

DUALMATCH: PROMOTING SEMI-SUPERVISED LEARNING WITH HIERARCHICAL LABEL AND CONTRASTIVE LEARNING

Anonymous authors

Paper under double-blind review

ABSTRACT

The recently proposed FixMatch and FlexMatch have achieved remarkable results in the field of semi-supervised learning. But these two methods go to two extremes as FixMatch and FlexMatch use a pre-defined constant threshold for all classes and an adaptive threshold for each category, respectively. By only investigating consistency regularization, they also suffer from unstable results and indiscriminative feature representation, especially under the situation of few labeled samples. In this paper, we propose a novel DualMatch method, which can learn an adaptive threshold for all classes to perform instance-level prediction matching as well as discriminative features by graph matching based contrastive learning. We first present a memory-bank based near-global threshold learning strategy to select highly-confident samples. In the meantime, we make full use of the structured information in the hierarchical labels to learn an accurate affinity graph for contrastive learning. DualMatch achieves very stable and superior results on several commonly-used benchmarks. For example, DualMatch achieves 8.44% and 9.02% error rate reduction over FlexMatch on CIFAR-100 under WRN-28-2 with only 4 and 25 labeled samples per class, respectively.

1 INTRODUCTION

Deep learning achieves great success in the past decade based on the large-scale labeled datasets. However, it is generally hard to collect and expensive to manually annotate such kind of large dataset in practice, which limits its application. Semi-supervised learning (SSL) attracts much attention recently, since it can make full use of a few labeled and massive unlabeled data to facilitate the classification.

For the task of SSL Chapelle et al. (2006); Grandvalet & Bengio (2004); Rasmus et al. (2015); Sajjadi et al. (2016), various methods have been proposed and promising results have been achieved. Consistency regularization Zhou et al. (2020) is one of the most influential techniques in this area. For example, pseudo-ensemble Bachman et al. (2014) and temporal ensembling Laine & Aila (2017) investigate the instance-level robustness before and after perturbation. Mean teacher Tarvainen & Valpola (2017) introduces the teacher-student framework and studies the model-level consistency. SNTG Luo et al. (2018) further constructs the similarity graph over the teacher model to guide the student learning. However, the supervised signal generated by only this strategy is insufficient for more challenging tasks.

Recently, by combining pseudo-labeling and consistency between weak and strong data augmentations, FixMatch Sohn et al. (2020) achieves significant improvement. But it relies on a high fixed threshold for all classes, and only a few unlabeled samples whose prediction probability is above the threshold are selected for training, resulting in undesirable efficiency and convergence. Towards this issue, FlexMatch Zhang et al. (2021) proposes a curriculum learning Bengio et al. (2009) strategy to learn adjustable class-specific threshold, which can well improve the results and efficiency. But it still suffers from the following limitations: (1) The results of both FixMatch and FlexMatch are unstable and of large variances, which is shown in Figure 1(a), especially when there are only a small amount of labeled samples; (2) Only instance-level consistency is investigated, which neglects the inter-class relationship and may make the learned feature indiscriminative.

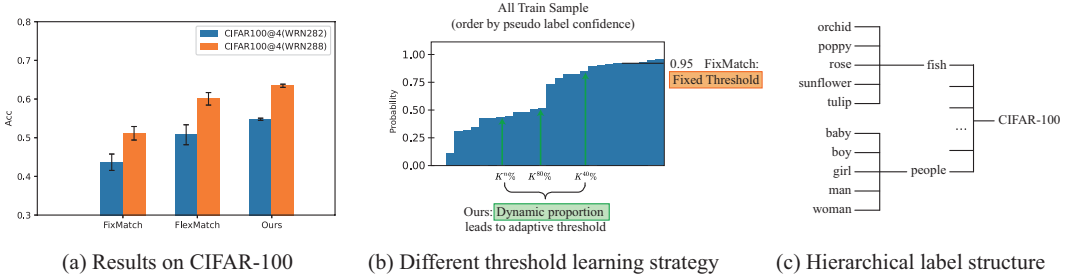


Figure 1: Motivation of our method. (a) The results of FixMatch and FlexMatch are unstable and of large variances, while our method can handle this issue. (b) FixMatch sets fixed threshold, while our method sets dynamic proportions in different epoch, leading to adaptive threshold. (c) Many datasets have hierarchical label structure, and we aim to take advantage of this to promote the SSL.

To address the above issues, we propose a novel DualMatch method based on hierarchical label and contrastive learning, which takes both the instance-level prediction matching and graph-level similarity matching into account. Specifically, we first present a memory-bank based adaptive threshold learning strategy, where we only need one parameter to compute the near-global threshold for all categories. We compare this strategy with FixMatch in Figure 1(b). Then this adaptive threshold is used for instance-level prediction matching under the similar FixMatch paradigm. More importantly, we further propose a hierarchical label guided graph matching module for contrastive feature learning, where the key lies in the construction of an accurate affinity graph. As shown in Figure 1(c), we notice that categories in many datasets as well as real-world applications have a hierarchical structure, which is neglected by existing methods for semi-supervised learning. We aim to make use of the coarse-grained labels to guide the general classification, which can also provide extra supervision signals especially under the situation of limited labeled samples. For implementation, we first add another head for the coarse-grained classification. Then each affinity graph is constructed by the fine-grained and coarse-grained classification branches. Together with the near-global threshold, we can get the precise affinity relationship after graph matching, which is then used for contrastive learning. We also conduct extensive experiments on several commonly-used datasets to verify the effectiveness of the proposed method as well as each module.

Our main contributions can be summarized as follows:

- We propose a novel DualMatch method, which contains instance-level and graph-level matching for assignment and feature learning, respectively. To the best of our knowledge, this is the *first* study that makes full use of the structured information in hierarchical labels to promote semi-supervised learning.
- We come up with a memory-bank based highly-confident sample selection strategy, which can generate better threshold for prediction-level matching, lead to more robust results, and accelerate the training process.
- Benefit from the graph matching guided by the hierarchical label, our method can construct a more accurate affinity graph for the proposed contrastive learning module, leading to more discriminative feature representation.
- We conduct extensive experiments on four benchmark datasets under different backbones, and the proposed DualMatch outperforms these state-of-the-art methods.

2 RELATED WORK

Semi-supervised learning (SSL) Assran et al. (2021); Cai et al. (2021) is a classic and important research area in the machine learning community since it only needs a few labeled samples. Along with the success of deep learning, SSL achieves significant improvement recently. We briefly review some highly-related deep learning based SSL methods in this section.

Pseudo-label Lee et al. (2013); Taherkhani et al. (2021); Hu et al. (2021); Nassar et al. (2021) and consistency regularization are two effective strategies in SSL. Pseudo-label Lee et al. (2013) picks

up the class which has the maximum predicted probability as true labels for supervised learning, which has the equivalent effect to entropy regularization. Consistency regularization assumes that under small perturbations, the results should be consistent and robust, which can be further divided into instance-level, model-level, and graph-level Iscen et al. (2019); Li et al. (2021) consistency. Temporal ensembling Laine & Aila (2017), VAT Miyato et al. (2018), UDA Xie et al. (2020), and MixMatch Berthelot et al. (2019; 2020) well investigate the influence of data augmentation and image fusion. Pseudo-ensemble Bachman et al. (2014) and mean teacher Tarvainen & Valpola (2017) study the effect of dropout and teacher-student framework, respectively. SNTG Luo et al. (2018) further constructs the similarity graph over teacher model to guide the student learning.

By combining the pseudo-label and consistency regularization, FixMatch Sohn et al. (2020) simplifies the SSL by introducing a new paradigm, where it adopts the highly-confident pseudo label of weak augmentation sample to guide the training of the corresponding sample after strong augmentation. Most lateral methods also follow this setting since it achieves state-of-the-art performance. CoMatch Li et al. (2021) further incorporates the graph based contrastive learning. FlexMatch Zhang et al. (2021) proposes a curriculum learning strategy to learn adaptive threshold for each category, which can well accelerate the convergence.

Our method also follows FixMatch, one of the main differences lies in the threshold learning. Besides, while existing methods mainly focus on consistency regularization, we incorporate the hierarchical label guided contrastive learning into the SSL to learn discriminative feature representation. Though CoMatch also combines with graph contrastive learning, it is very sensitive to the hyper-parameters, and how to use the graph information is also very different in our method. CoMatch assigns a consistency regularization on two graphs generated by weak and strong augmentations, while our method learns a more accurate graph for contrastive learning based on the matching between coarse-grained and fine-grained classification results. Compared with CoMatch, our method is more stable and achieves much better results.

3 DUALMATCH

3.1 PRELIMINARY

Given a batch of B labeled samples $\mathcal{X} = \{(x_b, y_b)\}_{b=1}^B$ and μB unlabeled samples $\mathcal{U} = \{u_b\}_{b=1}^{\mu B}$, where y denotes the one-hot label and μ is a parameter that controls the relative sizes of $\mathcal{U}$ and $\mathcal{X}$, the general deep learning based SSL method aims to learn an encoder $f(\cdot)$ and a classification head $h(\cdot)$ for discriminative feature representation and good performance. For this task, FixMatch first introduces the random strong augmentation $\mathcal{A}(\cdot)$ and weak augmentation $\alpha(\cdot)$, and then investigates their prediction-level consistency based on the pseudo-label learned by a high fixed threshold τ . Denote $p(y|x)$ as the predicted class probability and $H(p, q)$ as the cross-entropy loss between two probability distributions p and q . Then the objective function of FixMatch can be formulated as:

$$\min_{\theta_f, \theta_h} \mathcal{L}_s + \lambda \mathcal{L}_u = \frac{1}{B} \sum_{b=1}^B H(y, p(y|\alpha(x_b))) + \lambda \frac{1}{\mu B} \sum_{b=1}^{\mu B} \mathbb{1}(\max(q_b) \geq \tau) H(\hat{q}_b, p(y|\mathcal{A}(u_b))), \quad (1)$$

where θ denotes the learnable parameters, $q_b = p(y|\alpha(u_b))$ represents the predicted class probability for the weakly augmented unlabeled sample, $\hat{q}_b = \arg \max(q_b)$ is its pseudo-label, λ is a hyperparameter to balance the contribution of two items, $\mathcal{L}_s$ and $\mathcal{L}_u$ denote the losses for labeled and unlabeled samples, respectively.

Due to the fixed high threshold, only a few unlabeled data join the training with higher prediction confidence than the threshold, therefore FixMatch suffers from long training time. FlexMatch further proposes a curriculum pseudo labeling strategy to assign a specific threshold for each category.

Most existing SSL methods mainly focus on the above consistency regularization to make positive samples closer, and neglect the discriminative feature learning to push negative samples apart. Contrastive learning is a good choice to handle the above issue, which attracts much attention in the field of self-supervised learning. The basic contrastive loss for the sample u_b can be formulated as:

$$l_b^{simclr} = -\log \frac{\exp(f(\alpha(u_b)) \cdot f(\alpha(u_b))/t)}{\sum_{j=1, j \neq b}^{\mu B} \exp(f(\alpha(u_b)) \cdot f(\alpha(u_j))/t)}, \quad (2)$$

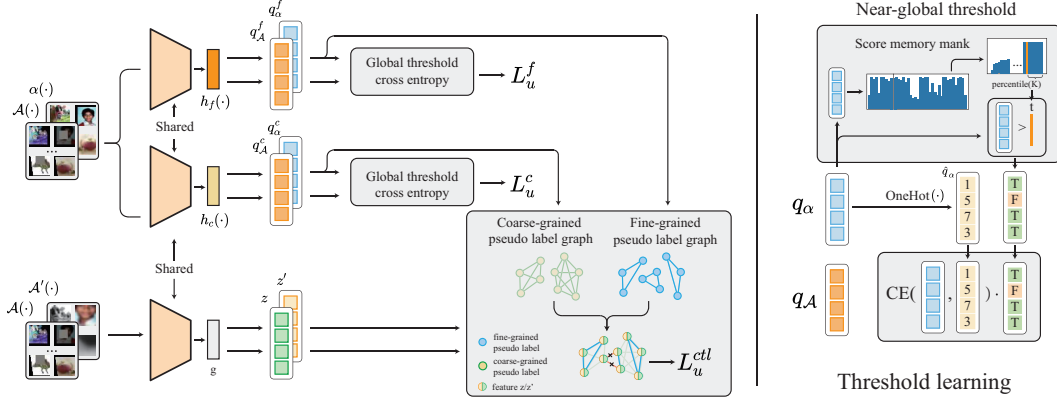


Figure 2: Framework of our DualMatch. (Left) Besides the general classification head $h_f(\cdot)$ in FixMatch, we add a coarse-grained classification head $h_c(\cdot)$ and a projection head $g(\cdot)$. We utilize the hierarchical label information, performing graph matching between fine-grained and coarse-grained pseudo label graphs to guide contrastive feature learning. (Right) A memory-back based strategy is proposed to learn adaptive near-global threshold to guide instance-level prediction matching.

where t denotes a temperature parameter. We can see that general contrastive learning treats all others in the batch as negative samples, which contains much noisy correspondence. Different from self-supervised learning, we have a few labeled samples in SSL. How to appropriately combine contrastive learning with SSL remains an open problem. And the key lies in how to learn an accurate graph. We follow the general pipeline of FixMatch, and propose strategies in the following to handle the above issues of threshold and accurate graph learning.

3.2 OVERVIEW OF DUALMATCH

The framework of our proposed DualMatch is shown in Figure 2. Based on the feature encoder $f(\cdot)$, different from general SSL methods that only have one classification head, DualMatch jointly learns the fine-grained classification head $h_f(\cdot)$, the coarse-grained classification head $h_c(\cdot)$, and the projection head $g(\cdot)$ for contrastive feature representation.

In practice, categories have a hierarchical structure, which is often neglected by existing methods. But it contains extra supervision signals for network training. So we add an extra coarse-grained classification head $h_c(\cdot)$ first. For each classification head, we learn an adaptive near-global threshold, denoted as τ_c and τ_f , to select highly-confident samples for consistent pseudo-label learning, which will be introduced in Section 3.3. Then we define the unsupervised classification losses, $\mathcal{L}_u^f$ and $\mathcal{L}_u^c$, for these two fine-grained and coarse-grained heads as follows:

$$\mathcal{L}_u^f + \mathcal{L}_u^c = \frac{1}{\mu B} \sum_{b=1}^{\mu B} \left(\mathbb{1}(\max(q_b^f) \geq \tau_f) H(\hat{q}_b^f, p(y^f | \mathcal{A}(u_b))) + \mathbb{1}(\max(q_b^c) \geq \tau_c) H(\hat{q}_b^c, p(y^c | \mathcal{A}(u_b))) \right), \quad (3)$$

where $q_b^f = h_f(f(\alpha(u_b)))$ and $\hat{q}_b^f$ denote the predicted class probability and its pseudo-label for the fine-grained classification head, respectively, which is also the similar to q_b^c and $\hat{q}_b^c$.

Based on the predicted probability of these two classification heads, we can generate an affinity graph for each branch. Then we perform graph matching to acquire an accurate graph for contrastive learning. We define the contrastive loss as $\mathcal{L}_u^{ctl}$ and impose it on the projection head $g(\cdot)$ to learn discriminative features. Details will be introduced in Section 3.4.

The overall training objective of DualMatch can be formulated as:

$$\min_{\theta_f, \theta_g, \theta_{h_c}, \theta_{h_f}} \mathcal{L} = \mathcal{L}_s + \mathcal{L}_u^f + \mathcal{L}_u^c + \mathcal{L}_u^{ctl}. \quad (4)$$

We simply fix all the weights before each loss term as 1 since our method is robust and insensitive to these parameters.

3.3 MEMORY-BANK BASED NEAR-GLOBAL THRESHOLD LEARNING

In FixMatch and related SSL methods, the threshold plays an important role in selecting highly-confident samples to construct pseudo-label for consistency learning. FixMatch manually set a fixed high threshold for all classes without learning, leading to undesirable training efficiency. FlexMatch learns class-specific thresholds based on curriculum learning. But results of these two methods are relatively unstable, especially when the labeled samples are limited.

Inspired by MoCo He et al. (2020), we propose a memory-bank based strategy to learn a near-global threshold for all classes, which can well handle the above issue of FixMatch and FlexMatch. Specifically, for each sample u_b , take the fine-grained classification head for example, we compute its maximum predicted probability by $\tilde{q}_b^f = \max q_b^f = \max h_f(f(\alpha(u_b)))$. Then we construct a memory-bank to save the maximum probability of the previous N weakly augmented samples as $Q_{MB}^f = [\tilde{q}_1^f, \tilde{q}_2^f, \dots, \tilde{q}_N^f]$, where N is much larger than the mini-batch size. At each epoch, we hope that a certain percentage $K\%$ of the samples are chosen for pseudo-label learning. For example, at the early stage of the training, $K\%$ should be small to ensure that the selected samples are of high confidence to guide the network training. In contrast, at the end of the training, $K\%$ should be large enough to guarantee that most samples can join the training. In this case, we gradually increase $K\%$ as the training progresses. There is a negative correlation between $K\%$ and τ_f , which can be formulated based on the memory-bank by:

$$|Q_{MB}^f > \tau_f|/N = K\%, \quad (5)$$

where $|Q_{MB}^f > \tau_f|$ denotes the number of samples in the memory-bank that have a larger prediction probability than τ_f . τ_f can be easily computed by the function $\tau_f = \text{percentile}(Q_{MB}^f, K)$ for each batch, which is equivalent to select the $N \times K\%$ -th value after the descend sorting. Then we can get τ_c for the coarse-grained classification head in the same way, after which the classification loss can be computed according to Eq. (3).

Based on only one dynamically changed parameter K , the thresholds τ_f and τ_c for all classes can be learned adaptively. The memory-bank mechanism can help the model acquire a good approximation of the global threshold for all samples with neglectable computational cost.

Note that the number of highly-confident samples that join the training matters, especially in the SSL settings. For two independent experiments, the dynamic proportion strategy in our DualMatch can select the same number of samples for training in different epochs. However, the fixed threshold in FixMatch or adaptive threshold in FlexMatch cannot guarantee this since their prediction is highly related to the random initialized network parameters. Besides, FixMatch and FlexMatch are more sensitive to manually set threshold than our DualMatch. For example, the simple threshold warm-up strategy in FlexMatch can lead to more than 1% difference in the results on CIFAR100. The fixed threshold in FixMatch can cause more than 2% difference on CIFAR10. As a comparison, our method is insensitive to parameters, which will be verified in experiments. In this case, our threshold learning strategy can lead to more stable results.

3.4 HIERARCHICAL LABEL BASED GRAPH MATCHING FOR CONTRASTIVE LEARNING

Consistency learning can only lead to instance-level correspondence in the feature space, which neglects the inter-class relationship learning and makes the representation not discriminative enough. In this situation, samples that belong to the same category might be of large variance and the margin between categories is not clear. Therefore, we propose a contrastive learning loss for SSL by leveraging pseudo-label information. According to the supervised contrastive learning Khosla et al. (2020), the way to incorporate label information matters, and it is very important to construct a precise graph to guide the contrastive learning based on the pseudo-labels.

For each sample u_b , given its fine-grained classification pseudo-label $\hat{q}_b^f$, we construct the fine-grained affinity graph $W^f \in \mathbb{R}^{\mu B \times \mu B}$ by:

$$W_{bj}^f = \begin{cases} 1 & \text{if } \hat{q}_b^f = \hat{q}_j^f, \\ 0 & \text{otherwise,} \end{cases} \quad (6)$$

where the edge between two samples equals to 1 only when they share the same fine-grained pseudo-label, which also denotes that they are positive pairs. Note that we do not assign any constraint on

Algorithm 1 Training algorithm for DualMatch.

```

1: Input:  $\mathcal{X} = \{(x_m, y_m) : m \in (1, \dots, M)\}$ ,  $\mathcal{U} = \{u_l : l \in (1, \dots, L)\}$   $M$  labeled data and  $L$ 
   unlabeled data.
2: while not reach the maximum iteration do
3:   for  $b \in \{1, \dots, B\}$  do
4:     Compute prediction  $q_{x_b}^f = h_f(f(\alpha(x_b)))$ ,  $q_{x_b}^c = h_c(f(\alpha(x_b)))$  for each labeled sample.
5:   end for
6:   for  $b \in \{1, \dots, \mu B\}$  do
7:     Compute the maximum probability  $\tilde{q}_b^f = \max h_f(f(\alpha(u_b)))$ , and  $\tilde{q}_b^c = \max h_c(f(\alpha(u_b)))$ .
8:     Update the memory-bank  $Q_{MB}^f, Q_{MB}^c$  by  $\tilde{q}_b^f, \tilde{q}_b^c$ .
9:     Update  $K$  according to the current number of epochs.
10:    Compute  $\tau_f = \text{percentile}(Q_{MB}^f, K)$  and  $\tau_c = \text{percentile}(Q_{MB}^c, K)$ .
11:    for  $j \in \{1, \dots, \mu B\}$  do
12:      Calculate  $W_{bj}^f, W_{bj}^c$ , and  $W_{bj}$  according to Eqs. (6) and (7).
13:    end for
14:  end for
15:  Compute  $L_u^f, L_u^c, L_u^{ctr}$  via Eqs. (1), (3), and (8).
16:  Update parameters  $\theta_f, \theta_g, \theta_{h_c}, \theta_{h_f}$  by optimizing Eq. (4) based on SGD.
17: end while
18: Return: A deep SSL model with desirable parameters.

```

the pseudo-label during the above graph construction, so its accuracy could be improved. We further turn to the hierarchical label for help and propose a graph matching strategy.

For the hierarchical labels, it can not only provide extra supervision information, but can also correct the affinity graph. Similar to Eq. (6), we can construct the coarse-grained affinity graph W^c . For each coarse-grained category, it contains several fine-grained classes. In this case, if two samples belong to the same fine-grained class, then they should have the same coarse-grained pseudo-label. However, this relationship is not always satisfied by the model, especially at the early training stage. So we can take advantage of W^c to correct W^f , and we name this process as graph matching, which can be formulated as:

$$W_{bj} = \begin{cases} 1 & \text{if } W_{bj}^f = 1 \text{ and } W_{bj}^c = 1, \\ 0 & \text{otherwise.} \end{cases} \quad (7)$$

Give the precise graph W , two samples are positive pair only if $W_{bj} = 1$, otherwise they are negative pairs. Contrastive learning Caron et al. (2020); Liu et al. (2021); He et al. (2020) aims to minimize the distance between positive pairs and maximize the distance between negative pairs. Denote $z_b = g(f(\mathcal{A}(u_b)))$ and $z'_b = g(f(\mathcal{A}'(u_b)))$ as the output features after projection head $g(\cdot)$ for sample u_b under two different strong augmentations. The graph matching based contrastive learning loss can be written as:

$$\mathcal{L}_u^{ctl} = -\frac{1}{\mu B} \sum_{b=1}^{\mu B} \left(\frac{1}{\sum_j W_{bj}} \log \frac{\sum_{j=1}^{\mu B} W_{bj} \exp((z_b \cdot z'_j)/t)}{\sum_{j=1}^{\mu B} (1 - W_{bj}) \exp(z_b \cdot z'_j/t)} \right), \quad (8)$$

where we also investigate the self-consistency between different augmentations when $b = j$.

We summarize the overall training process in Algorithm 1. By making full use of the hierarchical label, our method benefits from the following three aspects: (1) The coarse-grained classification head can provide extra supervision for the network learning, which is important for the situation of limited labeled samples; (2) By the graph matching of fine-grained and coarse-grained results, we can learn a more accurate graph, which can well filter the wrong relations and reduce the influence of noisy correspondence; (3) Contrastive learning with precise graph connections can lead to more discriminative feature representation.

Table 1: Error rates comparison on CIFAR-10, CIFAR-100, and STL-10.

Methods Label Amount	CIFAR-10			CIFAR-100@WRN-28-2			CIFAR-100@WRN-28-8			STL-10
	40	250	1000	400	2500	10000	400	2500	10000	1000
MixMatch (NeurIPS'19)	36.19± 6.48	13.63± 0.59	6.66± 0.26	-	-	-	67.61± 1.32	39.94± 0.37	28.31± 0.33	61.98± 8.29
FixMatch (NeurIPS'20)	13.91± 3.37	5.07± 0.65	4.26± 0.05	56.34± 2.12	34.53± 0.31	27.89± 0.10	48.85± 1.75	28.29± 0.11	22.60± 0.12	34.62± 0.42
CoMatch (ICCV'21)	6.91± 1.39	4.91± 0.33	4.56± 0.20	58.46± 2.31	36.84± 0.43	31.6± 0.14	41.89± 2.34	28.37± 0.35	20.86± 0.36	20.20± 0.38
FlexMatch (NeurIPS'21)	4.97± 0.06	4.98± 0.09	4.19± 0.01	49.23± 2.58	32.51± 0.20	26.58± 0.11	39.94± 1.62	26.49± 0.20	21.90± 0.15	-
DP-SSL (NeurIPS'21)	6.54± 0.98	4.78± 0.26	4.23± 0.20	-	-	-	43.17± 1.29	28.00± 0.79	22.24± 0.31	-
DualMatch (ours)	5.98± 0.19	4.91± 0.13	4.48± 0.10	45.23± 0.28	31.32± 0.47	24.84± 0.27	36.57± 0.41	24.10± 0.10	19.92± 0.29	10.36± 0.31

4 EXPERIMENTS

We conducted extensive experiments on four commonly-used SSL image classification datasets, including CIFAR-10 Krizhevsky et al. (2009), CIFAR-100, STL-10 Coates et al. (2011), and ImageNet Deng et al. (2009), under different amounts of labeled data and backbones.

Datasets. The CIFAR-10 dataset has 60,000 images that belong to 10 fine-grained classes. Each image has the size of 32×32 . Similarly, CIFAR-100 consists of 60,000 images corresponding to 100 fine-grained classes. STL-10 contains 5,000 labeled images of 10 classes and 100,000 unlabeled images with size 96×96 . For the hierarchical labels, CIFAR-100 naturally has 20 coarse-grained categories. As for CIFAR-10 and STL-10, we manually summarized two super classes, including animal and vehicle. For the ImageNet dataset, its hierarchical structure is unbalanced, so we first found these coarse-grained classes that contain at least 10 fine-grained classes. After sorting based on names, we selected the first 20 super classes as well as the first 10 fine-grained categories in each super class. Therefore, this ImageNet subset contains 20 coarse-grained categories and 200 fine-grained categories of about 256,483 images. We also tested the superiority of the proposed threshold learning strategy on the full ImageNet dataset.

Compared Methods. We mainly compared the results with current state-of-the-art SSL methods published in recent three years, including MixMatch Berthelot et al. (2019), FixMatch Sohn et al. (2020), CoMatch Li et al. (2021), DP-SSL Xu et al. (2021), FlexMatch Zhang et al. (2021).

Implementation Details. For fair comparison, we adopted the similar settings following FlexMatch and CoMatch. For CIFAR-10, we used WideResNet (WRN)-28-2 Zagoruyko & Komodakis (2016). For CIFAR-100, both WRN-28-2 and WRN-28-8 are adopted. ResNet-18 He et al. (2016) is used for STL-10, which is the same as CoMatch Li et al. (2021) since it has much lower computation cost compared to the WRN-37-2 used in Zhang et al. (2021). For ImageNet, ResNet-50 is adopted. The total training step is 2^{20} . The size of memory-bank N is set to 50,000. The commonly used distribution alignment strategy Berthelot et al. (2020) is also adopted in our method. For all these datasets except ImageNet, we used the standard stochastic gradient descent (SGD) Sutskever et al. (2013); Goyal et al. (2017) with a momentum of 0.9 for all experiments. The initial learning rate is set to 0.03 and a cosine learning rate decay schedule is adopted. The batch size is set to 64. μ is set to 7. We randomly run the experiments for three times and reported the average result. For ImageNet, we used the same settings as CoMatch, where the initial learning rate is set to 0.1 with weight decay Zhang et al. (2018) $1e-5$, the batch size is 160, and $\mu = 4$. For CIFAR-100 and ImageNet, $K\%$ is initialized as 5%, and linearly increases to 80% until the 100-th epoch. At the t -th epoch, $K^t = 5 + t * 0.75$ when $0 < t \leq 100$, and $K^t = 80$ when $t > 100$. For other two datasets, the upper value for $K\%$ is set to 95% since they are much simpler.

Augmentation. Our augmentation strategy is the same as CoMatch. For the weak augmentation, we used the standard crop-and-flip is adopted. For two kinds of strong augmentations $\mathcal{A}$ and $\mathcal{A}'$, RandAugment Cubuk et al. (2020) and augmentation strategy in SimCLR Chen et al. (2020) (random color jittering and grayscale conversion) are adopted, respectively.

4.1 MAIN RESULTS

In Table 1, we presented the semi-supervised classification results on CIFAR-10 under WRN-28-2, CIFAR-100 under both WRN-28-2 and WRN-28-8, and STL-10 under ResNet-18. We can see that on the CIFAR-100 dataset, our proposed DualMatch method achieves much better results than all these related methods under all various number of labeled samples and two different backbones. Specifically, compared with the strong baseline FlexMatch, our method can achieve an average 7.47% error rate reduction on these six settings of CIFAR-100. The error rate reduction is even

Table 2: Error rates results on ImageNet.

Method	Top1		Top5	
	Label fraction	1% 10%	1% 10%	
MixMatch	-	-	-	-
FixMatch	60.81	34.33	35.84	14.83
CoMatch	42.88	26.48	17.99	9.23
FlexMatch	54.37	29.82	31.61	12.27
DP-SSL	-	-	-	-
DualMatch(ours)	34.18	24.17	12.33	7.64

Table 3: Effect of each module.

Modules					Error rate
Graph matching	Coarse label	Fixed threshold	Fixed proportion	Dynamic proportion	
✓	✓	✓			50.14
✓	✓		✓		49.56
				✓	48.83
	✓			✓	47.14
✓	✓			✓	45.23

larger over FixMatch, which is 15.29% on average. The above results can well demonstrate the effectiveness of DualMatch, especially when the number of classes is large and the hierarchical structure is relatively balanced.

For STL-10, we simply copy the results of several related methods from CoMatch. We can see that our proposed method achieves the best results among these compared methods. Specifically, We can achieve the error rate of 10.36%, while the second best result is 20.20% realized by CoMatch. Moreover, we can observe that the error rate results on STL-10 are higher than that of CIFAR-10. The reason is that there exist out-of-distribution images in the unlabeled set of STL-10, which makes it more challenging and realistic. While the dataset is more challenging, our superiority is more significant, which can also verify the effectiveness of our method from another aspect.

On CIFAR-10, our results are also comparable with FixMatch. The main reason is that CIFAR-10 is relatively simple, where it only has 2 coarse-grained and 10 fine-grained classes. The difference between two super categories (animal and vehicle) is very clear. In this case, the accuracy of both fine-grained and coarse-grained classification is very high, and the coarse label graph cannot provide useful information in graph matching. Even though, our results are better than FixMatch under 40 and 250 labeled samples, which validates the superiority of our memory-bank based near-global threshold learning strategy, especially under limited labeled samples.

We also noticed that our results are very stable on all these datasets, where the maximum variance is less than 0.5%. In comparison, the variance of FlexMatch is larger than 2.5% and 1.6% on CIFAR-100 with 4 labeled samples per class under two different backbones, respectively. Other methods also have the similar disadvantage, especially under the situation of very limited labeled samples. The reason for this phenomenon can be found in the end of Subsection 3.3.

4.2 RESULTS ON IMAGENET

We validated our method on the more challenging ImageNet dataset, where we used a subset containing 20 super and 200 general classes to construct a balanced hierarchical structure. For a fair comparison, we used the similar parameters as CIFAR-100 for all these compared methods, which can also verify its robustness. The results are shown in Table 2. DualMatch achieves the best results among all these compared methods under both 1% and 10% settings. For example, the top-1 error rate of our method with 1% labeled samples is 34.18%, which significantly surpasses the result 42.88% of the second best method CoMatch.

We also conducted experiments on the full ImageNet dataset to verify the effectiveness of our memory-bank based threshold learning strategy. We adopted the same settings as FlexMatch, where 100K labeled samples are randomly choosed for evaluation. We simply replace the threshold learning mechanism in FixMatch or FlexMatch with our memory-bank based threshold learning strategy without the use of hierarchical labels. The top-1 error rate of our method is **27.18%**, which is much better than the results of FlexMatch 41.85% and FixMatch 43.66%.

4.3 ABLATION STUDY

We conducted experiments to verify the effectiveness of each proposed module on CIFAR-100 with 400 labeled samples under WRN-28-2. Experiments in the following subsection are also under this setting. The results are presented in Table 3. Based on the error rate on lines 1, 2, and 5, we can see that our dynamic proportion based adaptive threshold learning strategy is much better than the fixed

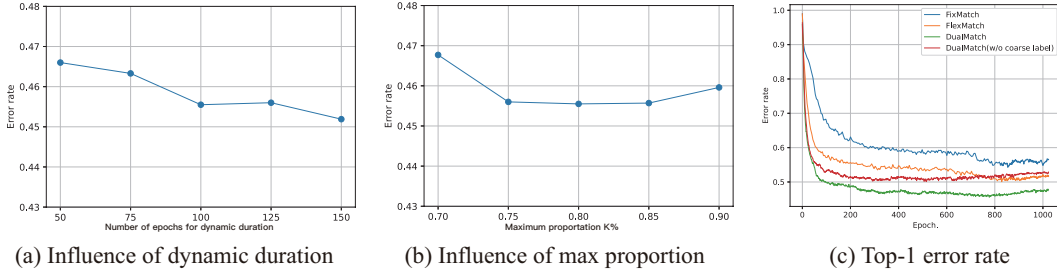


Figure 3: Parameter sensitivity and convergence analysis on CIFAR-100.

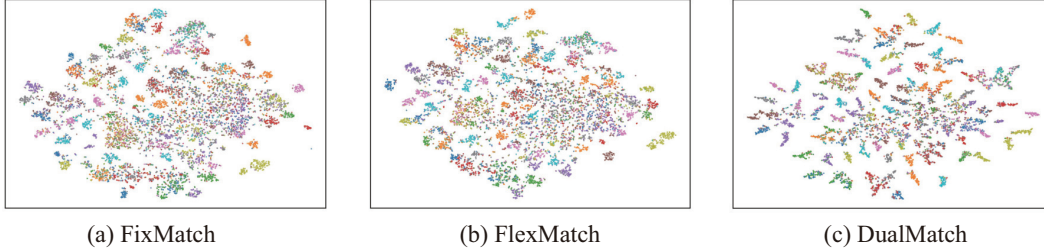


Figure 4: t-SNE visualization for feature representation on CIFAR-100.

proportion and fixed threshold strategies. By comparing the results on lines 3, 4, and 5, both coarse-grained classification head and graph matching strategy can obviously improve the performance.

4.4 QUALITATIVE ANALYSIS

Parameter Sensitivity Analysis. For the weight parameters before each loss term, we directly set all of them to 1 for all datasets, which demonstrates that our method is very robust and not sensitive to these parameters. For the number of epochs for dynamic duration and the maximum proportion $K\%$, we tested their influence and showed the results in Figure 3 (a) and (b). We can observe that the influence is neglectable for these two hyper-parameters. For the dynamic duration, the performance could even be improved if we set it to 150, which further demonstrates the robustness of DualMatch.

Convergence Analysis. We presented the change of top-1 error rate during the training process in Figure 3 (c), and compared it with FixMatch and FlexMatch. We can see that our DualMatch converges very fast, which only needs less than 400 epochs to achieve almost the best performance. In contrast, both FixMatch and FlexMatch need more than 800 epochs to get the best results. In this case, our method is much faster than them, which can be accumulated to the fact that we adopted the dynamic proportion mechanism and a large proportion of samples join the training after 100 epochs.

Visualization. The proposed graph matching based contrastive learning module can help the model learn discriminative feature representations. To verify this, we visualized the learned features by t-SNE Maaten & Hinton (2008) and compared the results with FixMatch and FlexMatch in Figure 4. It is obvious that the inter-class distance of our method is much larger than these of other two methods, which can lead to better results.

5 CONCLUSION AND FUTURE WORK

In this paper, we came up with a new semi-supervised learning method DualMatch, which performs both instance-level prediction matching and contrastive graph-level matching. We first introduced a memory-bank based strategy to learn near-global adaptive threshold, which can efficiently select highly-confident samples for pseudo-label consistency. Besides, we utilized the hierarchical label structure to improve the SSL from two aspects, including an extra head for fine-grained classification and graph matching for contrastive feature learning. Extensive experiments on various benchmarks demonstrate our superiority in classification results and robustness. In the future, we would like to improve our method under the unbalanced hierarchical label structure, where different super classes may have different number of fine-grained categories.

REFERENCES

- Mahmoud Assran, Mathilde Caron, Ishan Misra, Piotr Bojanowski, Armand Joulin, Nicolas Ballas, and Michael Rabbat. Semi-supervised learning of visual features by non-parametrically predicting view assignments with support samples. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 8443–8452, 2021.
- Philip Bachman, Ouais Alsharif, and Doina Precup. Learning with pseudo-ensembles. In *Advances in Neural Information Processing Systems*, volume 27, 2014.
- Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the International Conference on Machine Learning*, pp. 41–48, 2009.
- David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin A Raffel. Mixmatch: A holistic approach to semi-supervised learning. In *Advances in Neural Information Processing Systems*, volume 32, 2019.
- David Berthelot, Nicholas Carlini, Ekin D Cubuk, Alex Kurakin, Kihyuk Sohn, Han Zhang, and Colin Raffel. Remixmatch: Semi-supervised learning with distribution alignment and augmentation anchoring. In *Proceedings of the International Conference on Learning Theory*, 2020.
- Zhaowei Cai, Avinash Ravichandran, Subhansu Maji, Charles Fowlkes, Zhuowen Tu, and Stefano Soatto. Exponential moving average normalization for self-supervised and semi-supervised learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 194–203, 2021.
- Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. In *Advances in Neural Information Processing Systems*, volume 33, pp. 9912–9924, 2020.
- Olivier Chapelle, Mingmin Chi, and Alexander Zien. A continuation method for semi-supervised svms. In *Proceedings of the International Conference on Machine Learning*, pp. 185–192, 2006.
- Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *Proceedings of the International Conference on Machine Learning*, pp. 1597–1607, 2020.
- Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*, pp. 215–223, 2011.
- Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 702–703, 2020.
- Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009.
- Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch sgd: Training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677*, 2017.
- Yves Grandvalet and Yoshua Bengio. Semi-supervised learning by entropy minimization. In *Advances in Neural Information Processing Systems*, volume 17, 2004.
- Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016.
- Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9729–9738, 2020.

- Zijian Hu, Zhengyu Yang, Xuefeng Hu, and Ram Nevatia. Simple: Similar pseudo label exploitation for semi-supervised classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15099–15108, 2021.
- Ahmet Iscen, Giorgos Tolias, Yannis Avrithis, and Ondrej Chum. Label propagation for deep semi-supervised learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5070–5079, 2019.
- Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. In *Advances in Neural Information Processing Systems*, volume 33, pp. 18661–18673, 2020.
- Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.
- Samuli Laine and Timo Aila. Temporal ensembling for semi-supervised learning. In *International Conference on Learning Representations*, 2017.
- Dong-Hyun Lee et al. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Proceedings of the International Conference on Machine Learning Workshop on Challenges in Representation Learning*, volume 3, pp. 896, 2013.
- Junnan Li, Caiming Xiong, and Steven CH Hoi. Comatch: Semi-supervised learning with contrastive graph regularization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9475–9484, 2021.
- Xiao Liu, Fanjin Zhang, Zhenyu Hou, Li Mian, Zhaoyu Wang, Jing Zhang, and Jie Tang. Self-supervised learning: Generative or contrastive. *IEEE Transactions on Knowledge and Data Engineering*, 2021.
- Yucen Luo, Jun Zhu, Mengxi Li, Yong Ren, and Bo Zhang. Smooth neighbors on teacher graphs for semi-supervised learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8896–8905, 2018.
- Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *JMLR*, 9(Nov):2579–2605, 2008.
- Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, and Shin Ishii. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(8):1979–1993, 2018.
- Islam Nassar, Samitha Herath, Ehsan Abbasnejad, Wray Buntine, and Gholamreza Haffari. All labels are not created equal: Enhancing semi-supervision via label grouping and co-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7241–7250, 2021.
- Antti Rasmus, Mathias Berglund, Mikko Honkala, Harri Valpola, and Tapani Raiko. Semi-supervised learning with ladder networks. In *Advances in Neural Information Processing Systems*, volume 28, 2015.
- Mehdi Sajjadi, Mehran Javanmardi, and Tolga Tasdizen. Regularization with stochastic transformations and perturbations for deep semi-supervised learning. In *Advances in Neural Information Processing Systems*, volume 29, 2016.
- Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin A Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. In *Advances in Neural Information Processing Systems*, volume 33, pp. 596–608, 2020.
- Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. On the importance of initialization and momentum in deep learning. In *Proceedings of the International Conference on Machine Learning*, pp. 1139–1147, 2013.

- Fariborz Taherkhani, Ali Dabouei, Sobhan Soleymani, Jeremy Dawson, and Nasser M Nasrabadi. Self-supervised wasserstein pseudo-labeling for semi-supervised image classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12267–12277, 2021.
- Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *Advances in Neural Information Processing Systems*, volume 30, 2017.
- Qizhe Xie, Zihang Dai, Eduard Hovy, Thang Luong, and Quoc Le. Unsupervised data augmentation for consistency training. In *Advances in Neural Information Processing Systems*, volume 33, pp. 6256–6268, 2020.
- Yi Xu, Jiandong Ding, Lu Zhang, and Shuigeng Zhou. Dp-ssl: Towards robust semi-supervised learning with a few labeled samples. In *Advances in Neural Information Processing Systems*, volume 34, 2021.
- Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In *Proceedings of the British Machine Vision Conference*, 2016.
- 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*, volume 34, 2021.
- Guodong Zhang, Chaoqi Wang, Bowen Xu, and Roger Grosse. Three mechanisms of weight decay regularization. *arXiv preprint arXiv:1810.12281*, 2018.
- Tianyi Zhou, Shengjie Wang, and Jeff Bilmes. Time-consistent self-supervision for semi-supervised learning. In *Proceedings of the International Conference on Machine Learning*, pp. 11523–11533, 2020.

A APPENDIX

A.1 DETAILED EXPERIMENTAL SETTINGS

In Table 3, we presented the detailed settings of hyperparameters in our semi-supervised learning experiments.

Table 3: Algorithm parameters

Dataset	CIFAR-10	STL-10	CIFAR-100	CIFAR-100	ImageNet
Model	WRN-28-2	ResNet-18	WRN-28-2	WRN-28-8	ResNet-50
Batch Size	64		64		160
μ	7		7		4
Max proportion	0.95		0.8		
Initial proportion			0.05		
Dynamic duration			100		

A.2 PARAMETER SENSITIVITY ANALYSIS

As described in Subsection 3.2, the overall objective function of DualMatch is:

$$\min_{\theta_f, \theta_g, \theta_{h_c}, \theta_{h_f}} \mathcal{L} = \mathcal{L}_s + \alpha \mathcal{L}_u^f + \beta \mathcal{L}_u^c + \gamma \mathcal{L}_u^{ctl}, \quad (9)$$

where α , β , and γ are three balanced parameters. In experiments, we simply set the weights to 1 for various loss terms on all datasets, which can well demonstrate the robustness and insensitivity to these parameters of our method. We further conduct experiments to analyze the sensitivity. We vary each parameter in the range of [0.75, 1, 1.5, 2] while fixed other weights as 1. The error rate results on CIFAR100 under WRN-28-2 is presented in Table 4. The reported error rate results in the paper is 45.23%. We can see that our results can be further improved to 44.53% based on other settings of these parameters. Besides, we can see that when these weights vary in a certain range, our results are relatively stable.

Table 4: Error rate on CIFAR100 under WRN-28-2 with different weights. We vary each parameter and fix other weights as 1.

	0.75	1	1.5	2
α	46.37	45.23	46.51	44.53
β	46.60	45.23	46.15	44.96
γ	45.61	45.23	44.99	45.05

A.3 GENERALIZATION ANALYSIS

To evaluate the generalization ability of our propose method, we conducted additional experiments under domain shift settings. We use the model trained on the CIFAR100 dataset under WRN-28-2 to test the results on the sketch domain. The ImageNet-Sketch dataset consists of 50,000 sketch images, 50 images for each of the 1,000 ImageNet classes. We selected the sketch images from ImageNet-Sketch that belong to the overlapped categories between CIFAR100 and ImageNet for testing. Note that one category in CIFAR100 may correspond to several classes in ImageNet. Specifically, 11,950 images that belong to 62 categories of CIFAR100 are selected. The accuracy for DualMatch, FlexMatch, and FixMatch are 15.55%, 12.31%, and 8.75%, respectively. We can see that our method also achieves much better results than these two strong baselines.

A.4 RESULTS ON FULL IMAGENET WITH MEMORY-BANK BASED THRESHOLD LEARNING

We also conducted experiments on the full ImageNet dataset to verify the effectiveness of our memory-bank based threshold learning strategy. We adopted the same settings as FlexMatch, where 100K labeled samples are randomly choosed for evaluation. We simply replace the threshold learning mechanism in FixMatch or FlexMatch with our memory-bank based threshold learning strategy without the use of hierarchical labels. The results are shown in Table 5. We can find that the top-1 error rate of our method is 27.18%, which is more than 10% lower than the results of FlexMatch 41.85%. We can also see the similar improvement over FlexMatch under top-5 results. The above results can well demonstrate the superiority of our threshold learning strategy.

Table 5: Error rate comparison on the full ImageNet dataset. 'DualMatch-thres' denotes our method only with the memory-bank based threshold learning strategy.

Method	Top-1	Top-5
FixMatch	43.66	21.80
FlexMatch	41.85	19.48
DualMatch-thres	27.18	9.00

A.5 GRAPH MATCHING EXAMPLES

As illustrated in Section 3.4, our graph matching strategy can learn an accurate affinity graph by taking advantage of coarse pseudo label graph to correct the fine pseudo label graph. Specifically, when there are many indistinguishable super classes, two samples might have the same fine-grained pseudo-label, but their coarse-grained pseudo-label are different. We show some real cases in Figure 5. We can see that both the fourth and fifth images are predicted as 'turtle' since they are very similar, while the fourth image is misclassified. However, they have different coarse pseudo labels. In this case, by our graph matching strategy, they will not be regarded as the positive pairs to guide the contrastive learning, which can well demonstrate the effect of our graph matching strategy.

								
cpl: $\hat{q}_\alpha^c$	fish				reptiles			
y	aquarium fish	shark	ray	flatfish	turtle	dinosaur	crocodile	lizard
fpl: $\hat{q}_\alpha^f$	aquarium fish	cattle	ray	turtle	turtle	crocodile	crocodile	lizard

Figure 5: Predicted results of fine-grained and coarse-grained classifications. 'cpl', y , and 'fpl' correspond to coarse-grained pseudo label, ground-truth label, and fine-grained pseudo label, respectively. The misclassified fine-grained pseudo-label can be corrected by coarse-grained pseudo-label in graph construction.

A.6 COMPUTATIONAL ANALYSIS OF THE MEMORY-BANK BASED THRESHOLD LEARNING

For the threshold computation in step 10 of the Algorithm 1, its time cost can be neglectable. Specifically, we use the function "torch.topk" in PyTorch library to select the $K\%$ largest values from the memory bank to compute the threshold. The computational complexity for this operation is $\mathcal{O}(n+t)$, where $t = n * K\%$, and n denotes the size of the memory bank. In our experiments, we set $n = 50,000$, and it only needs 2.6ms for each minibatch to compute the threshold. Moreover, this operation in our method is irrelevant to the dataset.

In comparison, FlexMatch needs the predicted probability of the whole dataset to compute the threshold for each category by the "counter" function, whose computational complexity is $\mathcal{O}(m+c)$, where m and c denote the number of training samples and classes, respectively. Since m is generally larger than n , its computational complexity is higher than ours. Specifically, it generally needs 2.66ms on CIFAR and 109ms on ImageNet to execute this function. For the whole training, it costs extra 0.7, and 31.7 hours for such an operation on these two datasets.

A.7 DETAILED RESULTS UNDER OTHER EVALUATION METRICS

To comprehensively evaluate the performance under various metrics, we further reported the Precision, Recall, and F1-score on CIFAR-100 with 400 labels samples under two different backbones. As shown in Table 6, DualMatch also achieves the best performance under these metrics.

Table 6: Precision, Recall, F1-score comparison on CIFAR-100 with 400 labeled samples.

Model	WRN-28-2			WRN-28-8		
	Precision	Recall	F1-score	Precision	Recall	F1-score
FixMatch	0.5177	0.4574	0.4129	0.5615	0.54	0.4983
FlexMatch	0.4914	0.4991	0.4886	0.5930	0.6027	0.5931
DualMatch	0.5585	0.5536	0.5497	0.6641	0.6557	0.6541

A.8 CLASS-WISE ACCURACY COMPARISON

Table 7 shows a detailed class-wise accuracy comparison on CIFAR-10. Even without assigning class-specific threshold as FlexMatch does, we still achieved competitive results on those hard-to-learn classes.

Table 7: Class-wise accuracy comparison on CIFAR-10 40-label split.

Class Number	0	1	2	3	4	5	6	7	8	9
FixMatch	0.964	0.982	0.697	0.852	0.974	0.890	0.987	0.970	0.982	0.981
FlexMatch	0.967	0.980	0.921	0.866	0.957	0.883	0.988	0.975	0.982	0.968
DualMatch	0.975	0.978	0.890	0.876	0.966	0.871	0.985	0.971	0.982	0.981

Figure 6 shows the smooth growth of the accuracy of our method for each category during the training on the CIFAR-10 dataset.

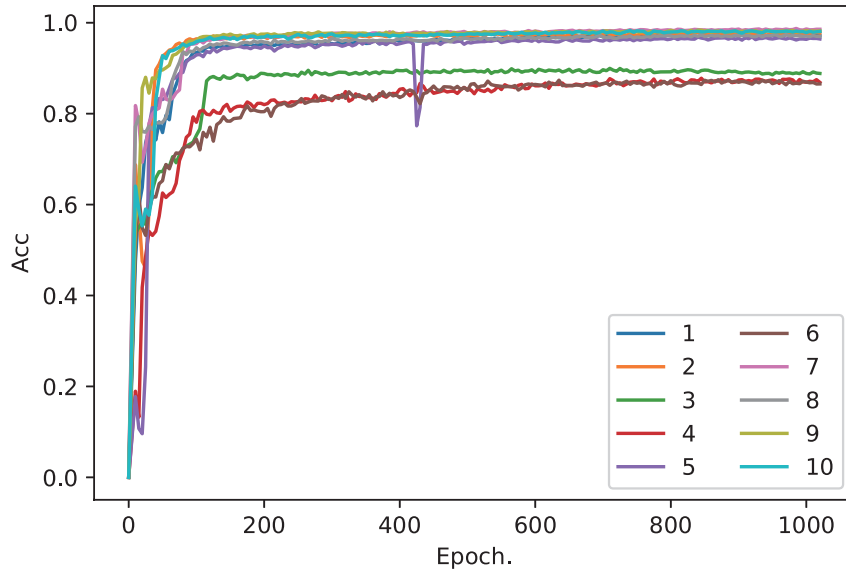


Figure 6: The class-wise accuracy convergence curve on CIFAR-10 with 40 labels.

We also presented the class-wise accuracy on CIFAR-100, which is shown in Figures 7 and 8. We can see that the coarse-grained classification accuracy is higher than the fine-grained results, which can provide useful information for guidance and correctness. Besides, according to Figure 8, our method can converge very fast and is very stable.

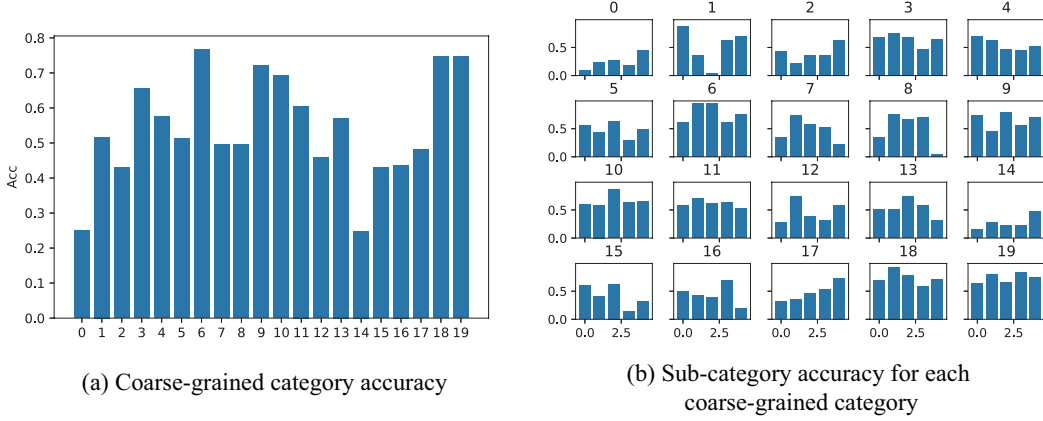


Figure 7: Accuracy of each category on CIFAR-100.

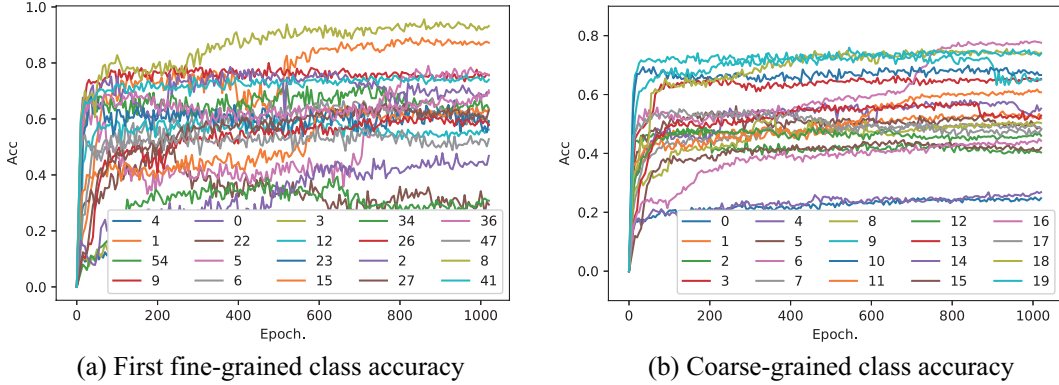


Figure 8: The convergence curves for selected classes on CIFAR-100.

A.9 THE CONSTRUCTION OF IMAGENET SUBSET AND VISUALIZATION

The detailed construction process of the ImageNet subset can be summarized as follows:

- Step 1: The categories of the ImageNet dataset have a tree structure. We iterate through all non-leaf nodes starting from the root node, and record the number of leaf nodes they cover on all non-leaf nodes.
- Step 2: Delete all non-leaf nodes that cover the number of leaf nodes greater than K (In the operation, we set $K=112$). These deleted non-leaf nodes have coarse granularity.
- Step 3: Delete all non-leaf nodes that have less than 10 leaf nodes.
- Step 4: Delete all non-leaf nodes whose children contain non-leaf nodes to ensure that all non-leaf nodes do not overlap.
- Step 5: Sort these non-leaf nodes in alphabetical order based on category identity. Then take the top 20 non-leaf nodes as the coarse-grained category.
- Step 6: Sort leaf nodes under these selected 20 non-leaf nodes with the same rule. Then take the top 10 leaf nodes of each non-leaf node as their fine-grained categories.

This process finally retains 200 leaf nodes and 20 non-leaf nodes, where each leaf node corresponds to a category in ImageNet1k and each non-leaf node corresponding to a coarse-grained category. The number of 200 fine-grained and 20 coarse-grained classes are set empirically. The code for construction and detailed categories will be released after acceptance.

We also visualized the training classes of ImageNet obtained by our algorithm in Figure 9. Each row corresponds to one coarse-grained class, which contains 20 images from 10 different fine-grained classes.



Figure 9: Visualization of the balanced ImageNet dataset. Each row corresponds to one coarse-grained category with 10 fine-grained categories, where each fine-grained category has two images for visualization.

A.10 DETAILED COMPASION WITH CoMATCH

CoMatch achieves very good performance for the task of semi-supervised learning. Its main contributions lie in two aspects: memory-smoothed pseudo labeling and graph-based contrastive learning to learn better representation.

In comparison, our contributions and differences lie in two aspects, including the memory-bank based dynamic proportion strategy and hierarchical label based graph matching. Both of these two mechanism are effective and have not been studied in the field of semi-supervised learning. First, different from FixMatch and FlexMatch which are based on a manually set threshold, our dynamic proportion strategy can select the same number of samples for training in different epochs for two independent experiments and is more robust to parameters, leading to more stable results. Second, our hierarchical labels based graph matching can construct a more accurate affinity graph for contrastive learning, contributing to more discriminative feature representation and better performance. We need to mention that our graph matching strategy is significantly different from that in CoMatch. On the one hand, our matching denotes the consistency between coarse-grained graph and fine-grained graph to get more correct supervision information, while CoMatch computes a similarity graph to constrain the feature representation. On the other hand, the way to construct the graph is also different. CoMatch adopts the similarity between probabilities to regularize the similarity between features. It also presents a memory-smoothed pseudo-labeling strategy to use similarity in feature space to update the probabilities in label space. In this case, there are so many bidirectional interactions between features and probabilities and it cannot well solve the issue claimed in CoMatch that “since the features are highly correlated with the class predictions, the same types of errors are likely to exist in both the feature space and the label space”. As a comparison, we directly

use the one-hot pseudo-label to construct the graph without the above complex interactions, which can weaken the above issue.