AdapMatch: Adaptive Bias Decoupling for Semi-Supervised Partial Label Learning under Unknown Class Distributions

Yuheng Jia[1*] **Yaodong Shen**[1*] **Jiahao Jiang**[1] **Hui Liu**[2] **Junhui Hou**[3†]

[1]Southeast University [2]Saint Francis University [3]City University of Hong Kong
{yhjia,yaodong.shen,jhjiang}@seu.edu.cn, h2liu@sfu.edu.hk, jh.hou@cityu.edu.hk

Abstract

Partial Label Learning (PLL) trains models from instances with candidate label sets containing the ground-truth, while Semi-Supervised Partial Label Learning (SSPLL) further leverages the available unlabeled data to prompt the performance. The previous SSPLL methods implicitly assume the unlabeled data are class-balanced distributed. However, in realistic scenarios, the class distribution of unlabeled data is unknown, resulting in performance collapse on minority classes for the existing SSPLL methods. We identify two coupled biases behind this failure: (i) pseudo-label selection bias, where high-confidence filtering tends to over-select samples from dominant classes, causing the training data to become increasingly skewed toward these classes; and (ii) disambiguation bias, where dominant classes are more likely to win within candidate sets, even when they are incorrect. Theoretically, we show that these two biases reinforce each other over training, forming a feedback loop that degrades generalization performance. To weaken this loop, we propose AdapMatch, which uses DAS to reduce the skew in training data caused by pseudo-label selection bias via prior-aware pseudo-label admission and CAD to lower the disambiguation error by suppressing candidate-set domination. Across five benchmarks, AdapMatch yields gains up to 41.31% at most , showing strong robustness to varying scale, imbalance, and ambiguity. The code is available at `https://github.com/yaodong-shen/AdapMatch`.

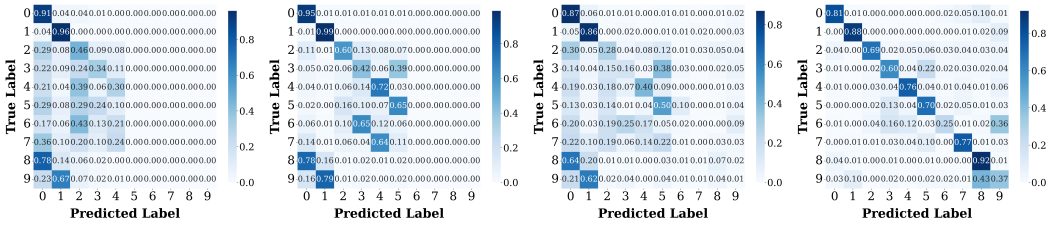
1 Introduction

Partial Label Learning (PLL) [1] is a weakly supervised paradigm where each instance is annotated with a candidate label set, only one of which is correct. This relaxed annotation form is useful when exact labeling is costly or ambiguous, such as in multimedia content analysis [2], web mining [3], and facial age estimation [4]. The central challenge is to resolve candidate-label ambiguity, for which prior methods mainly adopt disambiguation [5] or reweighting [6]. However, when labeled data is limited, these methods may lack sufficient supervision. Semi-Supervised Partial Label Learning (SSPLL) addresses this limitation by exploiting abundant unlabeled data, commonly using confident pseudo-labels of the unlabeled data to support candidate-label disambiguation.

Despite this progress, most SSPLL pipelines implicitly assume roughly balanced class frequencies. In realistic applications, class distributions are often unknown and highly non-uniform, e.g., long-tailed distributions where a few classes dominate many rare ones [7]. This mismatch between balanced assumptions and real-world class distributions substantially challenges SSPLL. Empirically, under non-uniform and unknown class frequencies: we observe severe degradation of existing SSPLL

*These authors contributed equally to this work.
†Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).



(a1) FixMatch+ (32.60%) (a2) ConCont (41.15%) (a3) FairMatch (43.74%) (a4) AdapMatch (82.46%)

Figure 1: Confusion matrices on CIFAR-10 with $\gamma = 10$, $\rho = 0.1$, and $q = 0.7$. Baselines (a1–a3) show strong tail-to-head mis-disambiguation, while ours (a4) preserves a clearer diagonal for minority classes. Here, γ, ρ, and q denote the imbalance factor, partial-label ratio, and ambiguity rate.

pipelines (Figure 1 (a1–a4)). This degradation is not simply due to the data being harder under imbalance; it reflects two failure modes in the standard SSPLL pipeline, which together create a strong preference for dominant classes and often drive minority-class accuracy close to zero.

Bias 1: Skewed pseudo-label admission (selection bias). SSPLL trains on partially labeled samples and pseudo-labeled unlabeled samples selected during training. When the labeled and unlabeled data are imbalanced, this selection process can be biased toward dominant classes, since their samples usually obtain higher confidence and are more likely to be admitted. Consequently, the selected pseudo-labeled samples become increasingly dominated by frequent classes, causing the effective training distribution to drift away from the true class distribution. Figure 2 shows this effect using two representative SSPLL baselines: the confidence-based ConCont over-selects head classes, whereas the class-wise fair FairMatch produces a nearly uniform selected distribution. Our method better matches the target prior and obtains the lowest KL divergence.

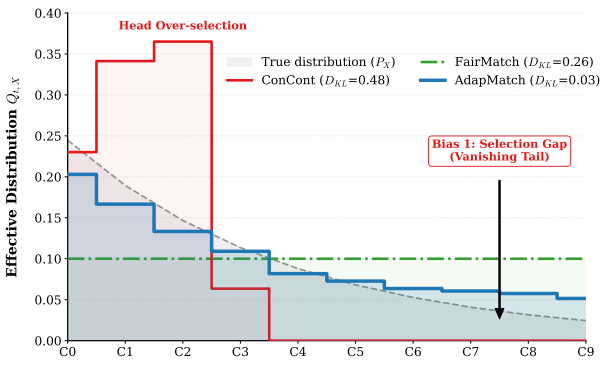


Figure 2: **Diagnosis of Selection Bias (Bias 1).** Class distribution of selected high-confidence samples on CIFAR-10 ($\gamma = 10$). Compared with representative SSPLL baselines, ours obtains the lowest KL divergence ($D_{KL} = 0.05$).

Bias 2: Candidate-set domination (disambiguation bias). In PLL, each candidate set contains the true label together with several false candidates. The model therefore has to decide which candidate is most likely to be correct. Under imbalanced and unknown class distributions, dominant classes often receive higher confidence. When such a dominant class is included as a false candidate, it may dominate the true label during disambiguation. This leads to directional errors from minority classes to dominant classes and further weakens the supervision for rare classes. Figure 1 illustrates this failure: existing methods frequently classify minority samples into head classes such as C0 and C1, whereas our method maintains a clearer diagonal, including for rare classes.

The two biases are closely connected. When pseudo-label admission is skewed, the model is trained on fewer minority samples and more dominant-class samples, which shifts the effective training distribution, as illustrated in Figure 2. This makes dominant classes more confident during training. Once a dominant class appears as a false candidate, it is more likely to override the true label, producing the minority-to-head errors shown in Figure 1. These errors then feed back into later pseudo-label selection by corrupting the model confidence used for admission. Section 3 formalizes this interaction as two coupled terms: an admission-induced distribution gap and a partial-label disambiguation error. This motivates our key principle: both pseudo-label admission and candidate-label disambiguation must be controlled jointly.

To this end, we propose **AdapMatch**, a collaborative bias-decoupling framework built upon two complementary components: the Distribution-Aware Selector (DAS) and the Calibration-Aware Disambiguator (CAD). Specifically, DAS mitigates selection bias (Bias 1) by aligning pseudo-label

admission with the estimated latent prior, preventing dominant classes from saturating the selected pseudo-labeled samples. CAD mitigates disambiguation bias (Bias 2) by calibrating the competition within candidate label sets, so that dominant-class false positives are less likely to override the ground-truth label. DAS and CAD are linked by self-correcting prior estimation: the estimated prior guides both pseudo-label admission and candidate-level disambiguation, while CAD-calibrated confidence updates the next-round prior. This enables joint control of admission skew and disambiguation bias. We further use consistency and interpolation-based regularization to stabilize training under residual pseudo-label noise, and apply uncertainty-aware entropy weighting to fuse DAS and CAD predictions at inference time. Extensive evaluations across five benchmarks demonstrate the robustness of AdapMatch, yielding absolute accuracy gains of up to 41.31% on CIFAR-10, 17.81% on STL-10, and 4.80% on ImageNet-127 over state-of-the-art baselines. Our contributions are summarized as follows:

1. **Problem setting.** We study SSPLL under *unknown* class distributions for the first time, and identify two coupled biases: skewed pseudo-label admission and dominant-class false positives within candidate sets, which drive minority-class collapse.
2. **Theoretical analysis.** We formalize the coupled dynamics between the admission-induced distribution gap and the partial-label disambiguation error, and connect them to target risk (Section 3). This motivates the *joint* control of admission skew and within-candidate domination in robust SSPLL.
3. **Model formulation.** Guided by this principle, we propose AdapMatch, which couples distribution-aware pseudo-label admission with prior-calibrated candidate disambiguation. Extensive experiments show consistent state-of-the-art performance, especially under severe ambiguity and class imbalance.

2 Related Work

Partial and semi-supervised partial label learning. Partial Label Learning (PLL) learns from candidate label sets that contain the unknown ground-truth label. Early PLL methods resolve ambiguity by averaging candidate predictions, identifying the latent true label, or exploiting feature-space smoothness [8–10]. Recent deep PLL methods further improve disambiguation with contrastive learning, prototype alignment, and consistency regularization, such as PiCO [11], ABL [12], and PLCR [13]. Semi-Supervised Partial Label Learning (SSPLL) extends PLL by using abundant unlabeled data. Wang [14] first formulated SSPLL with iterative label propagation, while ConCont [15] enables scalable deep SSPLL by treating unlabeled samples as fully ambiguous and training with pseudo-labeling, consistency regularization, and contrastive learning. FairMatch [16] improves pseudo-label selection through class-wise balanced high-confidence sampling. However, these SSPLL methods still largely rely on confidence-driven admission and are vulnerable when the underlying class distribution is unknown and imbalanced.

Long-tailed partial label learning. Long-Tailed Partial Label Learning (LTPLL) considers PLL under imbalanced class priors, where head classes dominate and tail classes are scarce. Representative methods mitigate imbalance through tail-aware sampling, distribution regularization, optimal transport, or head/tail classifier cooperation [17–20]. Although some studies partially consider data imbalance, under the combined challenges of long-tailed distributions and partial label ambiguity, these methods cannot directly leverage unlabeled data, and thus are not applicable to SSPLL tasks.

3 Theoretical Analysis

Let $\mathcal{X}$ be the input space and $\mathcal{Y} = \{1, \dots, C\}$ be the label space. Each sample is represented as (X, Y, S), where $X \in \mathcal{X}$, $Y \in \mathcal{Y}$ is the unknown ground-truth label, and $S \subseteq \mathcal{Y}$ is the candidate label set. For partially labeled samples, $Y \in S$ and S may contain false-positive labels. In SSPLL, we view an unlabeled sample as a special case with the full candidate set $S = \mathcal{Y}$.

Let P denote the target distribution over (X, Y, S), with class prior $\pi(y) = P(Y = y)$. At training round t, the model is updated on an effective distribution Q_t induced by the partially labeled data and the admitted pseudo-labeled samples, with class prior $\pi_t(y) = Q_t(Y = y)$. We define the

admission-induced distribution gap as

$$\delta_t := \|\pi_t - \pi\|_1. \tag{1}$$

It measures how much the effective training prior deviates from the target prior, e.g., when confidence-based admission over-selects head-class samples under unknown or long-tailed distributions.

Next, let $\hat{y}_t(x)$ be the class predicted by the score distribution used for candidate disambiguation. Since full-candidate samples with $S = \mathcal{Y}$ provide no informative candidate constraint, we evaluate disambiguation on genuine partial-label samples $U := \{|S| < C\}$ and denote $\rho_t := Q_t(U)$. The partial-label disambiguation accuracy is $A_t^{\mathrm{par}} := Q_t(\hat{y}_t = Y \mid U)$, and the corresponding error is

$$d_t := 1 - A_t^{\mathrm{par}} = Q_t(\hat{y}_t \neq Y \mid U). \tag{2}$$

Thus, d_t measures the error made when the learner must identify the true label from a non-trivial candidate set. Such errors can bias the confidence statistics used for later pseudo-label admission. With these definitions, we state the following theorem:

Theorem 3.1 (Coupled Bias Feedback)**.** *Suppose $\rho_t > 0$ and the full-candidate error satisfies $Q_t(\hat{y}_t \neq Y \mid |S| = C) \leq \zeta_t$. Under the regularity assumptions in Appendix G.4, there exist constants $\alpha, \gamma > 0$, a coefficient $\lambda_t > 0$, and non-negative residual terms r_t^{dis} and r_t^{adm} such that*

$$d_t \leq \alpha\delta_t + r_t^{\mathrm{dis}}, \tag{3}$$

and

$$\delta_{t+1} \leq \gamma\delta_t + \lambda_t d_t + r_t^{\mathrm{adm}}. \tag{4}$$

Consequently,

$$\delta_{t+1} \leq (\gamma + \lambda_t\alpha)\delta_t + \lambda_t r_t^{\mathrm{dis}} + r_t^{\mathrm{adm}}. \tag{5}$$

Here, α measures the sensitivity of disambiguation error to prior mismatch, γ measures the persistence of admission-induced prior skew, and λ_t measures the influence of disambiguation error on the next-round admitted distribution. The residual r_t^{dis} collects intrinsic candidate overlap, margin approximation error, and partial-log-loss error, while r_t^{adm} collects full-candidate error and admission noise.

Theorem 3.1 **shows that admission bias and disambiguation bias reinforce each other.** Eq. (3) indicates that a biased effective prior π_t can increase d_t by making over-represented classes more competitive within candidate sets. Eq. (4) indicates that wrong disambiguation can distort the confidence statistics used for later admission, thereby affecting δ_{t+1}. Together, they yield the feedback loop

$$\delta_t \;\longrightarrow\; d_t \;\longrightarrow\; \delta_{t+1}, \tag{6}$$

showing that the two biases form a self-reinforcing process rather than two independent errors.

Connection to target risk. The above coupled biases also affect the target risk. Specifically, admission skew changes the effective training distribution Q_t, while disambiguation errors introduce incorrect supervision on both partial-label and full-candidate samples. Therefore, controlling both the admission-induced gap δ_t and the disambiguation error d_t is important for target performance. The formal bound and derivation are provided in Appendix G.6.

4 Proposed Method: AdapMatch

Guided by Section 3, which calls for jointly controlling the distribution gap and disambiguation error, we propose **AdapMatch** to decouple these coupled biases (see Figure. 6 in the Appendix for the overall framework). AdapMatch uses a shared encoder $g(\cdot)$ with two branches: the *Distribution-Aware Selector* (DAS) $f_{\text{das}}(\cdot)$ and the *Calibration-Aware Disambiguator* (CAD) $f_{\text{cad}}(\cdot)$. Appendix B provides the full algorithmic details.

4.1 Preliminaries

Let $\mathcal{D}_p = \{(x_i, S_i)\}_{i=1}^N$ be the partially labeled set and $\mathcal{D}_u = \{x_i\}_{i=1}^M$ be the unlabeled set, where $S_i = \mathcal{Y}$ for $x_i \in \mathcal{D}_u$. In a mini-batch $\mathcal{B}$ of size B, for each branch $m \in \{\text{das}, \text{cad}\}$, let $z_i^m = f_m(g(\omega(x_i)))$ be the logits on weakly augmented input $\omega(x_i)$, and $r_i^m = \text{softmax}(z_i^m)$ be the class

probabilities. To provide ambiguity-consistent supervision, we adopt a candidate-vs-non-candidate formulation. For PLL samples ($S_i \subset \mathcal{Y}$), both branches minimize the partial binary loss:

$$\mathcal{L}_{\text{part}}^{m} = \sum_{i \in \mathcal{B}} \mathbb{I}(S_i \neq \mathcal{Y}) \left(-\log \sum_{j \in S_i} r_{ij}^{m} \right). \tag{7}$$

This loss encourages probability mass to concentrate on the candidate set while excluding full-candidate samples that provide no non-trivial disambiguation constraint.

4.2 Class Distribution Estimation

In SSPLL, the true class distribution is unknown and difficult to estimate accurately due to the limited labeled samples and ambiguous supervision. The observed supervision consists of ambiguous candidate labels and unlabeled data, while the model confidence used for estimation is often biased toward dominant classes and unstable in early training. A naive estimate from raw predictions may therefore amplify head-class contamination and further degrade pseudo-label admission and candidate-label disambiguation. To address this issue, we introduce a *cyclic optimization mechanism*, where the estimated distribution and label confidence iteratively refine each other. This self-correcting process produces more accurate and stable distribution estimates over training. Detailed comparisons of the prediction status are provided in Appendix E.

Concretely, for a given sample x_i, we maintain a label-confidence memory vector $\pi_i = [\pi_{i1}, \pi_{i2}, \ldots, \pi_{iC}] \in \mathbb{R}^C$, initialized as:

$$\pi_{ij} = \begin{cases} \frac{1}{|S_i|}, & \text{if } j \in S_i, \\ 0, & \text{otherwise}, \end{cases} \tag{8}$$

where S_i is the candidate label set of x_i and $|S_i|$ denotes its cardinality. Then, the initial class distribution vector $\boldsymbol{d} = [d_1, d_2, \ldots, d_C] \in \mathbb{R}^C$ is estimated by averaging the confidence memories over all training samples:

$$d_j = \frac{1}{N+M} \sum_{i=1}^{N+M} \pi_{ij}, \quad j = 1, 2, \ldots, C, \tag{9}$$

where N and M denote the number of partially labeled and unlabeled samples, respectively. Since predictions from a single training epoch may not be reliable, we adopt an exponential moving average (EMA) to update the confidence memory:

$$\pi_i = \mu \pi_i + (1 - \mu) r_i^{\text{adj}}, \tag{10}$$

where $\mu \in [0, 1]$ is the EMA decay rate. Instead of directly updating the memory with raw softmax predictions, we use the calibrated confidence $\boldsymbol{r}_i^{\text{adj}}$ defined later in Eq. (14). This design prevents the estimated distribution from being dominated by unstable or head-biased predictions in a single epoch. By softly integrating the calibrated confidence with historical memory, the estimation process becomes progressive and more robust, allowing the class prior and label confidence to gradually stabilize each other over training.

4.3 Distribution-Aware Selector (DAS)

To mitigate **Bias 1 (selection bias)** under unknown class distributions, we propose the Distribution-Aware Selector (DAS). This module dynamically regulates the admission of pseudo-labels to ensure that the effective training distribution aligns with the latent class priors, thereby preventing dominant classes from saturating the admitted set.

Given the estimated class distribution $\boldsymbol{d}$ from Section 4.2, the expected number of samples for class j in a batch of size B is approximated by $n_j = \text{round}(B \cdot d_j)$. Let r_i^{das} denote the predicted class probabilities from the DAS head on the weakly augmented input $\omega(x_i)$. To account for the model's current learning state, we define $\mathcal{M}_j^{n_j}$ as the set of top-n_j samples with the highest prediction scores for class j. The learning status of class j is quantified as: $\tau_j^{\text{das}} = \sum_{i=1}^{B} r_{ij}^{\text{das}} \cdot \mathbb{I}(x_i \in \mathcal{M}_j^{n_j})$. In an ideal scenario, the model would assign a probability of 1 to the ground-truth class, resulting in a target learning status $\tau_j^{\text{das}*} = n_j$. We thus define the *local selection coefficient* (LSC) for class

j as $\lambda_j^{\text{local}} = \tau_j^{\text{das}}/\tau_j^{\text{das}*}$. To further assess the overall learning confidence across the entire label space, we introduce a *global selection coefficient* (GSC): $\lambda^{\text{global}} = \frac{1}{C}\sum_{j=1}^{C} \max_{i\in\{1,\dots,B\}} r_{ij}^{\text{das}}$. By combining LSC and GSC, the adaptive number of high-confidence samples to be admitted for class j is computed as:

$$K_j^{\text{das}} = \left\lceil n_j \cdot \lambda_j^{\text{local}} \cdot \lambda^{\text{global}} \right\rceil, \tag{11}$$

where the ceiling function $\lceil\cdot\rceil$ ensures that even minority classes with lower current confidence receive at least one training sample per batch, effectively reducing the admission-induced distribution gap. As shown in Figure 9 (Appendix), our dynamic K_j^h consistently outperforms any fixed constant, supporting the need for self-adaptive neighbor selection.

For each class j, the top-K_j^{das} samples with the highest predicted probabilities form the admitted set $\mathcal{M}_j^{K_j^{\text{das}}}$. Each sample $x_i \in \mathcal{M}_j^{K_j^{\text{das}}}$ is assigned a hard pseudo-label $\widetilde{\boldsymbol{y}}^j$, where $\widetilde{y}_j^j = 1$ and 0 otherwise. If a sample is selected by multiple classes, it is assigned to the class with the highest score to maintain label uniqueness. DAS enforces consistency regularization by aligning the predictions of strongly augmented versions $\Omega(x_i)$ with these pseudo-labels. The consistency loss is defined as:

$$\mathcal{L}_{\text{cons}}^{\text{das}} = \frac{1}{K_{\text{total}}^{\text{das}}} \sum_{i=1}^{B}\sum_{j=1}^{C} \mathbb{I}(x_i \in \mathcal{M}_j^{K_j^{\text{das}}}) \cdot \mathcal{H}(r_i^{\text{das},s}, \widetilde{\boldsymbol{y}}^j), \tag{12}$$

where $r_i^{\text{das},s} = \text{softmax}(f_{\text{das}}(g(\Omega(x_i))))$ and $\mathcal{H}(\cdot,\cdot)$ denotes the cross-entropy loss.

To enhance robustness against residual pseudo-label noise within the admitted set $\mathcal{M}^{\text{das}}$, we apply MixUp [21]. For samples $x_i, x_k \in \mathcal{M}^{\text{das}}$ with pseudo-labels $\widetilde{\boldsymbol{y}}_i, \widetilde{\boldsymbol{y}}_k$, we construct:$x_i^m = \phi x_i + (1-\phi)x_k, \quad \widetilde{\boldsymbol{y}}_i^m = \phi\widetilde{\boldsymbol{y}}_i + (1-\phi)\widetilde{\boldsymbol{y}}_k$, where $\phi \sim \text{Beta}(\alpha,\alpha)$. The MixUp loss is computed as: $\mathcal{L}_{\text{mix}}^{\text{das}} = \frac{1}{|\mathcal{M}^{\text{das}}|}\sum_{i=1}^{|\mathcal{M}^{\text{das}}|} \mathcal{H}\left(r_i^{\text{das},m}, \widetilde{\boldsymbol{y}}_i^m\right)$, where $r_i^{\text{das},m}$ is the DAS branch prediction on the weakly augmented MixUp sample $\omega(x_i^m)$. The total DAS loss is:

$$\mathcal{L}_{\text{das}} = \mathcal{L}_{\text{partial}}^{\text{das}} + \beta\left(\mathcal{L}_{\text{cons}}^{\text{das}} + \mathcal{L}_{\text{mix}}^{\text{das}}\right), \tag{13}$$

with the warm-up schedule $\beta = \min(T_{\text{now}}/T_w, 1)$, where T_{now} and T_w represent the current epoch and the warm-up period, respectively. This schedule assigns a small weight to the consistency losses at early stages, when pseudo-labels and branch predictions are still unreliable, and gradually increases their influence as the model becomes more stable.

4.4 Calibration-Aware Disambiguator (CAD)

To mitigate **Bias 2 (Disambiguation Bias)**, we focus on the biased competition within each candidate label set. In SSPLL, a frequent class may appear as a false-positive candidate label and receive overconfident predictions, causing the true label of a minority sample to be suppressed during disambiguation. CAD is designed to reduce this candidate-level domination by using the estimated prior as a calibration signal. This not only improves candidate-label disambiguation, but also closes the feedback loop of class distribution estimation: the calibrated confidence is written back to the memory in Eq. (10), providing a less head-contaminated signal for the next-round prior estimate.

Let $z_i^{\text{cad}} = f_{\text{cad}}(g(\omega(x_i)))$ be the logits from the CAD branch. For each sample, candidate labels only compete within S_i, where a frequent false-positive label can dominate the true label due to prior-induced confidence bias. We therefore use the estimated prior $\boldsymbol{d}$ as a calibration signal to compute the candidate-level confidence:

$$r_{ij}^{\text{adj}} = \mathbb{I}(j \in S_i) \cdot \frac{\exp(z_{ij}^{\text{cad}} - t\log d_j)}{\sum_{k\in S_i}\exp(z_{ik}^{\text{cad}} - t\log d_k)}, \tag{14}$$

where $t > 0$ is a temperature parameter. By penalizing classes with higher estimated frequencies, Eq. (14) reduces the contamination factor d_t defined in Section 3.

To maintain a balanced learning status across all classes, we define the target number of samples per class in a batch as $m = \text{round}(B/C)$. The ideal learning status is $\tau^{\text{cad}*} = B$, while the current status is measured as $\tau^{\text{cad}} = \sum_{i,j} r_{ij}^{\text{adj}} \cdot \mathbb{I}(x_i \in \mathcal{M}_j^m)$, where $\mathcal{M}_j^m$ contains the top-m samples for class j. The number of selected samples per class is then:

$$K^{\text{cad}} = \lceil m \cdot (\tau^{\text{cad}}/\tau^{\text{cad}*})\rceil. \tag{15}$$

Table 1: Accuracy (%, mean ± std) comparisons on CIFAR-10 with $\rho = 10\%$ under different γ and q settings.

Method	CIFAR-10 ($\gamma = 20$)				CIFAR-10 ($\gamma = 10$)			
	$q = 0.1$	$q = 0.3$	$q = 0.5$	$q = 0.7$	$q = 0.1$	$q = 0.3$	$q = 0.5$	$q = 0.7$
PiCO	51.35 ± 0.84	40.82 ± 0.34	31.01 ± 0.95	21.79 ± 4.39	56.62 ± 0.16	47.88 ± 0.60	34.23 ± 0.88	24.30 ± 1.05
PLCR	48.71 ± 0.90	35.68 ± 1.39	28.00 ± 0.35	24.20 ± 0.37	62.30 ± 2.77	41.88 ± 1.39	28.56 ± 0.40	25.37 ± 1.60
SoLar	57.81 ± 0.52	45.05 ± 1.46	36.14 ± 0.37	31.44 ± 1.29	67.65 ± 0.58	55.85 ± 0.94	43.22 ± 0.95	29.16 ± 1.43
SoLar*	77.67 ± 2.13	51.66 ± 4.17	28.95 ± 1.68	17.12 ± 2.40	79.62 ± 3.94	62.75 ± 3.27	27.35 ± 2.33	24.82 ± 2.05
FixMatch+	73.16 ± 1.70	56.05 ± 1.65	39.90 ± 0.28	31.30 ± 1.36	82.76 ± 0.53	73.42 ± 1.00	44.64 ± 1.56	32.60 ± 0.44
FreeMatch+	70.75 ± 0.47	54.94 ± 3.99	45.52 ± 3.90	39.21 ± 0.98	82.74 ± 0.23	75.24 ± 1.25	55.17 ± 1.15	39.01 ± 1.69
ConCont	75.73 ± 0.34	$\underline{68.75 \pm 3.68}$	49.43 ± 2.21	$\underline{42.11 \pm 0.90}$	82.15 ± 0.41	80.25 ± 0.68	$\underline{71.42 \pm 0.53}$	41.15 ± 0.44
FairMatch	77.50 ± 0.51	66.79 ± 1.50	47.85 ± 1.67	33.92 ± 1.95	$\underline{83.07 \pm 0.20}$	77.15 ± 0.48	68.53 ± 1.54	$\underline{43.74 \pm 2.23}$
CPE	$\underline{79.94 \pm 0.62}$	68.74 ± 1.27	$\underline{60.18 \pm 1.51}$	35.66 ± 1.27	81.05 ± 0.55	$\underline{80.71 \pm 0.74}$	71.28 ± 1.10	29.39 ± 1.37
SimPro	74.96 ± 0.80	61.32 ± 1.10	58.32 ± 2.50	31.04 ± 2.30	74.20 ± 0.60	71.52 ± 1.30	62.83 ± 2.80	27.21 ± 2.60
AdapMatch	$\mathbf{85.30 \pm 0.29}$	$\mathbf{78.42 \pm 1.76}$	$\mathbf{72.78 \pm 0.73}$	$\mathbf{69.40 \pm 0.56}$	$\mathbf{89.53 \pm 0.19}$	$\mathbf{88.70 \pm 0.12}$	$\mathbf{87.19 \pm 0.02}$	$\mathbf{82.46 \pm 1.02}$

The top-K^{cad} samples for each class form the admitted set $\mathcal{M}^{\text{cad}}$ with hard pseudo-labels $\tilde{\boldsymbol{y}}$. Following the implementation logic in Eqs. (12), we apply consistency regularization $\mathcal{L}_{\text{cons}}^{\text{cad}}$ and MixUp-based robust training $\mathcal{L}_{\text{mix}}^{\text{cad}}$ to the set $\mathcal{M}^{\text{cad}}$. The total CAD loss is defined as:

$$\mathcal{L}\text{cad} = \mathcal{L}\text{part}^{\text{cad}} + \beta \left(\mathcal{L}\text{cons}^{\text{cad}} + \mathcal{L}\text{mix}^{\text{cad}} \right), \tag{16}$$

where the definition and purpose of the warm-up schedule β are identical to those in Eq. (13).

4.5 Joint Optimization and Inference

AdapMatch optimizes the joint objective: $\mathcal{L} = \mathcal{L}_{\text{das}} + \mathcal{L}_{\text{cad}}$. Following [22, 23], we assign equal weights to both modules, ensuring balanced importance for distribution-aware selection and calibration-aware disambiguation.

For a test sample $\boldsymbol{x}$, we fuse the probability vectors $\hat{\boldsymbol{y}}^{\text{das}}$ and $\hat{\boldsymbol{y}}^{\text{cad}}$ using an adaptive entropy-based mechanism. Let $e^m = -\sum_{j=1}^{C} \hat{y}_j^m \log \hat{y}_j^m$ denote the Shannon entropy for branch $m \in \{\text{das}, \text{cad}\}$. To prioritize the more confident prediction, fusion weights are assigned based on the reciprocal of entropy: $w^m = (1/e^m)/(1/e^{\text{das}} + 1/e^{\text{cad}})$. The final prediction is:

$$\hat{\boldsymbol{y}}^{\text{test}} = w^{\text{das}} \hat{\boldsymbol{y}}^{\text{das}} + w^{\text{cad}} \hat{\boldsymbol{y}}^{\text{cad}}, \tag{17}$$

with the predicted label $\arg\max_j \hat{y}_j^{\text{test}}$. This dynamic weighting leverages the complementary strengths of both modules at test time.

5 EXPERIMENTS

5.1 Experimental Setup

Datasets and implementation. We evaluate AdapMatch on five benchmarks: CIFAR-10/100/100H [24], STL-10 [25], and downsampled ImageNet-127 [26]. We construct long-tailed SSPLL tasks by varying the partial-label ratio ρ, ambiguity rate $q = P(\bar{y}_i \in S_i \mid \bar{y}_i \neq y_i)$, and imbalance factor $\gamma = N_{\max}/N_{\min}$, where class sizes follow an exponential long-tailed distribution. **Partially labeled and unlabeled sets share the same imbalanced distribution**, and all methods are evaluated on the original balanced test sets. We use ResNet-18 as the backbone and report the mean and standard deviation over three independent runs. Detailed dataset configurations and implementation details are provided in Appendix C.

Baselines. To evaluate performance, we compare representative **PLL** methods PiCO [11], PLCR [13] and **SSPLL** methods FixMatch+ [27], FreeMatch+ [28], ConCont [15], FairMatch [16], where "+" indicates the addition of candidate-label disambiguation. To better address the long-tailed setting, we further include **LTPLL**-specific methods such as SoLar [19] and its semi-supervised variant SoLar*, which incorporate class priors via optimal transport. We also consider **LTSSL** baselines, including SimPro [29] and CPE [30]. All models are trained with the recommended hyperparameters to ensure reproducibility.

Table 2: Accuracy (%, mean ± std) comparisons on CIFAR-100 with $\rho = 20\%$ under different γ and q settings.

Method	CIFAR-100 ($\gamma = 10$)				CIFAR-100 ($\gamma = 5$)			
	$q = 0.05$	$q = 0.1$	$q = 0.15$	$q = 0.2$	$q = 0.05$	$q = 0.1$	$q = 0.15$	$q = 0.2$
PiCO	20.64 ± 0.60	13.12 ± 0.32	6.86 ± 0.61	6.45 ± 0.21	25.65 ± 1.65	15.08 ± 1.16	8.10 ± 0.38	4.85 ± 0.16
PLCR	33.12 ± 0.94	21.29 ± 0.52	16.42 ± 0.30	11.97 ± 0.05	39.01 ± 0.32	28.15 ± 0.51	19.18 ± 0.07	14.22 ± 0.96
SoLar	29.28 ± 1.56	19.74 ± 0.82	13.44 ± 0.48	8.94 ± 0.70	35.21 ± 0.44	23.62 ± 0.18	14.93 ± 0.41	12.41 ± 0.15
SoLar*	31.85 ± 4.25	18.49 ± 0.54	10.85 ± 0.45	7.89 ± 0.23	39.85 ± 2.58	25.64 ± 0.10	17.08 ± 1.27	10.80 ± 1.62
FixMatch+	27.87 ± 2.02	18.30 ± 0.99	10.41 ± 0.16	9.09 ± 0.89	36.50 ± 1.59	22.59 ± 0.41	13.57 ± 0.68	9.36 ± 0.41
FreeMatch+	28.45 ± 0.37	21.32 ± 0.59	17.98 ± 0.23	10.17 ± 1.87	34.22 ± 0.82	28.24 ± 0.37	18.99 ± 2.71	11.10 ± 1.81
ConCont	39.91 ± 0.51	33.66 ± 0.65	27.75 ± 0.50	$\mathbf{24.31 \pm 0.66}$	47.92 ± 0.21	40.90 ± 0.33	33.73 ± 1.23	28.33 ± 4.03
FairMatch	37.98 ± 0.60	29.28 ± 0.97	21.17 ± 0.78	16.44 ± 0.17	47.25 ± 0.86	37.73 ± 0.98	28.05 ± 1.64	21.44 ± 0.54
CPE	31.68 ± 0.91	19.28 ± 0.83	18.52 ± 0.91	17.81 ± 0.73	31.13 ± 0.80	33.07 ± 0.60	20.93 ± 0.80	18.08 ± 0.60
SimPro	28.79 ± 0.76	16.61 ± 0.80	15.24 ± 0.94	13.86 ± 0.89	27.14 ± 1.21	24.31 ± 1.16	18.31 ± 2.17	15.87 ± 1.02
AdapMatch	$\mathbf{46.07 \pm 0.90}$	$\mathbf{37.26 \pm 0.47}$	$\mathbf{29.83 \pm 0.56}$	23.26 ± 0.07	$\mathbf{55.05 \pm 0.49}$	$\mathbf{45.83 \pm 0.34}$	$\mathbf{38.15 \pm 0.96}$	$\mathbf{31.20 \pm 1.21}$

5.2 Main Results

Tables 1 and 2 present the main results on CIFAR-10/100, while Table 3 reports results on ImageNet-127 and STL-10. Figure 5 illustrates the training dynamics of DAS/CAD branches and disambiguation accuracy. Additional results, including CIFAR-100H, balanced settings, head/medium/tail breakdowns, DFS stability analysis, class-wise confidence-density diagnostics, and hyper-parameter sensitivity analysis, are deferred to Appendix. We summarize the key observations as follows:

AdapMatch demonstrates universal robustness across five benchmarks. Regardless of the imbalance factor γ or label ambiguity q, our method consistently delivers substantial absolute gains over state-of-the-art baselines: up to **41.31%** on CIFAR-10, **17.81%** on STL-10, **10.78%** on CIFAR-100H, **7.13%** on CIFAR-100, and **4.80%** on ImageNet-127. These results validate the framework's efficacy across diverse data scales and experimental settings.

Table 3: Accuracy (%) comparisons on Small-ImageNet-127 and STL-10.

Algorithm	ImageNet-127		STL-10	
	32×32	64×64	$q = 0.3$	$q = 0.5$
CPE	47.8	58.2	72.61	63.79
SimPro	44.6	53.8	64.31	54.14
AdapMatch	**52.6**	**63.0**	**82.20**	**81.60**

AdapMatch exhibits strong robustness on highly ambiguous tasks. While most baselines degrade significantly as q increases, AdapMatch maintains competitive performance. Even on CIFAR-100 under the most challenging setting ($\gamma = 10$, $\rho = 20\%$, $q = 0.2$), it only slightly underperforms ConCont by 1%, but outperforms others by over 4%, with some margins exceeding 10%.

AdapMatch provides reliable class distribution estimates. Fig. 3 evaluates distribution estimation on CIFAR-10 under severe imbalance ($\gamma = 100$, $\rho = 20\%$, $q = 0.2$) using the Distribution Fit Score (DFS), which measures the global alignment between the estimated and true class proportions. AdapMatch consistently achieves the highest DFS, increasing from 77.3 at Epoch 50 to 87.7 at the final stage, indicating that its estimated prior is more accurate and becomes progressively better aligned with the true distribution during training. In contrast, FairMatch remains substantially lower, while ConCont improves mainly at the late stage. These results suggest that AdapMatch provides a more reliable latent prior for pseudo-label admission and candidate-level disambiguation. More detailed comparisons under both balanced and long-tailed settings are provided in Appendix E.

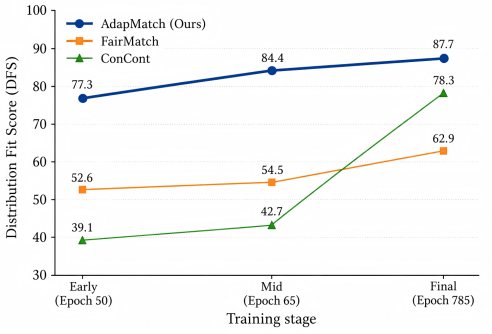


Figure 3: Distribution estimation quality under severe imbalance ($\gamma = 100$) on CIFAR-10. Early, mid, and final correspond to epochs 50, 65, and 785, respectively.

AdapMatch effectively mitigates class imbalance and excels in tail classes modeling. Fig. 4 shows the head–medium–tail accuracy comparison on CIFAR-10 and CIFAR-100. Compared with ConCont and CPE, AdapMatch achieves much stronger medium- and tail-class performance, while main-

taining competitive head-class accuracy. This suggests that AdapMatch avoids over-concentrating on dominant classes and provides more balanced class-group modeling under imbalance. More comprehensive results across datasets and partial-label settings are reported in Table 9 (Appendix).

Table 4: Classification accuracy (%) of different variants of AdapMatch.

Method	CIFAR-10 ($\gamma = 20, \rho = 10\%, q = 0.7$)				CIFAR-100 ($\gamma = 5, \rho = 20\%, q = 0.15$)				CIFAR-100H ($\gamma = 5, \rho = 20\%, q = 0.7$)			
	Overall	Head	Medium	Tail	Overall	Head	Medium	Tail	Overall	Head	Medium	Tail
AdapMatch	**68.80**	82.03	60.37	66.80	**36.97**	46.15	41.74	22.88	**55.88**	60.45	55.26	51.94
DAS Pred	66.24	85.77	52.50	65.03	36.39	49.94	40.06	19.06	55.03	65.68	52.98	43.21
CAD Pred	65.89	65.73	62.08	71.13	35.55	42.30	40.50	23.70	53.73	51.94	53.40	58.03
Equal Weight	60.64	85.70	59.90	31.10	35.04	44.82	34.82	25.48	53.98	60.45	53.82	47.67
w/o $\mathcal{L}_{\text{das}}$	58.02	91.90	55.55	27.43	27.57	38.00	34.88	9.61	41.59	49.64	45.13	21.62
w/o $\mathcal{L}_{\text{cad}}$	49.66	54.40	50.58	43.70	35.05	37.64	42.15	25.15	52.93	49.02	54.96	59.98
w/o LA	61.46	82.50	48.57	57.60	36.59	45.39	37.53	22.30	55.17	59.39	49.67	46.07
w/o GCE	67.47	73.10	67.62	61.63	35.17	45.39	39.85	20.12	53.02	54.99	52.86	46.61
w/o MixUp	51.18	88.63	53.12	11.13	27.51	37.24	30.76	14.42	41.56	49.87	40.43	13.71
w/o Binary	41.86	45.73	48.43	29.23	35.62	41.06	40.97	24.67	53.95	51.73	53.77	57.61

AdapMatch maintains robust and balanced performance across imbalance levels via collaborative bias decoupling. Some baselines (e.g., SoLar, FreeMatch+, ConCont) show a counterintuitive trend where accuracy at γ=10 is lower than at γ=20, suggesting that performance under stronger imbalance can be inflated by head-class dominance while minority collapse is masked. When the distribution becomes less skewed, more minority samples reveal unresolved disambiguation bias, causing overall degradation. In contrast, AdapMatch remains consistently strong across different γ, demonstrating the benefit of jointly controlling admission and disambiguation biases.

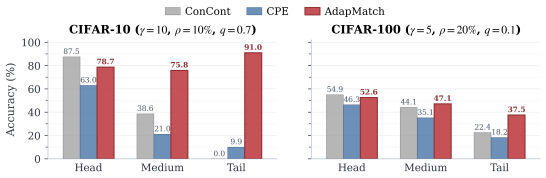


Figure 4: Head–medium–tail accuracy comparison on CIFAR-10 and CIFAR-100.

AdapMatch improves both admission and disambiguation reliability. As shown in Figure 5, the accuracy metrics of the DAS and CAD branches steadily increase and gradually converge during training. This dynamic trend indicates that pseudo-label admission becomes more reliable and candidate-level mis-disambiguation is progressively reduced. In terms of our theory, the improved DAS accuracy reflects a more reliable admitted set, which helps alleviate the admission-induced distribution gap δ_t; the improved CAD accuracy corresponds to a decrease in the partial-label disambiguation error d_t, especially by reducing dominant-class false positives within candidate sets. Since Theorem 3.1 shows the feedback path $\delta_t \to d_t \to \delta_{t+1}$, the simultaneous improvement of both branches suggests that AdapMatch progressively weakens this self-reinforcing loop during training.

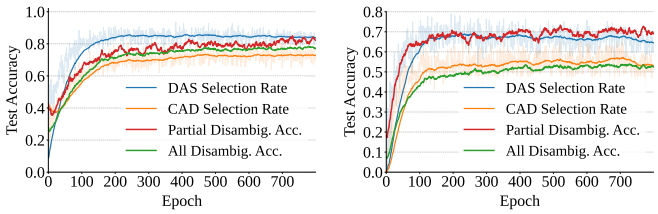


Figure 5: Training dynamics on CIFAR-10 and CIFAR-100 under long-tailed SSPLL settings.

5.3 Further Analysis

Ablation Analysis. Table 4 reports ablations of AdapMatch. The full model achieves the best performance on all benchmarks, while removing any component leads to a clear drop. Using only one branch (DAS Pred or CAD Pred) or naive Equal Weight fusion cannot jointly handle selection and disambiguation biases; in contrast, our entropy-based fusion (Eq. 17) better leverages branch complementarity. Removing $\mathcal{L}_{\text{das}}$, $\mathcal{L}_{\text{cad}}$, or logit adjustment (**w/o LA**) notably hurts performance, especially on minority classes, confirming their roles in bias decoupling and calibration. Disabling GSC, MixUp, or binary partial-label supervision (**w/o Binary**) also degrades accuracy, showing the benefit of robustness regularization and candidate-set constraints.

6 Conclusion

This paper investigates SSPLL under unknown class distributions and theoretically identifies selection bias and disambiguation bias as two coupled failure modes. We show that the admission-induced distribution gap and the partial-label disambiguation error form a self-reinforcing loop, which motivates their joint control. Based on this insight, we propose AdapMatch, which combines a Distribution-Aware Selector (DAS) for prior-aware pseudo-label admission and a Calibration-Aware Disambiguator (CAD) for reducing dominant-class false positives within candidate sets. Experiments on multiple benchmarks demonstrate that AdapMatch achieves state-of-the-art performance and remains robust under severe imbalance and high label ambiguity.

Acknowledgements and Disclosures

This work was supported in part by the National Natural Science Foundation of China under Grants U24A20322, 62576094 and 62422118, in part by the Hong Kong UGC under grants UGC/FDS11/E03/25 and UGC/FDS11/E04/26, and in part by the Hong Kong Research Grants Council under Grants RFS2627-1S02, R1018-25F. This research work was also supported by the Big Data Computing Center of Southeast University. The authors declare no competing interests.

References

[1] Timothée Cour, Benjamin Sapp, and Ben Taskar. Learning from partial labels. *J. Mach. Learn. Res.*, 12:1501–1536, 2011. doi: 10.5555/1953048.2021049. URL `https://dl.acm.org/doi/10.5555/1953048.2021049`.

[2] Zinan Zeng, Shijie Xiao, Kui Jia, Tsung-Han Chan, Shenghua Gao, Dong Xu, and Yi Ma. Learning by associating ambiguously labeled images. In *Proceedings of the IEEE Conference on computer vision and pattern recognition*, pages 708–715, 2013.

[3] Jie Luo and Francesco Orabona. Learning from candidate labeling sets. *Advances in neural information processing systems*, 23, 2010.

[4] Gabriel Panis and Andreas Lanitis. An overview of research activities in facial age estimation using the fg-net aging database. In *European Conference on Computer Vision*, pages 737–750. Springer, 2014.

[5] Jiaqi Lv, Miao Xu, Lei Feng, Gang Niu, Xin Geng, and Masashi Sugiyama. Progressive identification of true labels for partial-label learning. In *international conference on machine learning*, pages 6500–6510. PMLR, 2020.

[6] Ning Xu, Congyu Qiao, Xin Geng, and Min-Ling Zhang. Instance-dependent partial label learning. *Advances in Neural Information Processing Systems*, 34:27119–27130, 2021.

[7] Feng Hong, Jiangchao Yao, Zhihan Zhou, Ya Zhang, and Yanfeng Wang. Long-tailed partial label learning via dynamic rebalancing. *arXiv preprint arXiv:2302.05080*, 2023.

[8] Shuai Zhang, Lina Yao, Aixin Sun, and Yi Tay. Deep learning based recommender system: A survey and new perspectives. *ACM Computing Surveys*, 52(1):1–38, February 2019. ISSN 1557-7341. doi: 10.1145/3285029. URL `http://dx.doi.org/10.1145/3285029`.

[9] Chenglin Gong, Tongliang Liu, Yipeng Tang, and Dacheng Tao. A regularization approach for instance-based superset label learning. *IEEE Transactions on Cybernetics*, 48(3):967–978, 2017.

[10] Nam Nguyen and Rich Caruana. Classification with partial labels. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '08, page 551–559, New York, NY, USA, 2008. Association for Computing Machinery. ISBN 9781605581934. doi: 10.1145/1401890.1401958. URL `https://doi.org/10.1145/1401890.1401958`.

[11] Haobo Wang, Ruixuan Xiao, Yixuan Li, Lei Feng, Gang Niu, Gang Chen, and Junbo Zhao. Pico+: Contrastive label disambiguation for robust partial label learning, 2022. URL https://arxiv.org/abs/2201.08984.

[12] Shiyu Xia, Jiaqi Lv, Ning Xu, and Xin Geng. Ambiguity-induced contrastive learning for instance-dependent partial label learning. In *IJCAI*, pages 3615–3621, 2022.

[13] Dong-Dong Wu, Deng-Bao Wang, and Min-Ling Zhang. Revisiting consistency regularization for deep partial label learning. In *International conference on machine learning*, pages 24212–24225. PMLR, 2022.

[14] Qian-Wei Wang, Yufeng Li, Zhi-Hua Zhou, et al. Partial label learning with unlabeled data. In *IJCAI*, pages 3755–3761, 2019.

[15] Qian-Wei Wang, Bowen Zhao, Mingyan Zhu, Tianxiang Li, Zimo Liu, and Shu-Tao Xia. Controller-guided partial label consistency regularization with unlabeled data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 15571–15579, 2024.

[16] Jiahao Jiang, Yuheng Jia, Hui Liu, and Junhui Hou. Fairmatch: Promoting partial label learning by unlabeled samples. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '24, page 1269–1278, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400704901. doi: 10.1145/3637528.3671685. URL https://doi.org/10.1145/3637528.3671685.

[17] Jing Wang and Min-Ling Zhang. Towards mitigating the class-imbalance problem for partial label learning. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 2427–2436, 2018.

[18] Wenpeng Liu, Li Wang, Jie Chen, Yu Zhou, Ruirui Zheng, and Jianjun He. A partial label metric learning algorithm for class imbalanced data. In *Asian Conference on Machine Learning*, pages 1413–1428. PMLR, 2021.

[19] Haobo Wang, Mingxuan Xia, Yixuan Li, Yuren Mao, Lei Feng, Gang Chen, and Junbo Zhao. Solar: Sinkhorn label refinery for imbalanced partial-label learning. *Advances in neural information processing systems*, 35:8104–8117, 2022.

[20] Yuheng Jia, Xiaorui Peng, Ran Wang, and Min-Ling Zhang. Long-tailed partial label learning by head classifier and tail classifier cooperation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 12857–12865, 2024.

[21] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.

[22] Hyuck Lee, Seungjae Shin, and Heeyoung Kim. Abc: Auxiliary balanced classifier for class-imbalanced semi-supervised learning. *Advances in Neural Information Processing Systems*, 34: 7082–7094, 2021.

[23] Tong Wei and Kai Gan. Towards realistic long-tailed semi-supervised learning: Consistency is all you need. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3469–3478, 2023.

[24] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[25] Adam Coates, Andrew Y. Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In Geoffrey J. Gordon, David B. Dunson, and Miroslav Dudík, editors, *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2011, Fort Lauderdale, USA, April 11-13, 2011*, volume 15 of *JMLR Proceedings*, pages 215–223. JMLR.org, 2011. URL http://proceedings.mlr.press/v15/coates11a/coates11a.pdf.

[26] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), 20-25 June 2009, Miami, Florida, USA*, pages 248–255. IEEE Computer Society, 2009. doi: 10.1109/CVPR.2009.5206848. URL `https://doi.org/10.1109/CVPR.2009.5206848`.

[27] 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. *Advances in neural information processing systems*, 33:596–608, 2020.

[28] Yidong Wang, Hao Chen, Qiang Heng, Wenxin Hou, Yue Fan, Zhen Wu, Jindong Wang, Marios Savvides, Takahiro Shinozaki, Bhiksha Raj, et al. Freematch: Self-adaptive thresholding for semi-supervised learning. *arXiv preprint arXiv:2205.07246*, 2022.

[29] Chaoqun Du, Yizeng Han, and Gao Huang. Simpro: A simple probabilistic framework towards realistic long-tailed semi-supervised learning. *arXiv preprint arXiv:2402.13505*, 2024.

[30] Chengcheng Ma, Ismail Elezi, Jiankang Deng, Weiming Dong, and Changsheng Xu. Three heads are better than one: Complementary experts for long-tailed semi-supervised learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 14229–14237, 2024.

[31] 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*, pages 702–703, 2020.

A framework of AdapMatch

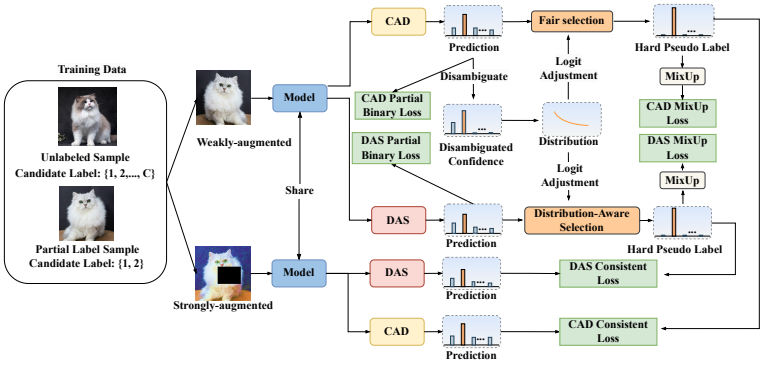


Figure 6: Illustration of the AdapMatch framework. Unlabeled samples are treated as partial-label samples with the full class set as candidates. All partial-label samples, including unlabeled ones, participate in high-confidence sample selection and consistency regularization. AdapMatch employs a collaborative dual-branch design: the Distribution-Aware Selector (DAS) utilizes a distribution-aware selection strategy to align pseudo-label admission with latent class priors, while the Calibration-Aware Disambiguator (CAD) applies a fair selection strategy combined with logit adjustment to suppress dominant-class overconfidence and mitigate disambiguation bias. Both modules incorporate MixUp augmentation to prevent overfitting and enhance the model's overall generalization capability.

B Algorithm

We present the complete training procedure of AdapMatch in Algorithm 1.

Algorithm 1 Training Procedure of AdapMatch

Input: Training dataset D, batch size B, maximum training epochs $T_{\max}$, warm-up epochs T_w, Beta distribution parameter α, EMA coefficient μ, logit calibration temperature t
Output: Model parameters
Initialize sample-wise label confidence π_i using candidate sets
Initialize class distribution $\boldsymbol{d}$ by averaging confidence over all samples
for $T_{\text{now}} = 1$ to $T_{\max}$ **do**
  for $b = 1$ to $\lfloor |D|/B \rfloor$ **do**
    Sample a mini-batch $\mathcal{B}$ from D
    Compute DAS partial-label loss $\mathcal{L}_{\text{partial}}^{\text{das}}$
    Compute adaptive admission counts K_j^{das} (distribution-aware selection)
    Compute DAS consistency loss $\mathcal{L}_{\text{cons}}^{\text{das}}$
    Compute DAS MixUp loss $\mathcal{L}_{\text{mix}}^{\text{das}}$
    Aggregate DAS loss $\mathcal{L}_{\text{das}}$
    Compute CAD partial-label loss $\mathcal{L}_{\text{partial}}^{\text{cad}}$
    Compute calibrated probabilities within candidate sets
    Compute adaptive selection size K^{cad}
    Compute CAD consistency loss $\mathcal{L}_{\text{cons}}^{\text{cad}}$
    Compute CAD MixUp loss $\mathcal{L}_{\text{mix}}^{\text{cad}}$
    Aggregate CAD loss $\mathcal{L}_{\text{cad}}$
    Update confidence memory π_i via EMA using calibrated predictions
    Update class distribution $\boldsymbol{d}$ by averaging updated confidence
    Optimize total loss $\mathcal{L} = \mathcal{L}_{\text{das}} + \mathcal{L}_{\text{cad}}$
  end for
end for
return model parameters

C Datasets and implementation Details

Table 5: Dataset-specific configurations for SSPLL experiments.

Dataset	γ	ρ	q	Notes
CIFAR-10	$\{1, 10, 20\}$	10%	$\{0.1, 0.3, 0.5, 0.7\}$	balanced test set
STL-10	10	10%	$\{0.3, 0.5\}$	balanced test set
CIFAR-100	$\{1, 5, 10\}$	20%	$\{0.05, 0.1, 0.15, 0.2\}$	balanced test set
CIFAR-100H	$\{1, 5, 10\}$	20%	$\{0.3, 0.5, 0.7, 0.9\}$	human-level ambiguity
ImageNet-127	–	10%	–	32×32 and 64×64 downsampled

Optimization and Hyperparameters. We optimize the model using Stochastic Gradient Descent (SGD) with a Nesterov momentum of 0.9. The initial learning rate is set to 0.01, which is further reduced by a factor of 10 at epoch 700. We apply a weight decay of 1×10^{-4} to prevent overfitting. Experiments used 6 NVIDIA RTX 3090 GPUs and 8 NVIDIA RTX 4090 GPUs, with detailed hyperparameter configurations for **AdapMatch** summarized in Table 6.

Table 6: Hyperparameter settings for AdapMatch.

Hyperparameter	**Value**
Batch size (B)	256
Total epochs ($T_{\max}$)	800
Warm-up epochs (T_w)	50
Initial learning rate	0.01
Momentum	0.9
Weight decay	1×10^{-4}
EMA update coefficient (μ)	0.6
MixUp parameter (α)	4
Logit adjustment temperature (t)	2
Data Augmentation	Weak (Flip/Crop) & Strong (RandAug)

Data Augmentation. Following the protocol in FairMatch, we employ a dual-augmentation strategy. Weak augmentation includes random horizontal flips and random crops with padding. Strong augmentation follows the RandAugment [31] policy, which is essential for generating reliable pseudo-labels in the DAS and CAD branches.

D More Experimental Results

Table 7 reports results on CIFAR-100H under $\gamma \in \{5, 10\}$ and $q \in [0.3, 0.9]$. **AdapMatch** achieves the best accuracy in all settings, typically surpassing FairMatch by over 10% absolute. As ambiguity increases to q=0.9, most baselines collapse, whereas AdapMatch remains strong (e.g., 37.14% vs. 26.36% at γ=5), and stays consistently superior under both moderate and severe imbalance. Overall, these results confirm that jointly addressing selection and disambiguation biases enables robust learning from highly noisy partial labels.

As shown in Table. 8, **AdapMatch** consistently achieves the best performance across all 12 settings (3 datasets $\times$ 4 partial label ambiguity rate q) under class-balanced distributions. This demonstrates that our method effectively adapts to unknown data distributions, regardless of whether the label distribution is balanced or long-tailed.

As shown in Table 9, **AdapMatch** consistently achieves state-of-the-art results across all benchmarks, outperforming PLL/SSL/SSPLL baselines by a large margin. On CIFAR-10 (γ=10, q=0.7), it reaches **81.21%** overall accuracy, nearly **40%** higher than the strongest competitor (ConCont). While several baselines (e.g., PLCR, FixMatch+, ConCont) collapse on **minority classes** (0.00% on CIFAR-10), **AdapMatch** remains balanced, achieving **90.97%** on CIFAR-10 minority classes and over **51%** on CIFAR-100H minority classes at q=0.7. These results verify that decoupling selection and disambiguation biases effectively mitigates dominant-class overconfidence and improves minority-class learning under severe ambiguity and imbalance.

Table 7: Accuracy (%) comparisons on CIFAR-100H with $\rho = 20\%$ under different γ and q settings.

Method	CIFAR-100H ($\gamma = 10$)				CIFAR-100H ($\gamma = 5$)			
	$q = 0.3$	$q = 0.5$	$q = 0.7$	$q = 0.9$	$q = 0.3$	$q = 0.5$	$q = 0.7$	$q = 0.9$
PiCO	29.12 ± 0.75	23.50 ± 0.61	16.86 ± 0.48	10.67 ± 0.36	31.71 ± 0.76	25.87 ± 0.60	16.90 ± 0.44	13.13 ± 0.36
PLCR	41.12 ± 0.88	35.57 ± 0.70	29.65 ± 0.65	18.07 ± 0.50	46.09 ± 0.98	41.34 ± 0.81	33.78 ± 0.72	21.60 ± 0.55
SoLar	34.21 ± 0.91	30.29 ± 0.77	23.30 ± 0.58	14.85 ± 0.45	40.86 ± 0.92	35.32 ± 0.79	26.82 ± 0.63	16.02 ± 0.46
SoLar*	36.42 ± 0.85	32.10 ± 0.68	21.25 ± 0.54	13.38 ± 0.41	44.37 ± 0.88	34.61 ± 0.74	25.08 ± 0.58	17.30 ± 0.42
FixMatch+	37.77 ± 0.79	36.28 ± 0.73	26.80 ± 0.61	17.95 ± 0.47	46.69 ± 0.83	44.07 ± 0.76	36.43 ± 0.67	20.53 ± 0.50
FreeMatch+	43.68 ± 0.93	24.10 ± 0.65	15.27 ± 0.42	14.08 ± 0.39	50.91 ± 0.97	46.44 ± 0.89	18.99 ± 0.41	16.73 ± 0.37
ConCont	42.24 ± 0.81	40.78 ± 0.73	34.40 ± 0.69	20.60 ± 0.56	51.02 ± 0.23	48.94 ± 0.27	42.23 ± 0.14	24.06 ± 0.31
FairMatch	46.37 ± 1.02	43.53 ± 0.91	34.98 ± 0.77	22.48 ± 0.62	55.88 ± 1.10	49.90 ± 0.96	45.37 ± 0.91	26.36 ± 0.68
CPE	38.59 ± 0.88	31.86 ± 0.93	30.54 ± 1.26	23.41 ± 1.25	37.56 ± 0.78	44.53 ± 0.89	34.07 ± 0.86	24.19 ± 0.74
SimPro	35.70 ± 0.73	29.19 ± 0.90	27.26 ± 1.30	19.46 ± 1.53	33.56 ± 1.18	35.77 ± 1.72	31.45 ± 2.33	21.98 ± 1.26
AdapMatch	**53.16 ± 0.87**	**49.67 ± 0.94**	**44.63 ± 0.89**	**29.08 ± 0.71**	**61.66 ± 0.33**	**57.78 ± 0.47**	**55.88 ± 0.82**	**37.14 ± 1.02**

Table 8: Accuracy (%) on CIFAR-10, CIFAR-100, and CIFAR-100H with balanced class distributions ($\gamma = 1$) and partial-label ratio $\rho = 10\%$ (CIFAR-10) / 20% (CIFAR-100, CIFAR-100H) under varying label ambiguity rates q (excluding CIFAR-10/100 q=0.1 and CIFAR-100H q=0.3).

Method	**CIFAR-10** ($\rho = 10\%$)			**CIFAR-100** ($\rho = 20\%$)			**CIFAR-100H** ($\rho = 20\%$)		
	$q = 0.3$	$q = 0.5$	$q = 0.7$	$q = 0.1$	$q = 0.15$	$q = 0.2$	$q = 0.3$	$q = 0.5$	$q = 0.7$
PLCR	83.45 ± 0.37	82.77 ± 0.18	63.71 ± 0.48	47.03 ± 0.50	40.39 ± 0.84	36.81 ± 0.34	56.67 ± 0.28	50.21 ± 0.63	30.28 ± 1.39
SoLar	83.98 ± 0.43	78.46 ± 0.31	39.90 ± 0.20	45.28 ± 0.64	29.34 ± 0.45	25.04 ± 0.68	52.00 ± 0.41	47.33 ± 0.49	25.56 ± 0.62
SoLar*	81.01 ± 0.39	31.10 ± 0.74	29.53 ± 0.56	47.00 ± 0.33	39.51 ± 0.31	23.09 ± 0.40	59.49 ± 0.60	50.97 ± 0.52	23.76 ± 0.46
FixMatch+	91.02 ± 0.46	79.25 ± 0.89	41.00 ± 0.88	17.56 ± 0.11	15.41 ± 0.75	10.41 ± 1.65	35.06 ± 1.11	28.88 ± 0.63	23.61 ± 0.44
FreeMatch+	92.84 ± 0.16	91.35 ± 0.40	85.70 ± 0.41	55.01 ± 0.88	28.73 ± 3.56	12.02 ± 0.37	60.80 ± 0.56	56.95 ± 0.14	34.72 ± 1.16
ConCont	90.89 ± 0.19	90.46 ± 0.18	82.20 ± 0.71	62.85 ± 0.16	58.62 ± 0.22	50.77 ± 0.26	62.66 ± 0.23	59.83 ± 0.16	34.68 ± 1.03
FairMatch	93.73 ± 0.11	93.09 ± 0.03	91.73 ± 0.04	62.61 ± 0.14	59.75 ± 0.77	54.32 ± 0.22	65.12 ± 0.68	62.68 ± 0.42	51.13 ± 2.03
CPE	94.60 ± 0.15	94.90 ± 0.10	85.40 ± 0.17	65.28 ± 0.51	64.87 ± 0.70	58.33 ± 0.35	68.61 ± 0.75	63.21 ± 0.54	52.51 ± 2.10
SimPro	90.20 ± 0.25	93.80 ± 0.12	82.00 ± 0.35	61.57 ± 0.41	59.92 ± 0.82	51.65 ± 0.49	62.89 ± 0.85	61.86 ± 0.55	47.43 ± 2.30
AdapMatch	**95.69 ± 0.11**	**95.64 ± 0.03**	**95.49 ± 0.03**	**70.00 ± 0.06**	**67.48 ± 0.03**	**60.82 ± 1.18**	**72.27 ± 0.27**	**70.38 ± 0.19**	**58.81 ± 0.24**

E Comparative Prediction Analysis under Unknown Class Distributions

We evaluate the accuracy of class-distribution prediction for three SSPLL methods—FairMatch, ConCont, and our proposed AdapMatch—on CIFAR-10 with partial-label ratio $\rho = 10\%$ and ambiguity rate $q = 0.7$. Experiments are conducted under both the class-balanced setting ($\gamma = 1$) and the long-tailed setting ($\gamma = 10$). Results at the 300th epoch are reported in Fig. 8.

Overall, AdapMatch produces more accurate distribution estimates in both regimes. In the long-tailed case, AdapMatch assigns slightly higher mass to tail classes and lower mass to head classes compared with the ground-truth distribution. We attribute this behavior to the logit adjustment in the tail classifier, which admits more high-confidence tail samples into training and thereby enhances minority representation.

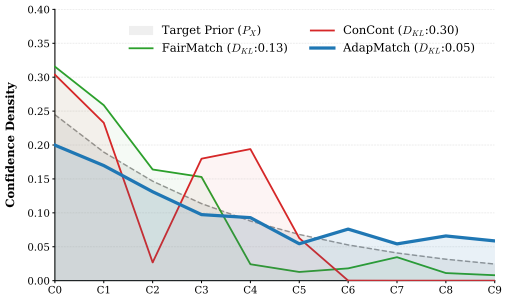
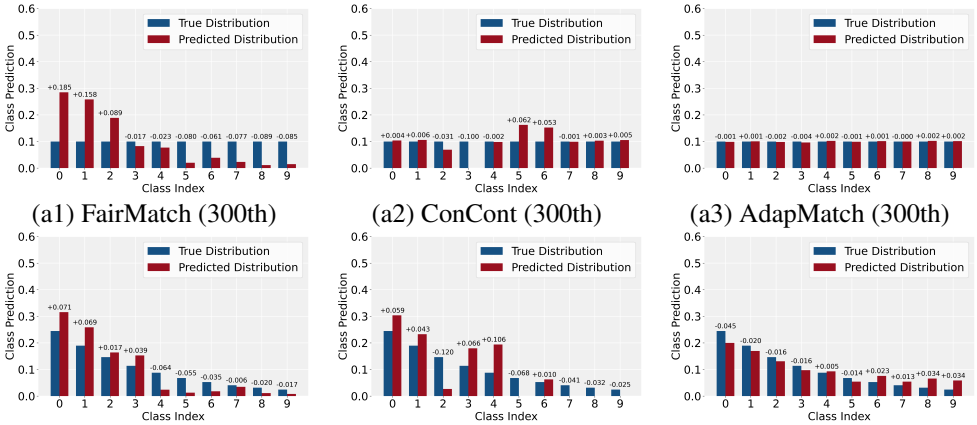


Figure 7: **Distribution fitting and bias diagnostic analysis.** DFS measures overall alignment stability under extreme imbalance. The class-wise confidence density is compared against the target prior (P_X); AdapMatch achieves the lowest KL divergence ($D_{KL} = 0.05$), indicating minimal bias.

To identify the source of this gain, we further visualize the class-wise confidence density on CIFAR-10 with $\gamma = 10$ and compare it with the target prior P_X (Fig. 7). Baselines show clear misalignment: FairMatch under-represents tail classes, while ConCont exhibits pronounced fluctuations, leading to a larger divergence ($D_{KL} = 0.30$). In contrast, **AdapMatch matches the target prior most closely**, achieving the lowest Kullback–Leibler divergence ($D_{KL} = 0.05$). This agreement indicates that AdapMatch effectively mitigates the head-class bias in long-tailed SSPLL, yielding confidence allocations that better reflect the true class frequency, including rare categories.

Table 9: Accuracy (%) on CIFAR-10, CIFAR-100, and CIFAR-100H under various partial-label settings.

Method	**CIFAR-10** ($\gamma = 10$, $\rho = 10\%$, $q = 0.7$)				**CIFAR-100** ($\gamma = 5$, $\rho = 20\%$, $q = 0.1$)				**CIFAR-100H** ($\gamma = 5$, $\rho = 20\%$, $q = 0.7$)			
	Overall	Head	Medium	Tail	Overall	Head	Medium	Tail	Overall	Head	Medium	Tail
PiCO	25.59	53.43	11.43	16.50	13.66	24.00	14.88	1.97	16.90	26.91	15.29	8.36
PLCR	27.32	70.77	15.22	0.00	27.52	44.36	28.56	9.61	33.78	56.21	28.47	6.82
SoLar	29.79	59.83	22.25	9.80	23.84	41.52	20.20	9.59	26.82	38.69	28.12	11.32
SoLar*	27.33	53.10	10.58	23.90	25.76	30.88	28.97	17.33	25.08	26.73	27.74	17.67
FixMatch+	32.05	78.80	21.05	0.00	23.09	36.21	25.56	7.42	26.80	49.55	23.71	7.24
FreeMatch+	36.93	76.63	29.00	7.80	27.79	41.09	30.65	11.55	18.99	28.05	20.89	7.98
ConCont	41.67	87.47	38.57	0.00	40.51	54.88	44.09	22.45	42.23	53.10	41.55	25.50
FairMatch	41.00	80.80	25.47	21.90	36.53	47.27	38.41	23.85	45.37	55.70	43.74	36.73
CPE	30.27	62.97	21.03	9.90	33.27	46.26	35.15	18.19	35.56	55.65	33.22	13.26
SimPro	28.23	61.78	15.34	11.87	25.46	33.60	27.47	15.17	32.46	54.12	29.42	9.37
AdapMatch	**81.21**	**78.67**	**75.80**	**90.97**	**46.25**	**52.58**	**47.12**	**37.48**	**55.88**	**60.45**	**55.26**	**51.94**



(a1) FairMatch (300th) (a2) ConCont (300th) (a3) AdapMatch (300th)

(b1) FairMatch (300th) (b2) ConCont (300th) (b3) AdapMatch (300th)

Figure 8: Comparison of predicted vs. true class distributions at the 300th epoch under different λ values. Subfigures (a1–a3) show the results under $\lambda = 1$, while subfigures (b1–b3) correspond to $\lambda = 10$. In each subplot, the blue bars represent the ground-truth class distribution, and the red bars indicate the predicted class distribution produced by the model. The numeric annotations above the bars denote the difference between the predicted and true class proportions. Values closer to zero indicate more accurate predictions.

F Additional Discussion and Empirical Analysis

In this section, we provide a more in-depth discussion regarding the hyperparameter sensitivity and model complexity of our proposed AdapMatch, further validating its efficiency and robustness.

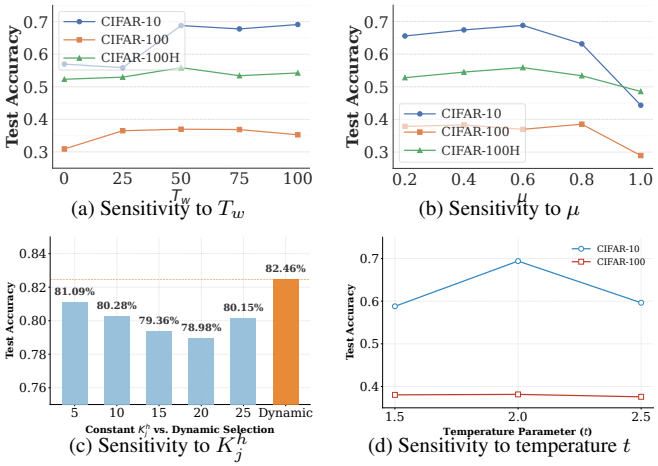
F.1 Ablation and Sensitivity Analysis

We study the sensitivity of AdapMatch to four key factors: (i) warm-up epochs T_w, (ii) EMA weight μ, (iii) replacing our **Dynamic Selection** with a fixed neighbor count, and (iv) the temperature parameter t.

Sensitivity of Warm-up Epochs T_w. Figure 9(a) reports the effect of warm-up epochs on CIFAR-10, CIFAR-100, and CIFAR-100H. A moderate warm-up achieves the best performance, with T_w=50 for CIFAR-10 and T_w=25 for CIFAR-100/100H. In contrast, removing warm-up (T_w=0) causes a clear performance drop, especially on CIFAR-10. This confirms that early training requires a short stabilization period before reliable pseudo-label admission and candidate disambiguation can be performed.

Sensitivity of EMA Weight μ. Figure 9(b) shows that AdapMatch is generally stable when $\mu \in \{0.2, 0.4, 0.6, 0.8\}$. However, setting μ=1 consistently degrades performance, indicating that directly replacing the memory with current predictions is less reliable. This result highlights the importance of historical smoothing in the confidence-memory update, which reduces the influence of noisy or biased early predictions.

Effect of Dynamic Selection vs. Fixed K. Note that K_j^h is *not* a tunable hyperparameter in our method; it is *dynamically determined* by the proposed selection logic for each anchor sample j. To better understand the role of adaptivity, we perform an ablation on CIFAR-10 by *forcing* a constant neighbor selection count K, i.e., replacing K_j^h with a fixed value. With a batch size of 256 and 10 classes, the theoretical maximum number of same-class neighbors in a batch is 25. As shown in Figure 9(c), the performance is relatively stable under different fixed K, while our **Dynamic Selection** achieves the best accuracy, highlighting the benefit of adaptive sample selection.



(a) Sensitivity to T_w (b) Sensitivity to μ (c) Sensitivity to K_j^h (d) Sensitivity to temperature t

Figure 9: **Sensitivity analysis of key hyperparameters.** (a) Sensitivity to warm-up epochs $T_w \in \{0, 25, 50, 75, 100\}$. (b) Sensitivity to EMA weight $\mu \in \{0.2, 0.4, 0.6, 0.8, 1\}$. (c) Comparison between constant K_j^h values and our dynamic selection on CIFAR-10. (d) Impact of the temperature parameter t on test accuracy for CIFAR-10 and CIFAR-100.

Sensitivity of Temperature t. Figure 9(d) reports the sensitivity of the temperature parameter t on CIFAR-10 and CIFAR-100. The model is robust to variations in t, with the best performance around t=2.0 on CIFAR-10. This suggests that an appropriate softening of the probability distribution helps candidate-level disambiguation, while the overall performance remains stable across a reasonable range of t.

Overall, AdapMatch is robust to a broad range of hyperparameter settings, while extreme choices such as removing warm-up or using μ=1 can weaken the stability of pseudo-label selection and memory-based prior estimation.

F.2 Model Complexity and Efficiency

Although AdapMatch integrates several adaptive modules, all hyperparameters are either fixed or self-adaptive, requiring no manual tuning. As summarized in Table 10, we evaluate the computational overhead in terms of parameter count (Params), floating-point operations (FLOPs), and training time per epoch.

Table 10: Model complexity analysis on CIFAR-10 with a ResNet backbone.

Method	Params (M)	FLOPs (G)	Time per Epoch (s)
FairMatch	11.50	22.85	32
ConCont	11.47	22.85	53
AdapMatch (Ours)	**11.18**	**11.42**	43

Notably, AdapMatch demonstrates exceptional computational efficiency, requiring approximately half the FLOPs (11.42G vs. 22.85G) compared to representative baselines like FairMatch and ConCont, while maintaining a competitive parameter count. This efficiency is primarily attributed to our streamlined architectural design: whereas FairMatch and ConCont often incur a doubled computational burden due to dual-view backbone processing (e.g., separate forward passes for multiple strong/weak augmentations), AdapMatch utilizes a shared-encoder $g(\cdot)$ with decoupled, lightweight heads. Although the online distribution estimation and sorting-based selection in the DAS module introduce a minor runtime overhead compared to FairMatch, AdapMatch remains significantly faster than ConCont. Consequently, our method achieves a superior Pareto balance between state-of-the-art predictive performance and practical computational expenditure

G Proofs for the Theoretical Analysis

This appendix formalizes a sufficient feedback mechanism for Theorem 3.1. The analysis focuses on a phenomenon specific to semi-supervised partial-label learning: admission-induced class-prior skew can increase partial-label disambiguation error, while disambiguation error can further bias the next-round admitted distribution.

G.1 Setup and Assumptions

Let P be the target distribution over (X, Y, S) and let Q_t be the effective training distribution at round t. Their class priors are

$$\pi(y) = P(Y = y), \qquad \pi_t(y) = Q_t(Y = y), \tag{18}$$

and the admission-induced prior gap is

$$\delta_t = \|\pi_t - \pi\|_1. \tag{19}$$

Let $\hat{y}_t(x)$ be the prediction used for candidate-label disambiguation. Since full-candidate samples do not provide an informative candidate constraint, we evaluate disambiguation on genuine partial-label samples

$$U = \{|S| < C\}, \qquad \rho_t = Q_t(U). \tag{20}$$

The corresponding disambiguation error is

$$d_t = Q_t(\hat{y}_t \neq Y \mid U). \tag{21}$$

All probabilities and expectations in this appendix are under Q_t unless stated otherwise.

Assumption G.1 (PLL realizability)**.** The ground-truth label belongs to the candidate set:

$$Q_t(Y \in S) = 1. \tag{22}$$

Assumption G.2 (Bounded candidate ambiguity)**.** There exists $1 \leq m_{\max} \leq C$ such that $|S| \leq m_{\max}$ on the support of Q_t. Define

$$\kappa = \log\left(1 + \frac{1}{m_{\max}}\right). \tag{23}$$

Assumption G.3 (Local prior-sensitive margin model)**.** Let $z_j^{(t)}(x)$ be the logit of class j. For each ground-truth class y and candidate false class $j \neq y$, the local pairwise margin admits the approximation

$$z_y^{(t)}(X) - z_j^{(t)}(X) = M_{y,j}(X) - \beta\Delta_t(j, y) + r_{t,yj}(X), \tag{24}$$

where $\beta > 0$,

$$\Delta_t(j, y) = \log\frac{\pi_t(j)}{\pi(j)} - \log\frac{\pi_t(y)}{\pi(y)}, \tag{25}$$

and $|r_{t,yj}(X)| \leq \bar{\varepsilon}_t$ for some $\bar{\varepsilon}_t \geq 0$.

Assumption G.3 is a mechanism-level local approximation rather than a global model of the whole training trajectory. It abstracts the dynamic logit-adjustment view in long-tailed partial-label learning: a mismatch between the effective and target class priors shifts pairwise margins through log-prior ratios. When $\Delta_t(j, y) > 0$, the false class j is relatively over-represented compared with the true class y, and the margin $z_y^{(t)} - z_j^{(t)}$ is reduced by the term $-\beta\Delta_t(j, y)$; when $\Delta_t(j, y) < 0$, the prior shift favors the true class relative to j.

Assumption G.4 (Margin regularity)**.** For each t, y, j, define

$$F_{t,y,j}(u) = Q_t(M_{y,j}(X) \leq u,\ j \in S \mid Y = y, U). \tag{26}$$

There exists $L > 0$ such that

$$|F_{t,y,j}(u) - F_{t,y,j}(v)| \leq L|u - v| \tag{27}$$

for all u, v.

Assumption G.5 (Positive class priors)**.** There exists $\pi_{\min} > 0$ such that, for all classes and analyzed rounds,

$$\pi(y) \geq \pi_{\min}, \qquad \pi_t(y) \geq \pi_{\min}. \tag{28}$$

Assumption G.6 (Admission sensitivity)**.** Let the prediction-induced prior be

$$\hat{\pi}_t(y) = Q_t(\hat{y}_t = y). \tag{29}$$

There exist $L_G > 0$ and $\xi_t \geq 0$ such that

$$\delta_{t+1} \leq L_G \|\hat{\pi}_t - \pi\|_1 + \xi_t. \tag{30}$$

Assumption G.6 is a stability condition on the admission operator. It states that if the prediction-induced class prior is close to the target prior, then the next effective training prior is also close to the target prior, up to the admission-side residual ξ_t.

Assumption G.7 (Risk transfer)**.** For any classifier g, the target risk satisfies

$$R_P(g) \leq R_{Q_t}(g) + \delta_t + \eta_t(g), \tag{31}$$

where $\eta_t(g) \geq 0$ captures class-conditional selection residual. If $P(X \mid Y) = Q_t(X \mid Y)$, one can take $\eta_t(g) = 0$.

G.2 Admission Gap to Disambiguation Error

We first decompose the partial-label disambiguation error into within-candidate and outside-candidate errors:

$$h_t^{\mathrm{par}} = Q_t(\hat{y}_t \in S \setminus \{Y\} \mid U), \qquad a_t^{\mathrm{par}} = Q_t(\hat{y}_t \notin S \mid U). \tag{32}$$

By Assumption G.1,

$$d_t = h_t^{\mathrm{par}} + a_t^{\mathrm{par}}. \tag{33}$$

Let $p_t(x)$ be the score distribution whose argmax defines $\hat{y}_t$. Define the partial-log loss

$$\ell_{\mathrm{pl}}(f_t; (x, S)) = -\log \sum_{j \in S} p_{t,j}(x), \tag{34}$$

and its risk

$$R_{Q_t}^{\mathrm{pl}}(f_t) = \mathbb{E}_{Q_t}\left[\ell_{\mathrm{pl}}(f_t; (X, S))\right]. \tag{35}$$

Lemma G.8 (Outside-candidate control)**.** *Under Assumption G.2, if $\rho_t > 0$, then*

$$a_t^{\mathrm{par}} \leq \frac{1}{\rho_t \kappa} R_{Q_t}^{\mathrm{pl}}(f_t). \tag{36}$$

Proof. Fix (x, S) and let $c = \hat{y}_t(x)$. If $c \notin S$, then $p_{t,c}(x) \geq p_{t,j}(x)$ for all $j \in S$. Hence

$$\sum_{j \in S} p_{t,j}(x) \leq |S| p_{t,c}(x) \leq m_{\max} p_{t,c}(x). \tag{37}$$

Since $c \notin S$,

$$p_{t,c}(x) + \sum_{j \in S} p_{t,j}(x) \leq 1. \tag{38}$$

Combining the two inequalities gives

$$\sum_{j \in S} p_{t,j}(x) \leq \frac{m_{\max}}{m_{\max} + 1}. \tag{39}$$

Therefore, on the event $\{\hat{y}_t(x) \notin S\}$,

$$\ell_{\mathrm{pl}}(f_t; (x, S)) \geq \log\left(1 + \frac{1}{m_{\max}}\right) = \kappa. \tag{40}$$

Thus

$$\mathbf{1}[\hat{y}_t(x) \notin S] \leq \frac{1}{\kappa} \ell_{\mathrm{pl}}(f_t; (x, S)). \tag{41}$$

Taking expectation and conditioning on U yield

$$a_t^{\mathrm{par}} = Q_t(\hat{y}_t \notin S \mid U) \leq \frac{Q_t(\hat{y}_t \notin S)}{\rho_t} \leq \frac{1}{\rho_t \kappa} R_{Q_t}^{\mathrm{pl}}(f_t). \tag{42}$$

□

The lemma uses the unconditional partial-log risk because the pointwise implication $\hat{y}_t(x) \notin S \Rightarrow \ell_{\mathrm{pl}}(f_t;(x,S)) \geq \kappa$ holds for all samples. Conditioning on the partial-label subset U only costs the factor $1/\rho_t$.

Proposition G.9 (Admission gap increases disambiguation error)**.** *Under Assumptions G.1–G.5, if* $\rho_t > 0$*, then*

$$d_t \leq \alpha\delta_t + r_t^{\mathrm{dis}}, \tag{43}$$

where

$$\alpha = \frac{LC\beta}{\pi_{\min}}, \tag{44}$$

and

$$r_t^{\mathrm{dis}} = e_t^\star + L(C-1)\bar{\varepsilon}_t + \frac{1}{\rho_t \kappa} R_{Q_t}^{\mathrm{pl}}(f_t), \tag{45}$$

with

$$e_t^\star = \sum_y Q_t(Y=y \mid U) \sum_{j \neq y} F_{t,y,j}(0). \tag{46}$$

Proof. If within-candidate mis-disambiguation occurs, then there exists $j \in S \setminus \{Y\}$ such that $z_j^{(t)}(X) \geq z_Y^{(t)}(X)$. By the union bound,

$$h_t^{\mathrm{par}} \leq \sum_y Q_t(Y=y \mid U) \sum_{j \neq y} Q_t(z_j^{(t)} \geq z_y^{(t)}, j \in S \mid Y=y, U). \tag{47}$$

By Assumption G.3,

$$z_j^{(t)} \geq z_y^{(t)} \Rightarrow M_{y,j}(X) \leq \beta\Delta_t(j,y) + |r_{t,yj}(X)| \leq \beta\Delta_t(j,y) + \bar{\varepsilon}_t. \tag{48}$$

Therefore,

$$Q_t(z_j^{(t)} \geq z_y^{(t)}, j \in S \mid Y=y, U) \leq F_{t,y,j}(\beta\Delta_t(j,y) + \bar{\varepsilon}_t). \tag{49}$$

By Assumption G.4,

$$F_{t,y,j}(\beta\Delta_t(j,y) + \bar{\varepsilon}_t) \leq F_{t,y,j}(0) + L\left(\beta|\Delta_t(j,y)| + \bar{\varepsilon}_t\right). \tag{50}$$

For any $a, b \geq \pi_{\min}$, $|\log a - \log b| \leq |a-b|/\pi_{\min}$. Thus Assumption G.5 gives

$$\begin{aligned}
&\sum_y Q_t(Y=y \mid U) \sum_{j \neq y} |\Delta_t(j,y)| \\
&\quad \leq \frac{1}{\pi_{\min}} \sum_y Q_t(Y=y \mid U) \sum_{j \neq y} \left(|\pi_t(j) - \pi(j)| + |\pi_t(y) - \pi(y)|\right) \\
&\quad \leq \frac{1}{\pi_{\min}} \sum_y Q_t(Y=y \mid U) \left(\|\pi_t - \pi\|_1 + (C-1)|\pi_t(y) - \pi(y)|\right) \\
&\quad \leq \frac{C}{\pi_{\min}} \|\pi_t - \pi\|_1 = \frac{C}{\pi_{\min}} \delta_t.
\end{aligned} \tag{51}$$

Therefore,

$$h_t^{\mathrm{par}} \leq e_t^\star + \frac{LC\beta}{\pi_{\min}} \delta_t + L(C-1)\bar{\varepsilon}_t. \tag{52}$$

Combining this bound with (33) and Lemma G.8 proves (43). □

The residual r_t^{dis} has three interpretable components: intrinsic candidate overlap $e_t^\star$, margin approximation error $\bar{\varepsilon}_t$, and outside-candidate error controlled by the partial-log loss.

G.3 Disambiguation Error to Next-round Admission Gap

Define the all-sample prediction error

$$\varepsilon_t^{\mathrm{all}} = Q_t(\hat{y}_t \neq Y). \tag{53}$$

Define the full-candidate residual error as

$$\varepsilon_t^{\mathrm{full}} = \begin{cases} Q_t(\hat{y}_t \neq Y \mid |S| = C), & Q_t(|S| = C) > 0, \\ 0, & Q_t(|S| = C) = 0. \end{cases} \tag{54}$$

Then

$$\varepsilon_t^{\mathrm{all}} = \rho_t d_t + (1 - \rho_t)\varepsilon_t^{\mathrm{full}}. \tag{55}$$

Lemma G.10 (Prediction-prior perturbation)**.** *The prediction-induced prior satisfies*

$$\|\hat{\pi}_t - \pi_t\|_1 \leq 2\varepsilon_t^{\mathrm{all}}. \tag{56}$$

Proof. For any class y,

$$|\hat{\pi}_t(y) - \pi_t(y)| = |Q_t(\hat{y}_t = y) - Q_t(Y = y)| \tag{57}$$

$$\leq Q_t(\hat{y}_t = y, Y \neq y) + Q_t(\hat{y}_t \neq y, Y = y). \tag{58}$$

Summing over y gives

$$\|\hat{\pi}_t - \pi_t\|_1 \leq 2Q_t(\hat{y}_t \neq Y) = 2\varepsilon_t^{\mathrm{all}}. \tag{59}$$

□

Proposition G.11 (Disambiguation error propagates to admission bias)**.** *Suppose $\varepsilon_t^{\mathrm{full}} \leq \zeta_t$. Under Assumption G.6,*

$$\delta_{t+1} \leq \gamma\delta_t + \lambda_t d_t + r_t^{\mathrm{adm}}, \tag{60}$$

where

$$\gamma = L_G, \qquad \lambda_t = 2L_G\rho_t, \tag{61}$$

and

$$r_t^{\mathrm{adm}} = 2L_G(1 - \rho_t)\zeta_t + \xi_t. \tag{62}$$

Proof. By Lemma G.10 and (55),

$$\|\hat{\pi}_t - \pi_t\|_1 \leq 2\rho_t d_t + 2(1 - \rho_t)\varepsilon_t^{\mathrm{full}} \leq 2\rho_t d_t + 2(1 - \rho_t)\zeta_t. \tag{63}$$

Moreover,

$$\|\hat{\pi}_t - \pi\|_1 \leq \|\hat{\pi}_t - \pi_t\|_1 + \|\pi_t - \pi\|_1 = \|\hat{\pi}_t - \pi_t\|_1 + \delta_t. \tag{64}$$

Substituting this into Assumption G.6 proves (60). □

G.4 Proof of Theorem 3.1

Proposition G.9 gives

$$d_t \leq \alpha\delta_t + r_t^{\mathrm{dis}}, \tag{65}$$

and Proposition G.11 gives

$$\delta_{t+1} \leq \gamma\delta_t + \lambda_t d_t + r_t^{\mathrm{adm}}. \tag{66}$$

Substituting the first inequality into the second yields

$$\delta_{t+1} \leq (\gamma + \lambda_t\alpha)\delta_t + \lambda_t r_t^{\mathrm{dis}} + r_t^{\mathrm{adm}}. \tag{67}$$

This proves Theorem 3.1.

G.5 Stability Consequence

Corollary G.12 (Stability of the coupled bias recursion)**.** *Let*

$$\omega_t = \gamma + \lambda_t \alpha, \qquad \epsilon_t = \lambda_t r_t^{\mathrm{dis}} + r_t^{\mathrm{adm}}. \tag{68}$$

If $\omega_t \le \bar{\omega} < 1$ *for all* $t = 0, \dots, T-1$*, then*

$$\delta_T \le \bar{\omega}^T \delta_0 + \sum_{s=0}^{T-1} \bar{\omega}^{T-1-s} \epsilon_s. \tag{69}$$

If, in addition, $\epsilon_s \le \bar{\epsilon}$ *for all* s*, then*

$$\limsup_{T \to \infty} \delta_T \le \frac{\bar{\epsilon}}{1 - \bar{\omega}}. \tag{70}$$

Proof. Theorem 3.1 gives

$$\delta_{t+1} \le \omega_t \delta_t + \epsilon_t \le \bar{\omega} \delta_t + \epsilon_t. \tag{71}$$

Unrolling this recursion yields (69). The second claim follows from the geometric-sum bound. □

Corollary G.12 shows that $\gamma + \lambda_t \alpha$ is the effective feedback strength. When it is below one and the residual terms are small, the admission-induced gap remains controlled; otherwise, early skew can be inherited or amplified across rounds.

G.6 Target-risk Consequence

Proposition G.13 (Target risk under coupled bias)**.** *Under Assumptions G.7 and the conditions of Theorem 3.1, if* $\varepsilon_t^{\mathrm{full}} \le \zeta_t$*, then*

$$R_P(\hat{y}_t) \le (1 + \rho_t \alpha)\delta_t + \rho_t r_t^{\mathrm{dis}} + (1 - \rho_t)\zeta_t + \eta_t(\hat{y}_t). \tag{72}$$

Proof. By Assumption G.7,

$$R_P(\hat{y}_t) \le R_{Q_t}(\hat{y}_t) + \delta_t + \eta_t(\hat{y}_t). \tag{73}$$

Since R_{Q_t} is the 0-1 classification risk,

$$R_{Q_t}(\hat{y}_t) = \varepsilon_t^{\mathrm{all}} = \rho_t d_t + (1 - \rho_t)\varepsilon_t^{\mathrm{full}}. \tag{74}$$

Using $\varepsilon_t^{\mathrm{full}} \le \zeta_t$ gives

$$R_P(\hat{y}_t) \le \rho_t d_t + (1 - \rho_t)\zeta_t + \delta_t + \eta_t(\hat{y}_t). \tag{75}$$

Finally, substitute $d_t \le \alpha \delta_t + r_t^{\mathrm{dis}}$ from Theorem 3.1. □

H Limitations

We identify two limitations of the current work and view them as promising directions for future investigation.

Backbone and modality diversity. All experiments in this paper are conducted with ResNet-18 on image classification benchmarks. While this is a standard and widely adopted protocol in the SSPLL literature, it remains an open question whether the proposed bias-decoupling mechanism transfers effectively to other architectures (e.g., Vision Transformers) or to non-vision modalities such as text and audio, where partial-label ambiguity may manifest differently.

Candidate set generation protocol. Following existing SSPLL studies, we construct candidate label sets by independently including each non-ground-truth class as a false candidate with probability q, resulting in variable-size candidate sets with class-agnostic corruption. In practice, however, partial-label noise is often instance-dependent or class-dependent—for example, visually similar classes are more likely to co-occur in a candidate set. The robustness of AdapMatch under such structured, non-uniform noise patterns has not been empirically validated, and this constitutes a meaningful gap between our experimental setup and real-world annotation scenarios.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction (Section 1) clearly state the main claims: (i) identifying two coupled biases (selection bias and disambiguation bias) in SSPLL under unknown class distributions, (ii) formalizing the coupled dynamics via Theorem 3.1 in Section 3, and (iii) proposing AdapMatch with DAS and CAD modules that yield up to 41.31% gain on CIFAR-10, 17.81% on STL-10, and 4.80% on ImageNet-127. These claims are consistently supported by the theoretical analysis and experimental results.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Yes. See Section H, where we discuss two aspects: (i) experiments are limited to ResNet-18 on image benchmarks, leaving generalization to other architectures and modalities unexplored; and (ii) candidate label sets are generated via class-agnostic random corruption with probability q, which differs from more realistic instance- or class-dependent noise.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best

judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: Theorem 3.1 (Coupled Bias Feedback) is stated in Section 3 with explicit conditions. The full set of assumptions and complete proofs for all propositions, lemmas, the corollary, and the theorem are provided in Appendix G, including the target-risk consequence (Appendix G.6).

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The paper provides comprehensive reproducibility details: dataset configurations , hyperparameter settings , optimizer details (SGD with Nesterov momentum 0.9, learning rate 0.01, weight decay 1×10^{-4}), data augmentation protocols (weak: flip/crop, strong: RandAugment), backbone architecture (ResNet-18), and the complete training procedure in Algorithm 1.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.

(b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
(c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The paper provides a public code repository at `https://github.com/yaodong-shen/AdapMatch`.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Section 5.1 and Appendix C provide complete experimental details: data splits with varying imbalance factors $\gamma \in \{1, 5, 10, 20\}$, partial-label ratios ρ, ambiguity rates q, optimizer configuration (SGD, momentum 0.9, lr 0.01, decay at epoch 700), batch size 256, total epochs 800, warm-up epochs 50, and all other hyperparameters.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: All main experimental results are reported as mean $\pm$ std over three independent runs, clearly reflecting performance variability, with standard deviations explicitly provided in the corresponding tables.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The paper provides model complexity analysis in Section F, including FLOPs, parameter count, and training time per epoch. We also explicitly specify the types of GPUs used for the experiments.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: This work is a fundamental machine learning method for semi-supervised partial label learning and does not involve any ethical concerns such as human subjects, sensitive data, or potential for misuse.

   Guidelines:

- The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [No]

    Justification: The paper does not discuss potential societal impacts. As a fundamental method for semi-supervised partial label learning, the work is not tied to specific applications or deployments.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper does not release any models or datasets with a high risk for misuse. It uses standard public benchmarks and proposes a method for classification under partial labels.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.

- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All datasets and baseline methods used in this paper are properly cited: CIFAR-10/100 [24], STL-10 [25], ImageNet-127 [26], and all baseline methods (PiCO, PLCR, SoLar, FixMatch, FreeMatch, ConCont, FairMatch, CPE, SimPro) are credited with their original papers.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The paper releases code at `https://github.com/yaodong-shen/AdapMatch` and provides detailed algorithmic description (Algorithm 1), model architecture (shared encoder with DAS and CAD branches), and complete hyperparameter specifications.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing or research with human subjects.

    Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing or research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.