with *margin error*, we need to evaluate it under strong augmentation views. However, variations between strong augmentation views lead to significant differences in the network outputs. To tackle this, we approximate the classification boundary of strong augmentations using weak augmentation views, by aligning the features of both views. First, we define the feature distance between strong and weak augmentation views as:

$$L_{s-w}(U; g) = \mathop{\mathbb{E}}_{x \in U} \mathop{\mathbb{E}}_{\substack{x_s \sim \mathcal{A}(x) \\ x_w \sim \alpha(x)}} ||g(x_s) - g(x_w)||^2, \tag{13}$$

where g is the feature extractor. We denote *margin error* for sample x in Equation 6 as $\Delta_m(x; g)$ and express it as:

$$\begin{aligned} \Delta_m(x; g) &= f^{\hat{y}}_D(x) - f^y_D(x) = W^T_{\hat{y}} g(x) - W^T_y g(x) \\ &= (W^T_{\hat{y}} - W^T_y) g(x). \end{aligned} \tag{14}$$

Our goal is to use $\Delta_m(x_w; g)$ from the weak augmentation view to approximate$\Delta_m(x_s; g)$ from the strong augmentation view. We define the gap Δ_{s-w} between them as follows:

$$\begin{aligned} \Delta_{s-w}(x; g) = &\mathop{\mathbb{E}}_{\substack{x_s \sim \mathcal{A}(x) \\ x_w \sim \alpha(x)}} |\Delta_m(x_w) - \Delta_m(x_s)| \\ \leq &\Gamma \cdot \mathop{\mathbb{E}}_{\substack{x_s \sim \mathcal{A}(x) \\ x_w \sim \alpha(x)}} ||g(x_s) - g(x_w)||^2, \end{aligned} \tag{15}$$

where $\Gamma = ||W^T_{\hat{y}} - W^T_y||^2$. Furthermore, we define the distance difference between strong and weak augmentation views on dataset U as follows:

$$\Delta_{s-w}(U; g) = \mathop{\mathbb{E}}_{x \in U} \Delta_{s-w}(x) \leq \Gamma \cdot L_{s-w}(g). \tag{16}$$

Now, we can conclude that $\Delta_{s-w}(U; g)$ is upper bounded by $L_{s-w}(U; g)$.

Assuming that the features generated by g have a dimension of d and are all normalized to a modulus of 1, we adopt an invariant feature approach[52], constraining the model with the following contrastive loss:

$$\mathcal{L}_c(x) = \sum_{i=1}^{d} (1 - \mathop{\mathbb{E}}_{\substack{x_s \sim \mathcal{A}(x) \\ x_w \sim \alpha(x)}} [g_i(x_s)^T g_i(x_w)])^2. \tag{17}$$

This formula represents constraining the same dimension of features generated from the same sample under strong and weak augmentation views to ensure consistency. Then, the overall loss function on dataset U is given by:

$$\mathcal{L}_c(U; g) = \sum_{i=1}^{d} (1 - \mathop{\mathbb{E}}_{x \in U} \mathop{\mathbb{E}}_{\substack{x_s \sim \mathcal{A}(x) \\ x_w \sim \alpha(x)}} [g_i(x_s)^T g_i(x_w)])^2. \tag{18}$$

Through simple derivation in Equation 19, we can understand that the upper bound of $L_{s-w}(U; g)$ is determined by this loss.

$$\begin{aligned} L_{s-w}(U; g) = &\mathop{\mathbb{E}}_{\substack{x_s \sim \mathcal{A}(x_s) \\ x_w \sim \alpha(x_w)}} ||g(x_s) - g(x_w)||^2 \\ &\leq 2\sqrt{d \cdot \mathcal{L}_c(U; g)}. \end{aligned} \tag{19}$$

By optimizing the contrastive loss $\mathcal{L}_c(x)$, we reduce the gap in *margin errors* between strong and weak augmentations, facilitating the application of uncertainty-based methods on weak augmented views.

Dataset	Methods	Random	CoreSet	AlpMix	ActFT	Noise	Avg
CIFAR100 # 100	Supervised	38.22	37.94	39.35	37.37	36.43	37.77$_{(-)}$
	FixMatch	43.83	42.73	46.84	49.61	41.92	45.27$_{(7.50\uparrow)}$
	+ FA	-	**43.12**	**47.02**	**51.58**	**44.21**	**46.48**$_{(8.71\uparrow)}$
	Supervised	38.22	37.94	39.35	37.37	36.43	37.77$_{(-)}$
	FreeMatch	52.71	56.43	51.43	53.65	52.28	53.44$_{(15.67\uparrow)}$
	+ FA	-	**59.57**	**62.51**	**64.39**	**57.25**	**60.93**$_{(23.16\uparrow)}$
CIFAR100 # 200	Supervised	53.21	52.62	57.75	63.82	53.23	56.85$_{(-)}$
	FixMatch	63.70	63.73	66.29	68.34	61.10	64.86$_{(8.01\uparrow)}$
	+ FA	-	**65.12**	**70.58**	**68.64**	**62.91**	**66.81**$_{(9.96\uparrow)}$
	Supervised	53.21	52.62	57.75	63.82	53.23	56.85$_{(-)}$
	FreeMatch	71.96	72.55	73.70	74.64	70.97	72.96$_{(16.11\uparrow)}$
	+ FA	-	**76.68**	**77.23**	**76.15**	**74.62**	**76.17**$_{(19.32\uparrow)}$
TinyImgnet # 200	Supervised	28.33	27.24	28.47	29.29	29.17	28.54$_{(-)}$
	FixMatch	39.08	37.99	38.51	39.46	38.24	38.55$_{(10.01\uparrow)}$
	+ FA	-	**39.51**	**40.82**	**39.97**	**40.04**	**40.08**$_{(11.54\uparrow)}$
	Supervised	28.33	27.24	28.47	29.29	29.17	28.54$_{(-)}$
	FreeMatch	38.03	39.06	38.36	39.56	38.02	38.75$_{(10.21\uparrow)}$
	+ FA	-	**40.45**	**40.44**	**40.64**	**41.41**	**40.73**$_{(12.19\uparrow)}$
TinyImgnet # 400	Supervised	42.56	41.96	42.51	43.01	43.03	42.63$_{(-)}$
	FixMatch	50.76	50.97	51.34	51.47	50.80	51.14$_{(8.51\uparrow)}$
	+ FA	-	**51.48**	**52.54**	**51.71**	**51.72**	**51.86**$_{(9.23\uparrow)}$
	Supervised	42.56	41.96	42.51	43.01	43.03	42.63$_{(-)}$
	FreeMatch	51.55	51.34	52.07	52.13	52.70	52.06$_{(9.43\uparrow)}$
	+ FA	-	**51.92**	**52.98**	**53.07**	**53.22**	**52.79**$_{(10.16\uparrow)}$

Table 1. Experimental results on CIFAR100 and TinyImagenet. Accuracy(%) is used as a metric to measure model performance. All results are averages of 3 trials. The arrows in the "Avg" column indicate the comparison with the "Supervised" performance of the same group.

4. Experiments

4.1. Experiment Settings

We conducted experiments on two commonly used datasets, CIFAR-100 and TinyImagenet, which contains 100 and 200 categories. We randomly select one or two samples from each category as the initial labeled dataset and continue to select the same number of samples in each subsequent round. Specifically, #100 in Table 1 represents the initial dataset size of 100 samples. In all experiments conducted in this study, we only performed one selection iteration. For SSL algorithms, we selected the classic FixMatch [37] and the latest state-of-the-art method FreeMatch [39]. For active learning algorithms, we employed four methods: CoreSet [28], AlphaMix [34], ActiveFT [53] and Noise[54]. Additionally, we used ViT [55] as the classification model and utilized the pre-trained weights provided by the framework.

4.2. Results Analysis

The overall experimental results are shown in Table 1. We conducted a total of 32 sets of comparative experiments, each combining one AL method with three training strategies: Supervised, SSL, and SSL + FA. Among them, "Supervised" represents the standard supervised learning using only labeled data; the second row (FixMatch and FreeMatch) displays the results using the SSL training strategy; and the third row "+ FA" indicates our proposed Feature Alignment module on top of SSL. The "Avg" values in the last column represent the mean of the four AL methods in each row, reflecting the average performance when combined with AL under different SSL settings.The results show that after introducing our proposed module, the final performance is enhanced in all cases.

Notably, directly applying AL in SSL sometimes underperforming than random selection. For instance, under the CIFAR100 #100 setting when using the FreeMatch SSL algorithm, the performance of AL sample selection algorithms such as AlpMix(51.43%) and Noise(52.28%) is inferior to Random(52.71%). However, after adding our proposed FA module, the performance shows significant improvement compared to Random. This further validates the effectiveness of our Feature Alignment module.

4.3. Ablation Study

The feature alignment module proposed in this paper includes two key loss functions, namely the contrastive loss

$\mathcal{L}_c$ and the similarity loss $\mathcal{L}_s$. To verify the effectiveness of this module, we conducted ablation experiments using the FreeMatch SSL algorithm under the CIFAR100 #100 setting. The experimental results are shown in Table 2. Both loss functions contribute to performance improvement, and the experimental results further demonstrate the effectiveness of these two loss functions.

Method	CoreSet	AlpMix	ActFT	Noise
FreeMatch	56.43	51.43	53.65	52.28
+ $\mathcal{L}_s$	58.72	61.02	63.28	56.81
+ $\mathcal{L}_s$ + $\mathcal{L}_c$	**59.57**	**62.51**	**64.39**	**57.25**

Table 2. Ablation study on CIFAR100 #100.

4.4. Hyperparameter Settings

In our method, there is a hyperparameter T that represents the length of the strong augmentation view queue. A larger value of T means that the queue contains more strong augmentation views. According to some research results in self-supervised learning, a greater number of strong augmentations generally leads to a higher feature similarity between two different samples within the same category[9]. Therefore, the value of T has a significant impact on the performance of the algorithm.

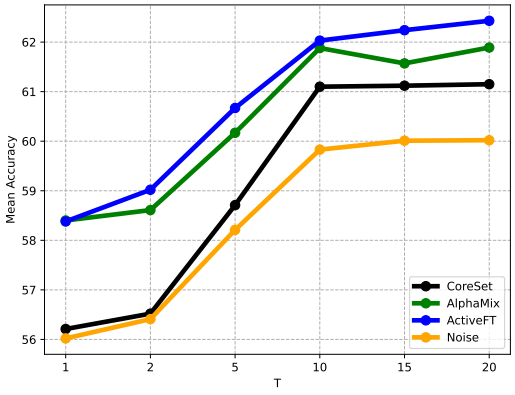
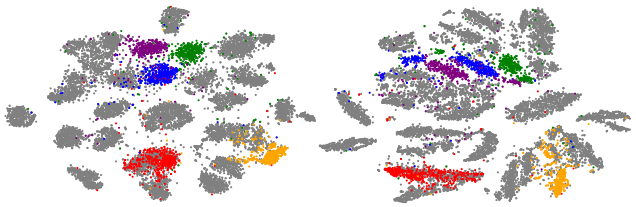


Figure 2. Accuracy(%) on CIFAR100 #100 with different T.

Figure 2 presents the algorithm results under different values of T for various active learning algorithms when using the FreeMatch SSL algorithm in the CIFAR100 #100 setting. We can observe that the performance of the algorithm improves as the value of T increases. When T is 1, only a single strongly augmented data is used for clustering unlabeled data. Due to the significant variation in sample features under strong augmentation views, the clustering effect is often poor when T is small. However, as T increases, the number of different strong augmentation views of samples saved in the queue gradually increases, leading to an improvement in algorithm performance. But when T is greater than 10, the performance improvement is not significant, and instead, more GPU memory is required. Therefore, we select the optimal parameter for T to be 10.

4.5. Effect of Similarity Loss Function



(a) (b)

Figure 3. Visualization of feature distribution for unlabeled data. The same color indicates samples with the same category. (a) shows the feature distribution when the similarity loss function is used. (b) depicts the feature distribution without the similarity loss function.

In this section, we conduct experimental validation of the similarity loss function $\mathcal{L}_s$. We train our model using the FreeMatch algorithm under the CIFAR100 #100 setting and utilize t-SNE to visualize the feature distribution of the first 30 categories of unlabeled data. To facilitate intuitive interpretation, we randomly select 5 of these categories and annotate them with distinct colors. The results demonstrate that the introduction of the similarity loss function leads to a more concentrated feature distribution and significantly improved clustering performance, which aligns well with our motivation.

5. Conclusion

In this paper, we formulate an SSAL framework which combines AL and SSL in a more flexible and efficient manner. We conduct a thorough analysis of the overall risk loss in SSL and integrate the objectives of AL into it. We elucidate the connection between the two AL paradigms and SSAL, while highlighting the issues arising from their differences. To address these issues, we propose a feature realignment module that effectively bridges the gap between SSAL and AL . Extensive experimental results demonstrate that our proposed method significantly enhances performance in combining AL with SSL.

Acknowledgments. This work was supported by the Collaborative Innovation Center of Novel Software Technology and Industrialization, the Natural Science Foundation of Jiangsu Province of China (BK20222012), the Special Project of Key Research and Development Plan of Henan Province (251111211800).

References

[1] Burr Settles. Active learning literature survey. Technical report, University of Wisconsin–Madison, 2009.

[2] Olivier Chapelle, Bernhard Schölkopf, and Alexander Zien, editors. *Semi-Supervised Learning*. The MIT Press, 2006.

[3] Sudhanshu Mittal, Maxim Tatarchenko, Özgün Çiçek, and Thomas Brox. Parting with illusions about deep active learning. *CoRR*, abs/1912.05361, 2019.

[4] Mingfei Gao, Zizhao Zhang, Guo Yu, Sercan Ömer Arik, Larry S. Davis, and Tomas Pfister. Consistency-based semi-supervised active learning: Towards minimizing labeling cost. In *European Conference on Computer Vision*, volume 12355, pages 510–526, 2020.

[5] Siyu Huang, Tianyang Wang, Haoyi Xiong, Jun Huan, and Dejing Dou. Semi-supervised active learning with temporal output discrepancy. In *IEEE/CVF International Conference on Computer Vision*, pages 3427–3436, 2021.

[6] Yanchao Li, Yongli Wang, Dong-Jun Yu, Ning Ye, Peng Hu, and Ruxin Zhao. ASCENT: active supervision for semi-supervised learning. *IEEE Trans. Knowl. Data Eng.*, 32(5):868–882, 2020.

[7] Alaa Tharwat and Wolfram Schenck. A survey on active learning: State-of-the-art, practical challenges and research directions. *Mathematics*, 11(4), 2023.

[8] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 1597–1607, 2020.

[9] Tongzhou Wang and Phillip Isola. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *International Conference on Machine Learning*, volume 119, pages 9929–9939, 2020.

[10] Nicholas Roy and Andrew McCallum. Toward optimal active learning through sampling estimation of error reduction. In *International Conference on Machine Learning*, pages 441–448, 2001.

[11] Neil Houlsby, Ferenc Huszar, Zoubin Ghahramani, and Máté Lengyel. Bayesian active learning for classification and preference learning. *CoRR*, abs/1112.5745, 2011.

[12] Yarin Gal, Riashat Islam, and Zoubin Ghahramani. Deep bayesian active learning with image data. In Doina Precup and Yee Whye Teh, editors, *International Conference on Machine Learning*, volume 70, pages 1183–1192, 2017.

[13] Guang Zhao, Edward R. Dougherty, Byung-Jun Yoon, Francis J. Alexander, and Xiaoning Qian. Uncertainty-aware active learning for optimal bayesian classifier. In *International Conference on Learning Representations*, 2021.

[14] Wei Tan, Lan Du, and Wray L. Buntine. Bayesian estimate of mean proper scores for diversity-enhanced active learning. *IEEE Trans. Pattern Anal. Mach. Intell.*, 46(5):3463–3479, 2024.

[15] Tobias Scheffer, Christian Decomain, and Stefan Wrobel. Active hidden markov models for information extraction. In *International Conference on Advances in Intelligent Data Analysis*, volume 2189, pages 309–318, 2001.

[16] Claude E. Shannon. A mathematical theory of communication. *Bell Syst. Tech. J.*, 27(3):379–423, 1948.

[17] Donggeun Yoo and In So Kweon. Learning loss for active learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 93–102, 2019.

[18] Kwanyoung Kim, Dongwon Park, Kwang In Kim, and Se Young Chun. Task-aware variational adversarial active learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 8166–8175. Computer Vision Foundation / IEEE, 2021.

[19] Tianyang Wang, Xingjian Li, Pengkun Yang, Guosheng Hu, Xiangrui Zeng, Siyu Huang, Cheng-Zhong Xu, and Min Xu. Boosting active learning via improving test performance. In *AAAI Conference on Artificial Intelligence*, pages 8566–8574, 2022.

[20] Seong Min Kye, Kwanghee Choi, Hyeongmin Byun, and Buru Chang. Tidal: Learning training dynamics for active learning. In *IEEE/CVF International Conference on Computer Vision*, pages 22278–22288, 2023.

[21] Yincheng Han, Dajiang Liu, Jiaxing Shang, Linjiang Zheng, Jiang Zhong, Weiwei Cao, Hong Sun, and Wu Xie. BALQUE: batch active learning by querying unstable examples with calibrated confidence. *Pattern Recognit.*, 151:110385, 2024.

[22] Samarth Sinha, Sayna Ebrahimi, and Trevor Darrell. Variational adversarial active learning. In *IEEE/CVF International Conference on Computer Vision*, pages 5971–5980. IEEE, 2019.

[23] Ali Mottaghi and Serena Yeung. Adversarial representation active learning. *CoRR*, abs/1912.09720, 2019.

[24] Beichen Zhang, Liang Li, Shijie Yang, Shuhui Wang, Zheng-Jun Zha, and Qingming Huang. State-relabeling adversarial active learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8753–8762, 2020.

[25] Heechul Lim, Kang-Wook Chon, and Min-Soo Kim. Active learning using generative adversarial networks for improving generalization and avoiding distractor points. *Expert Syst. Appl.*, 227:120193, 2023.

[26] Jae-Won Cho, Dong-Jin Kim, Yunjae Jung, and In So Kweon. MCDAL: maximum classifier discrepancy for active learning. *IEEE Trans. Neural Networks Learn. Syst.*, 34(11):8753–8763, 2023.

[27] Lin Geng, Ningzhong Liu, and Jie Qin. Multi-classifier adversarial optimization for active learning. In *AAAI Conference on Artificial Intelligence*, pages 7687–7695. AAAI Press, 2023.

[28] Ozan Sener and Silvio Savarese. Active learning for convolutional neural networks: A core-set approach. In *International Conference on Learning Representations*, 2018.

[29] Rafid Mahmood, Sanja Fidler, and Marc T. Law. Low-budget active learning via wasserstein distance: An integer programming approach. In *International Conference on Learning Representations*, 2022.

[30] Ofer Yehuda, Avihu Dekel, Guy Hacohen, and Daphna Weinshall. Active learning through a covering lens. In *Advances in Neural Information Processing Systems*, 2022.

[31] Yeachan Kim and Bonggun Shin. In defense of core-set: A density-aware core-set selection for active learning. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 804–812, 2022.

[32] Jordan T. Ash, Chicheng Zhang, Akshay Krishnamurthy, John Langford, and Alekh Agarwal. Deep batch active learning by diverse, uncertain gradient lower bounds. In *International Conference on Learning Representations*, 2020.

[33] Jordan T. Ash, Surbhi Goel, Akshay Krishnamurthy, and Sham M. Kakade. Gone fishing: Neural active learning with fisher embeddings. In *Advances in Neural Information Processing Systems*, pages 8927–8939, 2021.

[34] Amin Parvaneh, Ehsan Abbasnejad, Damien Teney, Reza Haffari, Anton van den Hengel, and Javen Qinfeng Shi. Active learning by feature mixing. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12227–12236, 2022.

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

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

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

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

[39] Yidong Wang, Hao Chen, Qiang Heng, Wenxin Hou, Yue Fan, Zhen Wu, Jindong Wang, Marios Savvides, Takahiro Shinozaki, Bhiksha Raj, Bernt Schiele, and Xing Xie. Freematch: Self-adaptive thresholding for semi-supervised learning. In *International Conference on Learning Representations*, 2023.

[40] David Berthelot, Rebecca Roelofs, Kihyuk Sohn, Nicholas Carlini, and Alexey Kurakin. Adamatch: A unified approach to semi-supervised learning and domain adaptation. In *International Conference on Learning Representations*, 2022.

[41] Yi Xu, Lei Shang, Jinxing Ye, Qi Qian, Yu-Feng Li, Baigui Sun, Hao Li, and Rong Jin. Dash: Semi-supervised learning with dynamic thresholding. In *International Conference on Machine Learning*, volume 139, pages 11525–11536, 2021.

[42] Junnan Li, Caiming Xiong, and Steven C. H. Hoi. Comatch: Semi-supervised learning with contrastive graph regularization. In *IEEE/CVF International Conference on Computer Vision*, pages 9455–9464, 2021.

[43] Hao Chen, Ran Tao, Yue Fan, Yidong Wang, Jindong Wang, Bernt Schiele, Xing Xie, Bhiksha Raj, and Marios Savvides. Softmatch: Addressing the quantity-quality tradeoff in semi-supervised learning. In *International Conference on Learning Representations*, 2023.

[44] Hugo Schmutz, Olivier Humbert, and Pierre-Alexandre Mattei. Don't fear the unlabelled: safe deep semi-supervised learning via simple debiasing. *CoRR*, abs/2203.07512, 2022.

[45] Xudong Wang, Zhirong Wu, Long Lian, and Stella X. Yu. Debiased learning from naturally imbalanced pseudo-labels. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14627–14637, 2022.

[46] Baixu Chen, Junguang Jiang, Ximei Wang, Pengfei Wan, Jianmin Wang, and Mingsheng Long. Debiased self-training for semi-supervised learning. In *Advances in Neural Information Processing Systems*, 2022.

[47] Jiannan Guo, Yangyang Kang, Xiaolin Li, Wenqiao Zhang, Kun Kuang, Changlong Sun, Siliang Tang, and Fei Wu. Unleash the power of inconsistency-based semi-supervised active learning by dynamic programming of curriculum learning. *IEEE Trans. Knowl. Data Eng.*, 36(11):7268–7282, 2024.

[48] Mingjian Xie, Yiqun Geng, Weifeng Zhang, Shan Li, Yuejiao Dong, Yongjun Wu, Hongzhong Tang, and Liangli Hong. Multi-resolution consistency semi-supervised active learning framework for histopathology image classification. *Expert Syst. Appl.*, 259:125266, 2025.

[49] Pan Du, Hui Chen, Suyun Zhao, Shuwen Chai, Hong Chen, and Cuiping Li. Contrastive active learning under class distribution mismatch. *IEEE Trans. Pattern Anal. Mach. Intell.*, 45(4):4260–4273, 2023.

[50] Huan Xu and Shie Mannor. Robustness and generalization. *Mach. Learn.*, 86(3):391–423, 2012.

[51] Christopher Berlind and Ruth Urner. Active nearest neighbors in changing environments. In *International Conference on Machine Learning*, volume 37, pages 1870–1879, 2015.

[52] Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. In Marina Meila and Tong Zhang, editors, *International Conference on Machine Learning*, volume 139, pages 12310–12320, 2021.

[53] Yichen Xie, Han Lu, Junchi Yan, Xiaokang Yang, Masayoshi Tomizuka, and Wei Zhan. Active finetuning: Exploiting annotation budget in the pretraining-finetuning paradigm. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23715–23724, 2023.

[54] Xingjian Li, Pengkun Yang, Yangcheng Gu, Xueying Zhan, Tianyang Wang, Min Xu, and Chengzhong Xu. Deep active learning with noise stability. In Michael J. Wooldridge, Jennifer G. Dy, and Sriraam Natarajan, editors, *AAAI Conference on Artificial Intelligence*, pages 13655–13663, 2024.

[55] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.