STRONG VIEWS FOR REPRESENTATION, WEAK VIEWS FOR CLASSIFICATION: RETHINKING AUGMENTATION IN CLASS-IMBALANCED SEMI-SUPERVISED LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Class-imbalanced semi-supervised learning (CISSL) aims to leverage unlabeled data for both effective representation learning and balanced classification. Existing methods commonly use weak-to-strong augmentation consistency for both objectives, but we show that the appropriate augmentation strength should instead be chosen according to the learning objective. Strong augmentation promotes invariant representation learning, but it also shifts class-conditional means toward their center in feature space substantially more than weak augmentation, reducing the distance between classes. Under class imbalance, these smaller distances amplify the imbalance-induced boundary shift toward tail classes, shrinking their decision regions. These findings motivate a role-separated principle: use strong augmentation to learn representations, but weak augmentation to estimate the decision boundary. We instantiate this principle with the Weak-View Classifier (WVC), which estimates the final decision boundary from weakly augmented samples, where the distances between class means are better preserved. To improve the pseudo-labels used for decision-boundary estimation, we further introduce Reference-Guided Logit Adjustment (RGLA), which uses class prototypes as a reference to adapt the strength of logit adjustment without knowledge of the unlabeled class distribution. Across four CISSL benchmarks and multiple base SSL algorithms, our approach consistently improves decision-boundary estimation and achieves competitive or superior performance across diverse settings.

1 INTRODUCTION

Semi-supervised learning (SSL) leverages unlabeled data to improve both representation learning and classifier training. Modern SSL methods commonly generate a pseudo-label from a weakly augmented sample and enforce that label on a strongly augmented counterpart (Sohn et al., 2020; Berthelot et al., 2020; Wang et al., 2023; Chen et al., 2023). This weak-to-strong objective promotes robust representation learning by encouraging the encoder to preserve class-relevant information across substantial input perturbations. Throughout the paper, we use *weak view* and *strong view* as shorthand for samples produced by weak and strong augmentation, respectively.

Under class imbalance, however, scarce tail-class labels and biased pseudo-labels make balanced classifier estimation particularly difficult (Kim et al., 2020). Existing class-imbalanced SSL (CISSL) methods address this difficulty through pseudo-label refinement (Kim et al., 2020; Oh et al., 2022), loss reweighting (Lai et al., 2022; Park et al., 2024), auxiliary balanced classifiers (Lee et al., 2021; Fan et al., 2022), expert ensembles (Ma et al., 2024; Hou & Jia, 2025), and prior-based logit correction (Lee & Kim, 2024; 2025). Despite their different rebalancing mechanisms, these methods largely inherit the same augmentation policy from standard SSL: strong views are used not only to learn invariant representations but also to estimate the classifier from unlabeled data. This implicitly assumes that the augmentation policy effective for representation learning is also appropriate for classifier estimation. We show that this assumption can fail under class imbalance.

To understand why, we compare weakly and strongly augmented features produced by the same fixed encoder. We find that strong augmentation shifts class-conditional feature means toward their

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

center substantially more than weak augmentation, reducing between-class separation relative to within-class variability, as qualitatively illustrated in Fig. 1. Our analysis in Sec. 5 shows that under class imbalance, the boundary displacement toward the tail class is inversely proportional to the distance between the head- and tail-class means and proportional to the logarithm of the head-to-tail imbalance ratio (Eq. (12)). Strong augmentation therefore amplifies this displacement by reducing class-mean separation, shrinking tail-class decision regions and reducing tail-class accuracy. When class frequencies are balanced, this imbalance-induced displacement vanishes, and classifiers trained with weak and strong views perform nearly identically.

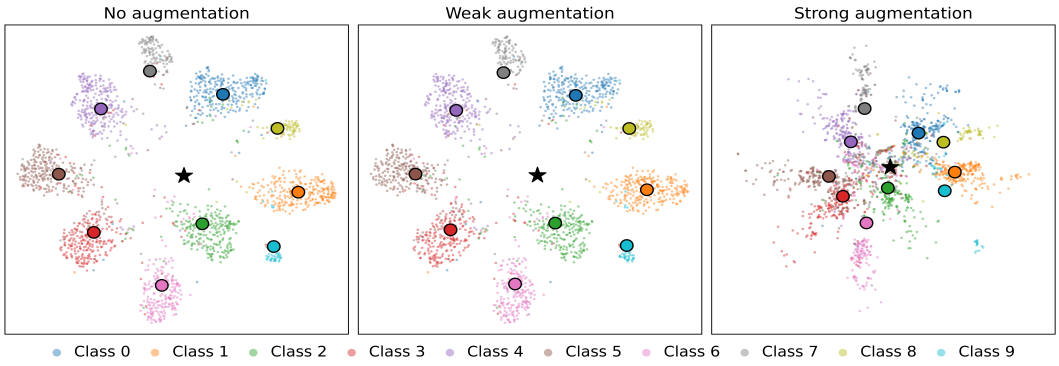


Figure 1: t-SNE visualization of features under no, weak, and strong augmentation on CIFAR-10-LT

These findings lead to our central principle: *use strong augmentation to learn representations, but weak augmentation to estimate the decision boundary*. We realize this principle with the *Weak-View Classifier (WVC)*, which estimates the final decision boundary from weakly augmented features with better-preserved distances between class means, while retaining the representation-learning benefits of strong augmentation. WVC is a separate classifier trained on detached features from the shared SSL encoder: backbone-generated pseudo-labels supervise WVC on weak views of the corresponding unlabeled samples, without propagating WVC's loss to the encoder. The weak-view principle also improves an existing balanced-classifier method without changing its encoder, pseudo-labels, or rebalancing mechanism (Sec. H).

WVC relies on backbone-generated pseudo-labels, making their quality critical to accurate decision-boundary estimation. Training-time logit adjustment (Menon et al., 2021) may mitigate class-frequency bias in these pseudo-labels, but its appropriate strength varies across datasets, unlabeled class distributions, and base SSL algorithms (Tab. 7). Adapting this strength is challenging because backbone predictions alone do not reveal the degree of class-frequency bias: they reflect both the model's own bias and the unknown class composition of the unlabeled data. We address this challenge with *Reference-Guided Logit Adjustment (RGLA)*, which uses a reference classifier to assess whether the backbone remains biased toward frequent classes or has been over-corrected. For this reference, we use a cosine nearest-class-mean classifier (cosNCM) (Kang et al., 2020), whose prototype-based decision rule does not directly encode class frequencies, making it suitable for assessing frequency bias in the backbone. RGLA compares the class distributions predicted by the backbone and cosNCM on the unlabeled data, increasing the adjustment strength when the backbone favors frequent classes more than cosNCM and decreasing it when the reverse occurs.

Experiments on CIFAR-10-LT (Cui et al., 2019), CIFAR-100-LT (Krizhevsky, 2009), STL-10-LT (Kim et al., 2020), and Small-ImageNet-127 (Fan et al., 2022), using FixMatch and ReMixMatch as base SSL algorithms, validate the proposed role separation across diverse CISSL settings. On the same trained encoder, WVC consistently outperforms its strong-view counterpart, with the advantage increasing as class imbalance becomes more severe and largely disappearing when class frequencies are balanced. WVC with RGLA achieves competitive or superior balanced accuracy across matched, mismatched, and reversed class-distribution settings.

2 Related Work

Class-imbalanced semi-supervised learning (CISSL) methods address class imbalance through pseudo-label refinement, balanced classifier learning, and loss correction. DARP (Kim et al., 2020)

refines pseudo-label distributions, while CReST (Wei et al., 2021) expands the labeled set with minority-class predictions. ABC (Lee et al., 2021) adds an auxiliary balanced classifier, whereas CoSSL (Fan et al., 2022) decouples representation and classifier learning. SAW (Lai et al., 2022) applies distribution-aware reweighting, DASO (Oh et al., 2022) combines linear and semantic pseudo-labels, and UDAL (Lazarow et al., 2023) integrates logit adjustment with distribution alignment.

Class-distribution mismatch further complicates CISSL because the appropriate logit-adjustment strength (Menon et al., 2021) varies across datasets and unlabeled distributions. ACR (Wei & Gan, 2023) matches the unlabeled distribution to one of three heuristic anchors, whereas CPE and Meta-Experts (Ma et al., 2024; Hou & Jia, 2025) use experts with fixed adjustment strengths. CDMAD (Lee & Kim, 2024) estimates classifier bias from a white-image, while LLA (Lee & Kim, 2025) learns class-wise adjustment parameters using class-averaged loss from training data. In contrast, RGLA adapts the adjustment strength using a reference classifier, without predefined anchors or estimation of the unlabeled data distribution. Although DASO also uses class prototypes, it uses them to generate pseudo-labels rather than to guide adjustment strength.

Across these approaches, classifier learning generally retains the conventional weak-to-strong augmentation scheme, with pseudo-labels generated from weak views supervising predictions on strong views. WVC instead separates the roles of augmentation: it retains strong augmentation for representation learning while estimating the final decision boundary from weakly augmented features, where the distances between class means are better preserved.

3 PRELIMINARIES

3.1 PROBLEM SETUP

For a K-class problem with classes indexed by $k \in \{1, \dots, K\}$, let $\mathcal{X} = \{(x_n^{\text{train}}, y_n^{\text{train}})\}_{n=1}^{N}$ and $\mathcal{U} = \{u_m^{\text{train}}\}_{m=1}^{M}$ be the labeled and unlabeled training sets, with one-hot label $y_n^{\text{train}} \in \{0,1\}^K$. Let N_k and M_k denote their class counts, and index labeled classes such that $N_1 \geq \cdots \geq N_K$. We define the imbalance ratios of the labeled and unlabeled sets as $\gamma_l = N_1/N_K$ and $\gamma_u = \max_k M_k / \min_k M_k$, respectively, where γ_u characterizes the experimental setting but is unavailable during training. We consider both matched ($\gamma_l = \gamma_u$) and mismatched ($\gamma_l \neq \gamma_u$) class-distribution settings, and sample labeled and unlabeled minibatches ($\mathcal{B}_X = \{(x_b, y_b)\}_{b=1}^{B_l}$ and $\mathcal{B}_U = \{u_b\}_{b=1}^{B_u}$), respectively. Let $z_\theta(x) \in \mathbb{R}^h$ be the encoder feature and $\phi : \mathbb{R}^h \to \mathbb{R}^K$ a classifier. We denote the backbone and WVC classifiers by ϕ_{back} and ϕ_{WVC}, respectively.

3.2 BASE SSL ALGORITHMS

Following prior CISSL work (Kim et al., 2020; Lee et al., 2021; Fan et al., 2022; Lee & Kim, 2024; 2025), we use FixMatch (Sohn et al., 2020) and ReMixMatch (Berthelot et al., 2020) as backbones. Both enforce consistency (Sajjadi et al., 2016; Miyato et al., 2018) between weak augmentation α and strong augmentation $\mathcal{A}$, implemented with RandAugment (Cubuk et al., 2020) and Cutout (DeVries & Taylor, 2017). Let $p(x) = \text{softmax}(\phi(z_\theta(x)))$ and let $\rho \in [0,1]$ be the confidence threshold. FixMatch forms $\hat{q}_b = p(\alpha(u_b))$ and computes weak-to-strong consistency regularization loss when $\max_k \hat{q}_{b,k} \geq \rho$, where $\hat{q}_{b,k}$ denotes the kth component of $\hat{q}_b$:

$$\mathcal{L}_{\text{con}} = \frac{1}{B_u} \sum_{u_b \in \mathcal{B}_U} \mathbf{I}\left(\max_k \hat{q}_{b,k} \geq \rho \right) \mathcal{H}(p(\mathcal{A}(u_b)),\, \hat{q}_b)\,, \tag{1}$$

where $\mathbf{I}$ is the indicator function and $\mathcal{H}$ is cross-entropy. FixMatch computes classification loss as

$$\mathcal{L}_{\text{cls}} = \frac{1}{B_l} \sum_{(x_b, y_b) \in \mathcal{B}_X} \mathcal{H}(p(\alpha(x_b)),\, y_b)\,. \tag{2}$$

Thus, the overall training loss of the base SSL algorithm is $\mathcal{L}_{\text{base}} = \mathcal{L}_{\text{cls}} + \mathcal{L}_{\text{con}}$. Details of ReMixMatch are provided in Sec. B.

3.3 LOGIT-ADJUSTED SOFTMAX CE LOSS

Logit adjustment (LA) (Menon et al., 2021) corrects class imbalance by adjusting logits according to the labeled distribution, either post hoc or within the loss. We use its loss-modification form in

CISSL, replacing Eqs. (1) and (2) with

$$\mathcal{L}_{\text{cls}} = \frac{1}{B_l} \sum_{(x_b, y_b) \in \mathcal{B}_X} \mathcal{H}(\text{softmax}(\phi(z_\theta(\alpha(x_b))) + \tau \cdot \lambda)\,,\, y_b)\,, \tag{3}$$

$$\mathcal{L}_{\text{con}} = \frac{1}{B_u} \sum_{u_b \in \mathcal{B}_U} \mathbf{I}\left(\max_k \hat{q}_{b,k} \geq \rho\right) \mathcal{H}(\text{softmax}(\phi(z_\theta(\mathcal{A}(u_b))) + \tau \cdot \lambda)\,,\, \hat{q}_b)\,, \tag{4}$$

where $\tau \geq 0$ is a hyperparameter that controls the adjustment strength and $\lambda = \log \pi$, with $\pi = (\pi_1, \ldots, \pi_K)$ and $\pi_k = N_k/N$. By adding larger offsets to more frequent classes during training, LA reduces their gradient contributions. Existing methods, however, typically choose τ heuristically.

4 METHOD

Our method separates the two roles of unlabeled data. The backbone retains the conventional weak-to-strong objective for representation learning, whereas WVC estimates the final decision boundary from the better-preserved class geometry under weak augmentation. WVC shares the backbone encoder but does not update it, allowing the two classifiers to use the same representation while serving different learning objectives. Because WVC depends on backbone-generated pseudo-labels, RGLA further improves its supervision by adapting logit adjustment strength through comparison with a prototype-based reference that does not directly favor frequent classes.

4.1 BACKBONE BRANCH

For each $u_b \in \mathcal{B}_U$, the backbone produces the weak-view pseudo-label $\hat{q}_b$. RGLA requires a stable estimate of the backbone's predicted class distribution over the unlabeled set. Because the distribution estimated from a single minibatch can be noisy, we aggregate minibatch estimates over iterations using an exponential moving average (EMA), which also avoids repeatedly evaluating the entire unlabeled set as $q_t^{\text{back}} = (1-\beta) q_{t-1}^{\text{back}} + \frac{\beta}{B_u} \sum_{b=1}^{B_u} \tilde{q}_b$, where $q_0^{\text{back}} = \pi$, β controls the EMA update rate, and $\tilde{q}_b = \text{onehot}(\arg\max_k \hat{q}_{b,k})$, with $\text{onehot}(j)$ denoting the K-dimensional one-hot vector for class j. RGLA uses q_t^{back} to determine the adjustment strength τ_t, and it is applied to the backbone losses $\mathcal{L}_{\text{base}} = \mathcal{L}_{\text{cls}} + \mathcal{L}_{\text{con}}$ in Eqs. (3) and (4), which update θ and ϕ_{back}.

4.2 WVC: WEAK-VIEW CLASSIFIER

WVC estimates the final decision boundary from weakly augmented labeled and unlabeled samples, whose class-mean distances are better preserved than under strong augmentation. WVC is implemented as a linear classifier on the shared encoder, with gradients stopped at the encoder to preserve the backbone's representation-learning objective.

Because WVC is supervised by both labeled and pseudo-labeled data, its logit adjustment should reflect the class composition of both sources. Using only the labeled prior π would ignore the class distribution of the unlabeled set. We therefore estimate the combined class distribution by weighting the two sources according to their sizes, N and M:

$$\pi_t^{\text{all}} = \frac{N\pi + M q_t^{\text{back}}}{N + M}. \tag{5}$$

Here, $N\pi$ gives the labeled class counts, while $M q_t^{\text{back}}$ estimates the unlabeled class counts from the backbone predictions. We then define $\lambda_t^{\text{all}} = \log \pi_t^{\text{all}}$ and apply it with unit strength to the WVC losses, introducing no additional adjustment-strength hyperparameter. The overall WVC objective is $\mathcal{L}_{\text{WVC}} = \mathcal{L}_{\text{WVC}}^l + \mathcal{L}_{\text{WVC}}^u$, where

$$\mathcal{L}_{\text{WVC}}^l = \frac{1}{B_l} \sum_{(x_b, y_b) \in \mathcal{B}_X} \mathcal{H}\big(\text{softmax}\big(\phi_{\text{WVC}}(z_\theta(\alpha(x_b))) + \lambda_t^{\text{all}}\big)\,, y_b\big)\,, \tag{6}$$

$$\mathcal{L}_{\text{WVC}}^u = \frac{1}{B_u} \sum_{u_b \in \mathcal{B}_U} \mathcal{H}\big(\text{softmax}\big(\phi_{\text{WVC}}(z_\theta(\alpha(u_b))) + \lambda_t^{\text{all}}\big)\,, \tilde{q}_b\big)\,. \tag{7}$$

As WVC is detached from the encoder, $\mathcal{L}_{\text{WVC}}$ updates only ϕ_{WVC}. For a test sample x^{test}, WVC predicts $\hat{y}^{\text{test}} = \arg\max_{k=1,\ldots,K} [\phi_{\text{WVC}}(z_\theta(x^{\text{test}}))]_k$.

4.3 RGLA: Reference-Guided Logit Adjustment

WVC relies on backbone-generated pseudo-labels, whose class-frequency bias can distort decision-boundary estimation. However, the backbone's predicted class distribution alone cannot distinguish such bias from the underlying class distribution of the unlabeled data. For example, head-heavy predictions may result either from the backbone's preference for frequent classes or from a genuinely head-heavy unlabeled distribution. RGLA addresses this ambiguity by comparing the backbone with the prototype-based reference on the same unlabeled data and using their discrepancy to guide logit adjustment. We use cosNCM as the reference because it predicts classes by cosine similarity to class prototypes without explicitly using class frequencies. The prototypes are updated from labeled features using EMA, with initialization and update details provided in Sec. J. For a weakly augmented sample $\alpha(u_b)$, let $\hat{y}_b^{\mathrm{ncm}}$ denote its cosNCM prediction. To obtain a stable reference distribution, we track the cosNCM predicted class distribution using EMA, analogously to q_t^{back} as $q_t^{\mathrm{ncm}} = (1-\beta)q_{t-1}^{\mathrm{ncm}} + \frac{\beta}{B_u}\sum_{b=1}^{B_u} \mathrm{onehot}(\hat{y}_b^{\mathrm{ncm}})$, where $q_0^{\mathrm{ncm}} = \pi$.

To quantify how strongly a given predicted class distribution q favors head classes, we assign each class k a frequency-based weight $\lambda_k = \log \pi_k$, which is larger for head classes. We then define the *head-preference score* $S(q)$ in Eq. (8) as the q-weighted average of λ_k. Thus, assigning more prediction mass q_k to head classes increases $S(q)$. RGLA then compares the backbone and cosNCM using the difference between their head-class preference scores, h_t in Eq. (8).

$$S(q) = \sum_{k=1}^{K} q_k \lambda_k, \qquad h_t = S(q_t^{\mathrm{back}}) - S(q_t^{\mathrm{ncm}}). \tag{8}$$

Because both scores use the same class-specific weights λ_k, their difference h_t captures whether the backbone favors the head classes more or less strongly than cosNCM. RGLA uses this difference as a control signal to adapt the logit-adjustment strength:

$$\tau_t = \tau_{t-1} + \eta h_t, \tag{9}$$

where $\tau_0 = 0$ and $\eta > 0$ is a controller rate. Increasing τ strengthens the logit adjustment toward tail classes. Thus, when $h_t > 0$, indicating that the backbone favors head classes more strongly than cosNCM, RGLA increases τ_t to reduce this discrepancy. Conversely, when $h_t < 0$, RGLA decreases τ_t, weakening the correction toward tail classes. The magnitude $|h_t|$ determines the size of update. Empirically, τ_t settles into a narrow range during training (Sec. L), and performance remains stable across the diverse controller hyperparameters (Sec. N).

4.4 End-to-end training

The overall training objective is

$$\mathcal{L} = \mathcal{L}_{\mathrm{base}} + \mathcal{L}_{\mathrm{WVC}}^{l} + \mathcal{L}_{\mathrm{WVC}}^{u}. \tag{10}$$

WVC is trained from the first iteration, with its gradients detached from the encoder. During warm-up, logit adjustment is disabled in both branches, while both marginals q_t^{back} and q_t^{ncm} are tracked before RGLA begins updating τ_t. Fig. 2 illustrates the WVC training procedure, while pseudo-code for the complete training algorithm is given in algorithm 1.

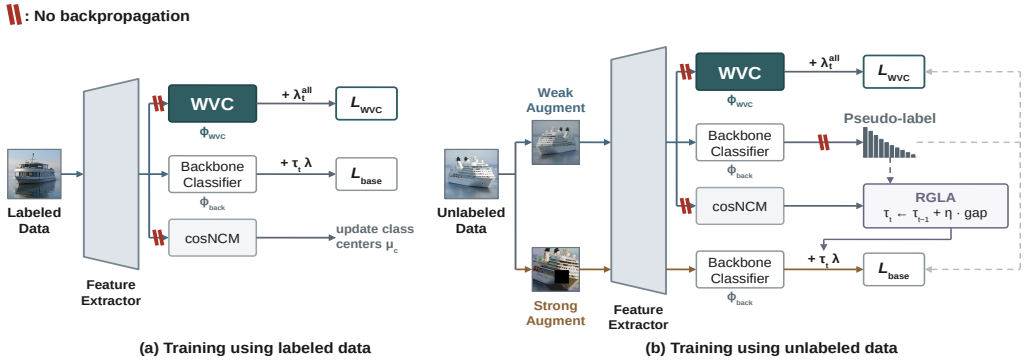


Figure 2: Training procedure of WVC.

5 WHY WEAK AUGMENTATION BENEFITS DECISION-BOUNDARY ESTIMATION

We investigate why weak augmentation is preferable to strong augmentation for decision-boundary estimation through theoretical and empirical analyses. First, strong augmentation shifts class-conditional feature means toward their center substantially more than weak augmentation, reducing the distances between class means (Sec. 5.1). Second, our two-class analysis shows that under class imbalance, smaller distances between class means amplify the imbalance-induced decision boundary shift toward the tail class, enlarging the head-class decision region (Sec. 5.2). Finally, we synthetically move weak-view class means closer together while preserving within-class variation and show that this reduces tail-class accuracy and balanced accuracy (Sec. 5.3).

5.1 STRONG AUGMENTATION MOVES CLASS MEANS CLOSER TOGETHER

We first examine whether strong augmentation moves class means toward their center more than weak augmentation on the same encoder. We refer to this inward movement as *class-mean contraction*. For each view $a \in \{\text{clean}, \text{weak}, \text{strong}\}$, let μ_k^a denote the mean feature vector of unlabeled samples from ground-truth class k, where clean, weak, and strong denote no, weak, and strong augmentation, respectively. For this analysis, we obtain class means using ground-truth labels rather than pseudo-labels to exclude pseudo-label errors, without using these ground-truth labels during training. We use the class-balanced centroid of the clean class means as the reference center: $\bar{\mu}^{\text{clean}} = \frac{1}{K}\sum_{k=1}^{K} \mu_k^{\text{clean}}$, where each class contributes equally regardless of class size. We then quantify class-mean contraction by measuring how far augmentation moves each class mean toward this centroid. Specifically, we project the augmentation-induced movement of the class mean, $\mu_k^a - \mu_k^{\text{clean}}$, onto the direction from the clean class mean to the centroid, $\bar{\mu}^{\text{clean}} - \mu_k^{\text{clean}}$, normalized by the squared distance to the centroid:

$$\omega_k^a = \frac{\left\langle \mu_k^a - \mu_k^{\text{clean}}, \bar{\mu}^{\text{clean}} - \mu_k^{\text{clean}} \right\rangle}{\left\| \bar{\mu}^{\text{clean}} - \mu_k^{\text{clean}} \right\|_2^2}, \tag{11}$$

where $\langle \cdot, \cdot \rangle$ denotes the inner product. Thus, ω_k^a measures the inward movement of class k's mean as a fraction of its original distance to the centroid. A value of $\omega_k^a = 0$ indicates no movement toward the centroid, whereas $\omega_k^a = 1$ indicates the component of movement toward the centroid equals the original class-to-centroid distance. Then, we define the class-averaged contraction coefficient $\bar{\omega}^a = \frac{1}{K}\sum_{k=1}^{K} \omega_k^a$. Across imbalanced CIFAR-10-LT settings, strong augmentation produces substantially larger $\bar{\omega}^a$ than weak augmentation (Sec. D). This confirms that strong augmentation induces greater class-mean contraction, bringing class means closer together and motivating the boundary analysis in Sec. 5.2.

5.2 CLOSER CLASS MEANS AMPLIFY BOUNDARY SHIFTS UNDER IMBALANCE

We next analyze how the reduced distance between class means affects the decision boundary under class imbalance. Consider a two-class problem with a head class and a tail class, and focus on the one-dimensional axis connecting their class means. For each view $a \in \{\text{weak}, \text{strong}\}$, we set the midpoint of the two class means to zero and take the direction toward the tail-class mean as positive. Let $d_a > 0$ denote the distance between the two class means under view a. We model the head- and tail-class features along this axis as Gaussian distributions with means $-d_a/2$ and $d_a/2$, respectively, and assume a common within-class variance σ^2 across classes and views. This assumption isolates the effect of the distance between class means. Let π_h and π_t denote the normalized proportions of the head and tail classes, with $\pi_h + \pi_t = 1$. The weak- and strong-view classifiers share the same π_h and π_t because they are trained using the same labeled and pseudo-labeled examples.

Proposition 1 (Boundary amplification under class imbalance)**.** *Under the model above, the Bayes decision boundary for view a is*

$$b_a = \frac{\sigma^2}{d_a} \log \frac{\pi_h}{\pi_t}, \tag{12}$$

where $b_a > 0$ denotes displacement from the midpoint toward the tail-class mean. If $\pi_h > \pi_t$ and $d_{\text{strong}} < d_{\text{weak}}$, then $b_{\text{strong}} > b_{\text{weak}} > 0$. Thus, the smaller class-mean distance under strong

augmentation amplifies the imbalance-induced boundary shift toward the tail class, enlarging the head-class decision region. If $\pi_h = \pi_t$, then $b_a = 0$ for either view, regardless of d_a.

The proof is provided in Sec. E. When both classifiers are evaluated on the same feature distribution with symmetric class means and equal within-class variance, the larger strong-view boundary enlarges the head-class decision region and lowers balanced accuracy. Since augmentation can also change within-class variation, the controlled intervention in Sec. 5.3 isolates the effect of class-mean distance by moving weak-view class means closer together while preserving within-class variation.

5.3 Moving weak-view class means closer together reduces tail accuracy

We next test how reducing the distances between class means affects classification performance under imbalance. For each unlabeled sample u_i, let $z_i^{\text{weak}} = z_\theta(\alpha(u_i))$ denote its weak-view feature, and let $k_i \in \{1, \dots, K\}$ denote its backbone pseudo-label class. Let $\widehat{\mu}_k$ be the mean weak-view feature of unlabeled samples assigned to class k. We then construct the modified feature $z_i(\omega) = z_i^{\text{weak}} + \omega(\bar{\mu} - \widehat{\mu}_{k_i})$, where $\bar{\mu} = \frac{1}{K}\sum_{k=1}^{K} \widehat{\mu}_k$ is the class-balanced centroid and $\omega \in [0, 1)$ controls how far the class means move toward the centroid. At $\omega = 0$, the features are unchanged. Increasing ω moves class means toward $\bar{\mu}$, multiplying their pairwise distances by $1 - \omega$ while preserving each sample's position relative to its class mean. For each ω, we train an additional detached classifier using $z_i(\omega)$ in place of the original weak-view features in the unlabeled loss. The labeled objective and all other training choices remain unchanged, and all classifiers are evaluated on the same unaugmented test features. This intervention therefore isolates the effect of reducing the distances between class means under the complete WVC objective.

Table 1: Controlled class-mean intervention on CIFAR-10-LT under $(\gamma_l, \gamma_u) = (150, 150)$. Head and tail recall are averaged over the three most and three least frequent classes, respectively.

Feature source	Real weak	Synthetic			Real strong
ω	–	0.35	0.576	0.75	–
bACC	82.5	81.7	80.0	79.9	79.8
Head recall	0.91	0.91	0.92	0.92	0.93
Tail recall	0.77	0.72	0.66	0.65	0.64

As ω increases, tail recall and balanced accuracy (bACC) decrease, while head recall remains high and slightly increases. In particular, $\omega = 0.576$ corresponds to the measured class-averaged contraction coefficient under strong augmentation, $\bar{\omega}^{\text{strong}}$ (Tab. 8). At this value, the modified weak-view features yield performance close to that obtained with real strong-view features, despite preserving the original weak-view within-class variation. These results support the mechanism predicted by Pro. 1: under class imbalance, reducing the distances between class means amplifies the boundary shift toward tail classes, shrinking their decision regions and reducing tail-class accuracy.

6 Experiments

6.1 Experimental setup

We evaluate on CIFAR-10-LT (Cui et al., 2019), CIFAR-100-LT (Krizhevsky, 2009), STL-10-LT (Kim et al., 2020), and Small-ImageNet-127 (Fan et al., 2022) (details in Sec. C), using balanced accuracy (bACC) (Huang et al., 2016) and geometric mean (GM) (Kubat et al., 1997). We compare WVC against representative CISSL baselines (Kim et al., 2020; Lee et al., 2021; Fan et al., 2022; Lai et al., 2022; Wei & Gan, 2023; Lee & Kim, 2024; Ma et al., 2024; Lee & Kim, 2025; Hou & Jia, 2025). Implementation and training details are provided in Sec. C. Unless otherwise specified, WVC refers to the complete proposed method, including RGLA for backbone logit adjustment.

6.2 Experimental Results

Tab. 2 summarizes the results on CIFAR-10-LT and CIFAR-100-LT under matched imbalance ($\gamma_l = \gamma_u$). WVC matches or outperforms the strongest baselines, including recent methods that assume $\gamma_l = \gamma_u$ and use the true unlabeled class distribution for alignment with ReMixMatch. The gains are particularly consistent on CIFAR-100-LT, where some classes contain only a single labeled sample under $\gamma = 100$, demonstrating that WVC remains effective even under severe label scarcity.

Table 2: CIFAR-10-LT (bACC/GM) and CIFAR-100-LT (bACC) under $\gamma = \gamma_l = \gamma_u$.

	CIFAR-10-LT			CIFAR-100-LT		
Algorithm	$\gamma = 50$	$\gamma = 100$	$\gamma = 150$	$\gamma = 20$	$\gamma = 50$	$\gamma = 100$
FixMatch	79.2±0.33/77.8±0.36	71.5±0.72/66.8±1.51	68.4±0.15/59.9±0.43	49.6±0.78	42.1±0.33	37.6±0.48
/+DARP	85.8±0.43/85.6±0.56	82.4±0.26/81.8±0.17	79.6±0.42/78.9±0.35	50.8±0.77	43.1±0.54	38.3±0.47
/+ABC	85.6±0.26/85.2±0.29	81.6±1.14/80.8±1.29	79.8±1.25/77.3±1.65	53.3±0.79	46.7±0.26	41.2±0.06
/+CoSSL	86.8±0.30/86.6±0.25	83.2±0.49/82.7±0.60	80.3±0.55/79.6±0.57	53.9±0.78	47.6±0.57	43.0±0.61
/+ACR	86.1±0.19/85.8±0.20	82.4±0.10/81.9±0.09	79.4±0.29/78.9±0.34	51.1±0.14	45.4±0.46	40.6±0.37
/+CPE	85.6±0.15/85.3±0.15	82.5±0.14/82.1±0.18	79.1±0.61/78.1±0.82	49.4±0.49	42.5±0.18	38.8±0.14
/+Meta-Experts	83.4±0.26/82.9±0.31	78.0±0.26/76.4±0.45	73.4±0.28/69.7±0.54	50.4±0.80	43.8±0.14	39.0±0.12
/+CDMAD	87.3±0.12/87.0±0.15	83.6±0.48/83.1±0.57	80.8±0.86/79.9±1.07	54.3±0.44	48.8±0.75	44.1±0.29
/+LLA	**88.1**±0.32/87.8±0.26	84.8±0.35/84.5±0.45	82.2±0.50/81.5±0.61	54.7±0.47	49.2±0.56	44.5±0.61
/+**WVC**	**88.1**±0.12/**87.9**±0.12	**85.0**±0.06/**84.7**±0.12	**82.4**±0.15/**81.9**±0.12	**55.5**±0.15	**49.5**±0.26	**44.6**±0.25
ReMixMatch	81.5±0.26/80.2±0.32	73.8±0.38/69.5±0.84	69.9±0.47/62.5±0.35	51.6±0.43	44.2±0.59	39.3±0.43
/+DARP+cRT	87.3±0.61/87.0±0.11	83.5±0.07/83.1±0.09	79.7±0.54/78.9±0.49	54.5±0.42	48.5±0.91	43.7±0.81
/+ABC	87.9±0.47/87.6±0.51	83.9±0.32/83.5±0.36	80.9±1.18/79.9±1.36	55.6±0.35	47.9±0.10	42.2±0.12
/+CoSSL	87.7±0.21/87.6±0.25	84.1±0.56/83.7±0.66	81.3±0.83/80.5±0.76	55.8±0.62	48.9±0.61	44.1±0.59
/+ACR	88.2±0.10/88.0±0.11	83.0±1.86/80.9±5.13	69.3±8.42/53.1±20.12	56.2±0.16	49.7±0.26	44.6±0.28
/+CPE	85.5±0.36/84.9±0.41	79.6±0.21/77.9±0.36	75.4±0.19/71.7±0.25	54.8±0.30	47.9±0.04	42.6±0.20
/+Meta-Experts	83.9±0.68/83.1±0.81	81.7±0.90/80.6±1.07	78.2±0.45/75.8±0.80	55.1±0.38	48.3±0.31	43.0±0.31
/+CDMAD	88.1±0.13/87.9±0.12	85.4±0.40/85.2±0.41	82.5±0.33/82.1±0.36	57.0±0.32	51.1±0.46	44.9±0.42
/+LLA	89.0±0.15/88.8±0.15	85.8±0.21/85.6±0.25	83.2±0.32/82.9±0.36	57.0±0.40	50.9±0.45	45.7±0.31
/+**WVC**	**89.1**±0.23/**88.9**±0.23	**86.0**±0.06/**85.7**±0.10	**83.4**±0.35/**83.0**±0.40	**57.5**±0.32	**51.4**±0.76	**46.4**±0.69

Tab. 3 summarizes results on CIFAR-10-LT with $\gamma_l \neq \gamma_u$ and STL-10-LT with unknown γ_u. WVC matches or outperforms the baselines, with RGLA adapting τ to each setting to improve pseudo-label quality and support WVC's classification of tail-class samples.

Table 3: bACC/GM on CIFAR-10-LT and STL-10-LT under $\gamma_l \neq \gamma_u$. ReMixMatch* uses the unlabeled class distribution estimate of Kim et al. (2020).

	CIFAR-10-LT ($\gamma_l = 100$)			STL-10-LT (γ_u unknown)	
Algorithm	$\gamma_u = 1$	$\gamma_u = 50$	$\gamma_u = 150$	$\gamma_l = 10$	$\gamma_l = 20$
FixMatch	68.9±1.95/42.8±8.11	73.9±0.25/70.5±0.52	69.6±0.60/62.6±1.11	72.9±0.09/69.6±0.01	63.4±0.21/52.6±0.09
/+DARP	85.4±0.55/85.0±0.65	77.3±0.17/75.5±0.21	72.9±0.24/69.5±0.18	77.8±0.33/76.5±0.40	69.9±1.77/65.4±3.07
/+ABC	84.5±0.49/83.7±0.68	82.7±0.64/82.0±0.76	78.4±0.87/77.2±1.07	79.1±0.46/78.1±0.57	73.8±0.15/72.1±0.15
/+SAW	81.2±0.68/80.2±0.91	79.8±0.25/79.1±0.32	74.5±0.97/72.5±1.37	78.3±0.25/77.0±0.19	71.9±0.81/69.0±0.81
/+ACR	87.0±1.01/86.7±1.07	83.9±0.50/83.6±0.56	81.6±0.38/81.1±0.40	79.3±0.41/78.2±0.50	75.7±0.65/74.3±0.66
/+CPE	**89.7**±0.21/89.5±0.21	83.3±0.37/82.9±0.36	81.0±0.20/80.4±0.25	82.1±0.33/81.4±0.40	79.5±0.12/78.7±0.12
/+Meta-Experts	88.4±0.53/88.0±0.61	79.9±0.38/78.7±0.48	76.9±0.36/74.9±0.44	81.6±0.35/81.0±0.47	79.5±0.19/78.5±0.21
/+CDMAD	87.5±0.46/87.1±0.50	85.7±0.36/85.3±0.38	82.3±0.23/81.8±0.29	79.9±0.23/78.9±0.38	75.2±0.40/73.5±0.31
/+LLA	88.6±0.35/88.4±0.40	**86.0**±0.26/85.7±0.26	83.2±0.25/82.7±0.35	82.6±0.21/81.9±0.25	79.5±0.23/78.6±0.26
/+**WVC**	89.6±0.35/89.4±0.35	**86.0**±0.12/**85.7**±0.15	**83.8**±0.36/**83.5**±0.42	**82.7**±0.35/**82.0**±0.40	**80.0**±0.30/**79.2**±0.40
ReMixMatch	48.3±0.14/19.5±0.85	75.1±0.43/71.9±0.77	72.5±0.10/68.2±0.32	67.8±0.45/61.1±0.92	60.1±1.18/44.9±1.52
ReMixMatch*	85.0±1.35/84.3±1.55	77.0±0.12/74.7±0.04	72.8±0.10/68.8±0.21	76.7±0.15/73.9±0.32	67.7±0.46/60.3±0.76
*/+DARP+cRT	88.7±0.25/88.5±0.25	83.5±0.53/83.1±0.51	80.9±0.25/80.3±0.31	80.9±0.53/80.0±0.46	76.7±0.50/74.9±0.70
/+ABC	81.0±5.34/79.8±6.05	85.2±0.20/84.7±0.25	80.4±0.40/80.0±0.44	76.8±0.52/74.8±0.64	71.2±1.37/67.4±1.89
/+ACR	89.2±0.15/89.0±0.25	83.6±2.23/80.3±5.71	73.8±6.00/59.2±17.10	84.2±0.26/83.5±0.35	81.5±0.42/80.2±0.42
/+CPE	88.9±2.05/88.4±2.45	81.9±0.31/80.8±0.36	78.2±0.58/76.1±0.78	82.8±0.55/81.9±0.70	77.7±0.66/75.5±1.20
/+Meta-Experts	87.8±0.50/86.9±0.38	83.2±0.45/82.4±0.50	78.7±3.35/76.3±4.97	83.6±0.55/82.6±0.72	81.2±0.41/79.2±0.66
/+CDMAD	89.9±0.45/89.6±0.46	86.9±0.21/86.7±0.17	83.1±0.46/82.7±0.50	83.0±0.38/82.1±0.35	81.9±0.32/80.9±0.44
/+LLA	90.4±0.25/90.2±0.36	87.2±0.25/87.0±0.26	83.8±0.20/83.6±0.26	84.0±0.44/83.2±0.53	82.1±0.29/81.1±0.32
/+**WVC**	**91.3**±0.29/**91.1**±0.29	**87.4**±0.15/**87.2**±0.15	**84.5**±0.31/**84.2**±0.31	**84.8**±0.17/**84.3**±0.17	**82.8**±0.61/**82.0**±0.72

Tab. 4 shows results on CIFAR-10-LT under the reversed setting, where the labeled and unlabeled class distributions are imbalanced in opposite directions. Despite the resulting challenge in pseudo-label generation, WVC and RGLA achieve consistently strong performance.

Table 4: bACC/GM on CIFAR-10-LT under reversed imbalance with $\gamma_l = \gamma_u = 100$.

Algorithm	ABC	SAW	SAW+LA	SAW+cRT	CDMAD	LLA	**WVC**
FixMatch+	69.5/66.8	72.3/68.7	74.1/72.0	75.5/73.9	77.1/75.4	74.0/73.0	**80.9/80.2**
ReMixMatch+	63.6/60.5	79.5/78.5	50.2/14.8	80.8/79.9	81.7/81.0	77.5/76.6	**84.7/84.2**

Tab. 5 summarizes bACC on Small-ImageNet-127, a large-scale long-tailed dataset with approximately 1.28M images from 127 classes and a severe imbalance ratio of about $\gamma \approx 286$. WVC outperforms all baselines at both image resolutions, demonstrating its effectiveness in large-scale settings with severe imbalance and many classes.

Table 5: bACC on Small-ImageNet-127 under matched imbalance ($\gamma = \gamma_l = \gamma_u$).

Image size	FixMatch	+CReST+LA	+ABC	+CoSSL	+CDMAD	+LLA	+**WVC**
32×32	29.7	40.9	46.9	43.7	48.4	52.2	**53.6**
64×64	42.3	55.9	56.1	53.8	59.3	62.1	**63.6**

Finally, to further assess WVC's compatibility with different base SSL algorithms, we evaluate it with FreeMatch (Wang et al., 2023) and SoftMatch (Chen et al., 2023) in Sec. P.

6.3 Detailed Analysis

We further analyze WVC and RGLA by examining whether weak views improve classifier learning over strong views on the same encoder, how the gains vary across head and tail classes, and how fixed adjustment strengths perform across different settings. Additional appendix studies extend the weak-view principle to another CISSL rebalancing method (Sec. H), analyze class-wise error patterns using confusion matrices (Fig. 3), compare RGLA with the post-hoc best fixed τ (Sec. M), and examine weak- versus strong-view learning under balanced class frequencies (Sec. G).

Weak versus strong views for classifier learning. Tab. 6 compares weak- and strong-view classifier training on the same trained encoder. WVC consistently outperforms its strong-view counterpart across all reported settings. Because the two classifiers differ only in the view used for the unlabeled loss, these results support the benefit of using weak views for decision-boundary estimation, consistent with the analysis in Sec. 5.

Table 6: Weak- versus strong-view classifier training with FixMatch on CIFAR-10-LT.

(γ_l, γ_u)	$(100, 150)$	$(150, 150)$	$(100, 100)$	$(100, 50)$	$(50, 50)$	$(100, 1)$
Classifier (strong view)	82.0	79.4	83.4	84.4	87.0	85.2
WVC (weak view)	**83.8**	**82.4**	**85.0**	**86.0**	**88.1**	**89.6**
$\Delta_{\text{weak}-\text{strong}}$	+1.8	+3.0	+1.6	+1.6	+1.1	+4.4

Need for setting-specific adaptation of τ. To examine how fixed adjustment strengths perform across different settings, we use FixMatch on CIFAR-10-LT with $(\gamma_l, \gamma_u) = (100, 100)$ as the reference setting and transfer the τ selected by RGLA to three target settings. Each target changes one factor: the unlabeled class distribution, dataset, or base SSL algorithm. We compare backbone classifiers using the transferred τ with those using the setting-specific τ selected by RGLA. Excluding the contribution of WVC isolates the benefit of adapting the adjustment strength. As shown in Tab. 7, RGLA selects different adjustment strengths and improves backbone performance in all three settings. Thus, an adjustment strength suitable for one setting may be suboptimal when the unlabeled distribution, dataset, or base SSL algorithm changes.

Table 7: Transfer of the reference $\tau = 0.80$, selected with FixMatch on CIFAR-10-LT under $(\gamma_l, \gamma_u) = (100, 100)$.

Target setting	Reference τ	Reference-τ backbone	RGLA τ	RGLA backbone	$\Delta_{\text{RGLA}-\text{Ref.}}$
CIFAR-10-LT, FixMatch, $(100, 1)$	0.80	86.7	1.11	**89.3**	+2.6
CIFAR-100-LT, FixMatch, $(100, 100)$	0.80	41.8	0.53	**43.6**	+1.8
CIFAR-10-LT, ReMixMatch, $(100, 100)$	0.80	81.9	0.50	**84.7**	+2.8

7 Conclusion

We revisited weak-to-strong augmentation in CISSL and showed that, while strong augmentation benefits representation learning, weak augmentation is better suited for decision-boundary estimation. Strong augmentation moves class means closer together than weak augmentation, amplifying boundary shifts toward tail classes under class imbalance. Controlled interventions confirm that reducing class-mean distances lowers tail-class and balanced accuracy, while the weak–strong gap disappears under balanced class frequencies. Based on these findings, we proposed WVC, a detached classifier trained on weak views, and RGLA, which adapts the backbone logit adjustment strength using a prototype-based reference. Across diverse imbalance and distribution-mismatch settings, WVC with RGLA matches or outperforms existing CISSL methods.

REPRODUCIBILITY STATEMENT

The complete objective, gradient paths, and controller are given in Sec. 4 and Sec. C. The supplementary material contains implementation details, all hyperparameters, dataset construction, proofs and anonymized source code.

AI USE STATEMENT

Generative AI tools were used to improve the manuscript's academic style and support code implementation and debugging. The authors reviewed all AI-assisted outputs and verified and tested all LLM-generated code for correctness. We take full responsibility for the final manuscript and all associated text, claims, and artifacts, including those developed with generative AI assistance.

ETHICS STATEMENT

We identify no ethical concerns associated with this work under the ICLR Code of Ethics.

REFERENCES

David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin Raffel. Mixmatch: A holistic approach to semi-supervised learning. In *NeurIPS*, 2019.

David Berthelot, Nicholas Carlini, Ekin D. Cubuk, Alex Kurakin, Kihyuk Sohn, Han Zhang, and Colin Raffel. Remixmatch: Semi-supervised learning with distribution matching and augmentation anchoring. In *ICLR*, 2020.

Hao Chen, Ran Tao, Yue Fan, Yidong Wang, Jindong Wang, Bernt Schiele, Xing Xie, Bhiksha Raj, and Marios Savvides. Softmatch: Addressing the quantity-quality trade-off in semi-supervised learning. In *ICLR*, 2023.

Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *In Proceedings of the fourteenth international conference on artificial intelligence and statistics (AISTATS)*, 2011.

Ekin Dogus Cubuk, Barret Zoph, Jon Shlens, and Quoc Le. Randaugment: Practical automated data augmentation with a reduced search space. In *NeurIPS*, 2020.

Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. Class-balanced loss based on effective number of samples. In *CVPR*, pp. 9268–9277, 2019.

Terrance DeVries and Graham W. Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017.

Yue Fan, Dengxin Dai, Anna Kukleva, and Bernt Schiele. Cossl: Co-learning of representation and classifier for imbalanced semi-supervised learning. In *CVPR*, pp. 14574–14584, 2022.

Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. In *ICLR*, 2018.

Yaxin Hou and Yuheng Jia. A square peg in a square hole: Meta-expert for long-tailed semi-supervised learning. In *ICML*, 2025.

Chen Huang, Yining Li, Chen Change Loy, and Xiaoou Tang. Learning deep representation for imbalanced classification. In *CVPR*, pp. 5375–5384, 2016.

Minyoung Huh, Pulkit Agrawal, and Alexei A Efros. What makes imagenet good for transfer learning? *arXiv preprint arXiv:1608.08614*, 2016.

Bingyi Kang, Saining Xie, Marcus Rohrbach, Zhicheng Yan, Albert Gordo, Jiashi Feng, and Yannis Kalantidis. Decoupling representation and classifier for long-tailed recognition. In *ICLR*, 2020.

Jaehyung Kim, Youngbum Hur, Sejun Park, Eunho Yang, SungJu Hwang, and Jinwoo Shin. Distribution aligning refinery of pseudo-label for imbalanced semi-supervised learning. In *NeurIPS*, 2020.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

Alex Krizhevsky. Learning multiple layers of features from tiny images. *Technical report, Department of Computer Science, University of Toronto*, 2009.

Miroslav Kubat, Stan Matwin, et al. Addressing the curse of imbalanced training sets: one-sided selection. In *Icml*, volume 97, pp. 179. Citeseer, 1997.

Zhengfeng Lai, Chao Wang, Henrry Gunawan, Sen-Ching Cheung, and Chen-Nee Chuah. Smoothed adaptive weighting for imbalanced semi-supervised learning: Improve reliability against unknown distribution data. In *ICML*, volume 162, pp. 11828–11843, 2022.

Justin Lazarow, Kihyuk Sohn, Chen-Yu Lee, Chun-Liang Li, Zizhao Zhang, and Tomas Pfister. Unifying distribution alignment as a loss for imbalanced semi-supervised learning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 5644–5653, 2023.

Hyuck Lee and Heeyoung Kim. Cdmad: Class-distribution-mismatch-aware debiasing for class-imbalanced semi-supervised learning. In *CVPR*, pp. 23891–23900, 2024.

Hyuck Lee and Heeyoung Kim. Learnable logit adjustment for imbalanced semi-supervised learning under class distribution mismatch. In *ICCV*, pp. 2664–2674, 2025.

Hyuck Lee, Seungjae Shin, and Heeyoung Kim. Abc: Auxiliary balanced classifier for class-imbalanced semi-supervised learning. In *NeurIPS*, 2021.

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 *AAAI*, 2024.

Aditya Krishna Menon, Sadeep Jayasumana, Ankit Singh Rawat, Himanshu Jain, Andreas Veit, and Sanjiv Kumar. Long-tail learning via logit adjustment. In *International Conference on Learning Representations*, 2021.

Takeru Miyato, Shin ichi Maeda, Masanori Koyama, and Shin Ishii. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(8):1979–1993, 2018.

Youngtaek Oh, Dong-Jin Kim, and In So Kweon. Daso: Distribution-aware semantics-oriented pseudo-label for imbalanced semi-supervised learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9786–9796, 2022.

Taemin Park, Hyuck Lee, and Heeyoung Kim. Rebalancing using estimated class distribution for imbalanced semi-supervised learning under class distribution mismatch. In *ECCV*, volume 15080, pp. 388–404, 2024.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

Mehdi Sajjadi, Mehran Javanmardi, and Tolga Tasdizen. Regularization with stochastic transformations and perturbations for deep semi-supervised learning. In *NeurIPS*, 2016.

Kihyuk Sohn, David Berthelot, Chun-Liang Li, Zizhao Zhang, Nicholas Carlini, Ekin D. Cubuk, Alex Kurakin, Han Zhang, and Colin Raffel. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. In *NeurIPS*, 2020.

Vikas Verma, Kenji Kawaguchi, Alex Lamb, Juho Kannala, Arno Solin, Yoshua Bengio, and David Lopez-Paz. Interpolation consistency training for semi-supervised learning. *Neural Networks*, 145:90–106, 2022.

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

Chen Wei, Kihyuk Sohn, Clayton Mellina, Alan Yuille, and Fan Yang. Crest: A class-rebalancing self-training framework for imbalanced semi-supervised learning. In *CVPR*, pp. 10857–10866, 2021.

Tong Wei and Kai Gan. Towards realistic long-tailed semi-supervised learning: Consistency is all you need. In *CVPR*, pp. 3469–3478, 2023.

Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In *In Proceedings of the British Machine Vision Conference (BMVC)*, 2016.

Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *ICLR*, 2018.

A PSEUDO-CODE

Algorithm 1 Pseudo-code for WVC training with RGLA

Require: Labeled dataset $\mathcal{X}$, unlabeled dataset $\mathcal{U}$, warm-up length T_{warm}
Ensure: Trained encoder z_θ and deployed classifier ϕ_{WVC}

Compute the labeled class prior π and set $\lambda = \log \pi$
Initialize $\tau_0 = 0$, prediction marginals $q^{\text{back}} = q^{\text{ncm}} = \pi$, and class centers $\{\mu_c\}_{c=1}^{K}$
for iteration $t = 1, \dots, T$ **do**
  Sample $\mathcal{B}_X = \{(x_b, y_b)\}_{b=1}^{B_l} \subset \mathcal{X}$ and $\mathcal{B}_U = \{u_b\}_{b=1}^{B_u} \subset \mathcal{U}$
  Generate weak views $\alpha(x_b), \alpha(u_b)$ and strong views $\mathcal{A}(u_b)$
  Compute weak-view features $z_b^l = z_\theta(\alpha(x_b))$ and $z_b^u = z_\theta(\alpha(u_b))$
  Generate backbone predictions $\hat{q}_b = softmax(\phi_{\text{back}}(z_b^u))$
  Set hard pseudo-labels $\tilde{q}_b = \text{onehot}(\arg\max_k \hat{q}_{b,k})$
  Update the backbone prediction marginal:
    $q^{\text{back}} \leftarrow (1-\beta) q^{\text{back}} + \frac{\beta}{B_u} \sum_{b=1}^{B_u} \tilde{q}_b$
  for each class c represented in $\mathcal{B}_X$ **do**
    Update the cosNCM center:
      $\mu_c \leftarrow \text{norm}\Big((1-m)\mu_c + m\ \text{norm}\Big(\frac{1}{|\mathcal{B}_{X,c}|} \sum_{(x_b,y_b)\in\mathcal{B}_{X,c}} z_b^l\Big)\Big)$
  end for
  Predict $\hat{y}_b^{\text{ncm}} = \arg\max_c \langle \text{norm}(z_b^u), \mu_c \rangle$
  Update the cosNCM prediction marginal:
    $q^{\text{ncm}} \leftarrow (1-\beta) q^{\text{ncm}} + \frac{\beta}{B_u} \sum_{b=1}^{B_u} \text{onehot}(\hat{y}_b^{\text{ncm}})$
  Estimate $\pi_{\text{all}} = \frac{N\pi + M q^{\text{back}}}{N+M}$
  if $t \le T_{\text{warm}}$ **then**
    Set $\tau_t = 0$ and $\lambda_{\text{WVC}} = \mathbf{0}$
  else
    Compute $\text{gap}_t = \langle q^{\text{back}} - q^{\text{ncm}}, \lambda \rangle$
    Update $\tau_t = \tau_{t-1} + \eta\, \text{gap}_t$
    Set $\lambda_{\text{WVC}} = \log \pi_{\text{all}}$
  end if
  Compute $\mathcal{L}_{\text{base}}$ using labeled weak views, unlabeled strong views, $\hat{q}_b$, and $\tau_t \lambda$
  Compute $\mathcal{L}_{\text{WVC}}^l$ using $\text{sg}(z_b^l)$, y_b, and λ_{WVC}
  Compute $\mathcal{L}_{\text{WVC}}^u$ using $\text{sg}(z_b^u)$, $\tilde{q}_b$, and λ_{WVC}
  Update $(\theta, \phi_{\text{back}})$ using $\mathcal{L}_{\text{base}}$
  Update ϕ_{WVC} using $\mathcal{L}_{\text{WVC}}^l + \mathcal{L}_{\text{WVC}}^u$
end for
return z_θ and ϕ_{WVC}

B BASE SSL ALGORITHMS

ReMixMatch generates pseudo-labels for unlabeled samples u_b by first computing $q_b = p(\alpha(u_b))$, and then applying distribution alignment and sharpening to obtain $\bar{q}_b$. Using these refined pseudo-labels, it optimizes both MixUp regularization (Zhang et al., 2018; Verma et al., 2022; Berthelot et al., 2019) and consistency regularization losses:

$$L_{mix} = \frac{1}{B_l} \sum_{x_b^{mix} \in \mathcal{X}_{mix}} \mathcal{H}(p(x_b^{mix}), y_b^{mix}) + \frac{1}{B_u} \sum_{u_b^{mix} \in \mathcal{U}_{mix}} \mathcal{H}(p(u_b^{mix}), \bar{q}_b^{mix}), \tag{13}$$

$$L_{con} = \frac{1}{B_u} \sum_{u_b \in \mathcal{B}_\mathcal{U}} \mathcal{H}(p(\mathcal{A}(u_b)), \bar{q}_b), \tag{14}$$

where $\mathcal{X}_{mix}$ and $\mathcal{U}_{mix}$ denote MixUp-augmented batches of labeled and unlabeled data, respectively. In addition, ReMixMatch employs a self-supervised rotation objective (Gidaris et al., 2018):

$$L_{rot} = \frac{1}{B_u} \sum_{u_b \in \mathcal{B}_\mathcal{U}} \mathcal{H}(f_{\theta_r}(r|\mathcal{R}(u_b, \hat{r})), \hat{r}), \tag{15}$$

where $\mathcal{R}(u_b, \hat{r})$ rotates image u_b by $\hat{r}$ degrees, and f_{θ_r} predicts the rotation angle using parameters θ_r. The overall objective of ReMixMatch is $L_{back} = L_{mix} + L_{con} + L_{rot}$.

C Detailed descriptions for the experimental setup

C.1 Datasets

CIFAR-10-LT and **CIFAR-100-LT** are long-tailed variants of CIFAR-10 and CIFAR-100 (Krizhevsky, 2009), respectively, constructed by reducing the number of training samples in each class according to a predefined imbalance ratio. We denote the number of labeled and unlabeled samples in the kth class by N_k and M_k, respectively, and define them as $N_k = N_1 \times \gamma_l^{-\frac{k-1}{K-1}}$ and $M_k = M_1 \times \gamma_u^{-\frac{k-1}{K-1}}$, where K is the total number of classes. For CIFAR-10-LT, we set $N_1 = 1500$ and $M_1 = 3000$, and first evaluate performance under the matched imbalance setting $\gamma = \gamma_l = \gamma_u$ with $\gamma \in \{50, 100, 150\}$. We further conduct experiments under mismatched imbalance settings by fixing $\gamma_l = 100$ and varying $\gamma_u \in \{1, 50, 150\}$. For CIFAR-100-LT, we set $N_1 = 150$ and $M_1 = 300$, and evaluate performance under the matched imbalance setting with $\gamma \in \{20, 50, 100\}$.

STL-10-LT is a long-tailed variant of STL-10 (Coates et al., 2011), constructed by reducing the number of labeled samples in each class according to a predefined imbalance ratio. We denote the number of labeled samples in the kth class by N_k and define it as $N_k = N_1 \times \gamma_l^{-\frac{k-1}{K-1}}$, where K is the total number of classes. STL-10-LT contains 100,000 unlabeled samples whose class distribution is unknown; we set $N_1 = 450$ and use the entire unlabeled set for training. We evaluate performance with $\gamma_l \in \{10, 20\}$.

Small-ImageNet-127 is a down-sampled version of ImageNet-127 (Huh et al., 2016), constructed by grouping the 1,000 classes from ImageNet (Russakovsky et al., 2015) into 127 classes. It contains approximately 1.28 million training samples with an imbalance ratio of about 286. Following Fan et al. (2022), we use 10% of the samples in each class as labeled samples and treat the remainder as unlabeled samples, with images down-sampled to resolutions of 32×32 and 64×64. As in (Wei et al., 2021; Fan et al., 2022), we conduct experiments on Small-ImageNet-127 only with FixMatch, and since its test set is also imbalanced we adopt balanced accuracy as the evaluation metric.

C.2 Training setups and baseline algorithms

We use Wide ResNet-28-2 (Zagoruyko & Komodakis, 2016) as the backbone network in all experiments. For optimization, we employ Adam (Kingma & Ba, 2015) with a learning rate of 1.5×10^{-3} for FixMatch and 2×10^{-3} for ReMixMatch. An exponential moving average (EMA) with a decay rate of 0.999 is used to update the network parameters at every iteration. Each experiment is repeated three times with different random seeds, and we report the mean and standard deviation of balanced accuracy (bACC) and geometric mean (GM) over the runs.

For CIFAR-100-LT, we set the weight decay to 0.08 for FixMatch and 0.1 for ReMixMatch to alleviate overfitting caused by the limited number of samples per class. For the other datasets, the weight decay is set to 0.04 for FixMatch and 0.06 for ReMixMatch when the total number of training samples is below 3×10^4, and to 0.01 for FixMatch and 0.02 for ReMixMatch otherwise, reflecting the lower risk of overfitting in larger datasets.

FixMatch is trained for 500 epochs, each consisting of 500 iterations, with labeled and unlabeled batch sizes of 32 and 64, respectively. To fully utilize the unlabeled data, we set the confidence threshold ρ to 0. For the large-scale dataset Small-ImageNet-127, we train the algorithm for 500 epochs using labeled and unlabeled batch sizes of 64 each. ReMixMatch is trained for 300 epochs, each with 500 iterations, using a labeled batch size of 64 and an unlabeled batch size of 128. We do not use the distribution alignment strategy of ReMixMatch, since the true class distribution of the unlabeled data is unknown, and instead include the supervised classification loss on weakly augmented labeled samples in the overall training loss. All experiments are conducted using NVIDIA RTX 3090 GPUs and PyTorch 1.8.1, and the source code is provided in the supplementary material.

D CONTRACTION MEASUREMENTS

This appendix reports the class-mean contraction measurements described in Sec. 5.1. For each CIFAR-10-LT setting, we hold the trained encoder fixed and compute features from the unlabeled set without augmentation and under weak and strong augmentation. We compute class means using ground-truth labels rather than pseudo-labels to avoid pseudo-label errors. These ground-truth labels are used only for this analysis, never during training.

We report the class-averaged contraction coefficient $\bar{\omega}^a = \frac{1}{K}\sum_{k=1}^{K} \omega_k^a$ for $a \in \{\text{weak}, \text{strong}\}$, using Eq. (11). This average gives equal weight to every class. Larger values indicate greater movement toward the class-balanced centroid computed without augmentation.

Table 8: Class-averaged contraction coefficients on CIFAR-10-LT.

(γ_l, γ_u)	$\bar{\omega}^{\text{strong}}$	$\bar{\omega}^{\text{weak}}$
$(50, 50)$	0.604	0.090
$(100, 1)$	0.521	0.041
$(100, 50)$	0.521	0.052
$(100, 100)$	0.602	0.066
$(100, 150)$	0.566	0.075
$(150, 150)$	0.576	0.059

Across all settings, strong augmentation moves class means toward the class-balanced centroid substantially more than weak augmentation. The average contraction coefficients range from 0.521 to 0.604 under strong augmentation, compared with 0.041 to 0.090 under weak augmentation.

E PROOF AND CONSEQUENCES OF BOUNDARY AMPLIFICATION

This appendix proves Pro. 1 and derives its implication for the controlled contraction intervention.

Proof of Pro. 1. Consider the head class h and tail class t described in Sec. 5.2. Let $U \in \mathbb{R}$ denote a feature projected onto their common direction, and let $Y \in \{h, t\}$ denote its class label. Under training view $a \in \{\text{weak}, \text{strong}\}$, assume

$$U \mid Y = h \sim \mathcal{N}\left(-\frac{d_a}{2}, \sigma^2\right), \qquad U \mid Y = t \sim \mathcal{N}\left(\frac{d_a}{2}, \sigma^2\right). \tag{16}$$

The Bayes classifier under this model assigns a projected feature value u to the head class when

$$\log \pi_h - \frac{(u + d_a/2)^2}{2\sigma^2} > \log \pi_t - \frac{(u - d_a/2)^2}{2\sigma^2}. \tag{17}$$

Rearranging gives

$$u < b_a, \qquad b_a = \frac{\sigma^2}{d_a} \log \frac{\pi_h}{\pi_t}. \tag{18}$$

If $\pi_h > \pi_t$, then $\log(\pi_h/\pi_t) > 0$. If $d_{\text{strong}} < d_{\text{weak}}$, then

$$b_{\text{strong}} = \frac{\sigma^2}{d_{\text{strong}}} \log \frac{\pi_h}{\pi_t} > \frac{\sigma^2}{d_{\text{weak}}} \log \frac{\pi_h}{\pi_t} = b_{\text{weak}} > 0. \tag{19}$$

Both boundaries therefore move toward the tail-class mean, and the contraction under strong augmentation produces the larger displacement.

Both classifiers are evaluated on the same feature distribution obtained without augmentation. Let $d_0 > 0$ denote the distance between its head- and tail-class means, and let $\sigma_0^2 > 0$ denote their common within-class variance along the same direction. We model these features as

$$U \mid Y = h \sim \mathcal{N}\left(-\frac{d_0}{2}, \sigma_0^2\right), \qquad U \mid Y = t \sim \mathcal{N}\left(\frac{d_0}{2}, \sigma_0^2\right). \tag{20}$$

For a classifier with boundary b, the head and tail recalls are

$$R_h(b) = \Phi\left(\frac{d_0/2 + b}{\sigma_0}\right), \qquad R_t(b) = \Phi\left(\frac{d_0/2 - b}{\sigma_0}\right), \tag{21}$$

where Φ is the standard Gaussian cumulative distribution function. Moving the boundary toward the tail class therefore increases head recall and decreases tail recall.

The corresponding balanced accuracy is

$$\mathrm{bACC}(b) = \frac{1}{2}\left[\Phi\left(\frac{d_0/2 + b}{\sigma_0}\right) + \Phi\left(\frac{d_0/2 - b}{\sigma_0}\right)\right]. \tag{22}$$

This function is symmetric around $b = 0$. For $b > 0$, its derivative is

$$\frac{\mathrm{d}}{\mathrm{d}b}\,\mathrm{bACC}(b) = \frac{1}{2\sigma_0}\left[\phi\left(\frac{d_0/2 + b}{\sigma_0}\right) - \phi\left(\frac{d_0/2 - b}{\sigma_0}\right)\right], \tag{23}$$

where ϕ is the standard Gaussian density. Since $|d_0/2 + b| > |d_0/2 - b|$ for $b > 0$, this derivative is negative. Therefore,

$$b_{\mathrm{strong}} > b_{\mathrm{weak}} > 0 \quad \Longrightarrow \quad \mathrm{bACC}_{\mathrm{strong}} < \mathrm{bACC}_{\mathrm{weak}}\,. \tag{24}$$

If $\pi_h = \pi_t$, then $\log(\pi_h/\pi_t) = 0$ and $b_{\mathrm{strong}} = b_{\mathrm{weak}} = 0$. The two boundaries therefore remain at the midpoint even when the class-mean distances differ. □

Corollary 1 (Effect of controlled contraction)**.** *Under the assumptions of Pro. 1, suppose that the class means move toward their midpoint by $\omega \in [0, 1)$ while every sample retains the same position relative to its class mean. Then the class-mean distance becomes $d(\omega) = (1 - \omega)d_{\mathrm{weak}}$, while the within-class variance remains σ^2. The resulting boundary is*

$$b(\omega) = \frac{\sigma^2}{(1 - \omega)d_{\mathrm{weak}}} \log\frac{\pi_h}{\pi_t} = \frac{b_{\mathrm{weak}}}{1 - \omega}. \tag{25}$$

If $\pi_h > \pi_t$, then $b(\omega)$ increases with ω. Contraction therefore increases head recall, decreases tail recall, and lowers balanced accuracy. If $\pi_h = \pi_t$, then $b(\omega) = 0$ for every ω.

For the synthetic features in Sec. 5.3, the mean of pseudo-label class k is

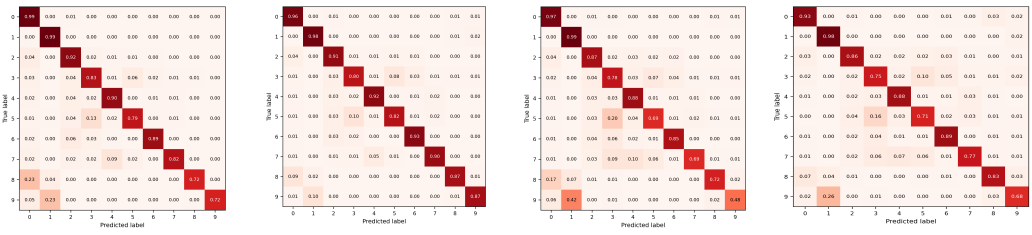
$$\mu_k(\omega) = (1 - \omega)\widehat{\mu}_k + \omega\bar{\mu}, \tag{26}$$

while every sample retains the same position relative to its class mean. The intervention therefore matches the changes in class-mean distance and within-class variance considered in corollary 1.

The proposition assumes the same within-class variance across the two classes and training views. This assumption isolates the effect of class-mean contraction rather than describing every change caused by augmentation. The controlled intervention tests the effect of class-mean contraction under the complete WVC training procedure.

F WHERE DOES WVC IMPROVE CLASSIFICATION?

Fig. 3 compares the test-set confusion matrices of the strong-view classifier and WVC, where entry (i, j) is the proportion of class-i samples predicted as class j. Classes are ordered from the most to the least frequent labeled class. Consistent with the boundary analysis in Sec. 5.2, weak-view learning reduces misclassification from tail to head classes.



(a) Strong (b) Weak (c) Strong (d) Weak

Figure 3: Test-set confusion matrices of the strong-view classifier and WVC on CIFAR-10-LT under (a–b) $(\gamma_l, \gamma_u) = (100, 1)$ and (c–d) $(150, 150)$.

G BALANCED SSL: CLASSIFIER AND CONTRACTION CONTROLS

This appendix examines whether the weak-view advantage persists under class balance and whether contraction alone produces a measurable performance gap. Experiments use FixMatch on CIFAR-10 with both the labeled and unlabeled sets uniformly distributed across classes. We consider 300, 100, and 10 labeled examples per class and fix the number of unlabeled examples to $3{,}000$ per class.

Strong augmentation still produces substantial class-mean contraction: the measured coefficient is 0.547, comparable to the 0.565 averaged over the imbalanced settings in Tab. 8. Class balance removes the class-frequency skew in the sampled data while leaving comparable geometric contraction in place, allowing us to test whether contraction alone produces a measurable weak–strong gap.

Table 9: bACC of WVC and the strong-view classifier under balanced SSL on CIFAR-10 using FixMatch.

Labeled/class	Backbone	Classifier (strong view)	WVC (weak view)	$\Delta_{\text{weak}-\text{strong}}$	$\Delta_{\text{weak}-\text{back.}}$
300	92.3	92.3	92.4	+0.1	+0.1
100	90.9	90.9	90.9	+0.0	+0.0
10	81.4	81.5	81.5	+0.0	+0.1

As shown in Tab. 9, WVC and the strong-view classifier differ by at most 0.1 bACC across all labeled-set sizes, and both remain close to the backbone. Two comparisons clarify the scope of this control. At 300 labeled examples per class, the total labeled supervision ($3{,}000$) is close to that of the imbalanced CIFAR-10-LT setting with $\gamma_l = 100$ ($\approx 3{,}700$), so the control is approximately matched in total supervision. At 10 examples per class, every class has fewer labeled examples than the least frequent class in the imbalanced reference setting ($N_K = 15$), yet no weak–strong gap emerges. Thus, abundant labeled supervision does not explain the result.

Class-mean contraction intervention. We further repeat the intervention of Sec. 5.3 with 10 labeled examples per class, sweeping ω while keeping all other training choices fixed.

Table 10: Class-mean contraction intervention under balanced SSL on CIFAR-10 with 10 labeled examples per class.

ω	0	0.25	0.50	0.75
bACC	81.5	81.4	81.5	81.4

Tab. 10 shows that contracting the class means leaves balanced accuracy essentially unchanged across the sweep. Together with the indistinguishable weak- and strong-view classifiers in Tab. 9, this indicates that label scarcity and geometric contraction alone are insufficient to produce the weak–strong gap. By contrast, under imbalanced training, the same contraction produces the asymmetric degradation documented in Sec. 5.3.

H COMPATIBILITY WITH ANOTHER CISSL ALGORITHM

We further examine whether weak-view classifier learning remains effective when combined with the balancing mechanism of another CISSL method. To this end, we train WVC using the same class-dependent undersampling scheme as ABC. Under this controlled construction, ABC and WVC use the same trained encoder, pseudo-labels, and undersampling rule, while differing in the view used for classifier learning: ABC uses strong views, whereas WVC uses weak views.

As shown in Tab. 11, ABC substantially improves upon the backbone head in all six settings, confirming the effectiveness of its undersampling mechanism. Nevertheless, replacing strong-view classifier learning with weak-view classifier learning yields a further improvement in every setting, with both FixMatch and ReMixMatch and under both matched and mismatched class distributions. This is what Sec. 5 predicts: undersampling changes the residual offset left to the classifier, and the view determines how strongly that residual affects boundary estimation. The weak-view principle

Table 11: bACC comparison of strong- and weak-view classifier learning using ABC on CIFAR-10-LT.

Base SSL	(γ_l, γ_u)	Backbone	ABC (strong view)	WVC (weak view)	$\Delta_{\mathrm{WVC-ABC}}$
FixMatch	$(100, 1)$	77.1	84.5	**86.7**	+2.2
FixMatch	$(100, 100)$	74.0	81.6	**84.6**	+3.0
FixMatch	$(150, 150)$	71.3	79.8	**83.0**	+3.2
ReMixMatch	$(100, 1)$	67.0	81.0	**86.1**	+5.1
ReMixMatch	$(100, 100)$	75.9	83.9	**85.9**	+2.0
ReMixMatch	$(150, 150)$	72.1	80.9	**83.6**	+2.7

is therefore compatible with an existing CISSL method rather than being specific to our proposed algorithm.

I cosNCM AND ADJUSTED CE ARE COMPLEMENTARY

cosNCM removes the learned classifier head from the readout and is therefore useful for measuring head-frequency bias. However, it is restricted to one normalized prototype per class and inherits the geometry of the shared encoder, whereas a learned CE head can form a more flexible discriminative boundary once its frequency bias is controlled. The two readouts therefore serve different roles: cosNCM provides a frequency-neutral direction for adapting τ, while the adjusted CE head generates the pseudo-labels used by WVC.

Tab. 12 empirically supports this separation. Without adjustment, cosNCM provides substantially better tail coverage than the backbone CE head. Its higher tail pseudo-label recall leads to clear improvements in final bACC, particularly under $(\gamma_l, \gamma_u) = (100, 1)$, despite not always improving overall pseudo-label accuracy. Thus, cosNCM is informative precisely when the unadjusted CE head severely omits tail examples.

RGLA further improves the tail coverage of both pseudo-labelers. Under $(100, 100)$, RGLA considerably increases tail pseudo-label recall while slightly reducing overall pseudo-label accuracy. Under $(100, 1)$, where the unlabeled distribution is balanced, RGLA substantially increases both overall pseudo-label accuracy and tail recall. Thus, the effect of adjustment is setting-dependent, but it consistently mitigates the severe omission of tail examples.

At the RGLA-selected adjustment strength, the backbone CE head achieves higher tail recall and final bACC than cosNCM in both settings. Unlike cosNCM, which summarizes the global class geometry through class prototypes, the CE head is trained to discriminate individual examples and therefore supplies more effective pseudo-labels to WVC after its frequency bias has been corrected. Accordingly, the better labeler at $\tau = 0$ need not remain the better labeler after adjustment.

The reference itself also evolves during training. Logit adjustment changes the backbone objective and hence the encoder on which both readouts operate; if this improves tail-class separation, the cosNCM prototypes and predictions improve as well. RGLA therefore forms a complementary loop rather than a competition between two fixed classifiers: cosNCM supplies a frequency-neutral control direction, the adjusted backbone improves the shared representation, and the more flexible CE head supplies the final pseudo-labels.

J A PROTOTYPE-BASED REFERENCE

RGLA requires a reference that reflects the current encoder geometry without explicitly incorporating class-frequency information. We use cosNCM for this purpose because it classifies samples by their cosine similarity to one prototype per class, without learning a classifier head. To track the evolving encoder while reducing minibatch variation, we maintain each class prototype using EMA.

Table 12: Pseudo-labeler comparison before and after adjustment on CIFAR-10-LT using FixMatch.

Setting	Encoder regime	Pseudo-labeler	PL accuracy	Tail PL recall	Test set bACC
$(100, 100)$	$\tau = 0$	Backbone CE	94.8	69.7	81.1
	$\tau = 0$	cosNCM	94.5	74.9	84.5
	RGLA	Backbone CE	93.6	85.8	85.0
	RGLA	cosNCM	93.6	81.8	84.6
$(100, 1)$	$\tau = 0$	Backbone CE	71.4	37.5	77.5
	$\tau = 0$	cosNCM	78.5	57.2	85.8
	RGLA	Backbone CE	83.3	72.3	89.6
	RGLA	cosNCM	82.7	66.9	88.1

We initialize $\mu_{k,0} = \mathbf{0}$ and update

$$\mu_{k,t} = \mathrm{norm}\left((1-m)\mu_{k,t-1} + m \cdot \mathrm{norm}\left(\frac{1}{|\mathcal{B}_{X,k}|} \sum_{(x_b,y_b)\in\mathcal{B}_{X,k}} z_\theta(\alpha(x_b))\right)\right). \tag{27}$$

Here, norm denotes ℓ_2 normalization, $\mathcal{B}_{X,k} = \{(x_b, y_b) \in \mathcal{B}_X : y_{b,k} = 1\}$, and m controls the EMA update rate. If $\mathcal{B}_{X,k}$ is empty, we retain the previous prototype, i.e., $\mu_{k,t} = \mu_{k,t-1}$. Given these prototypes, cosNCM predicts

$$\hat{y}_b^{\mathrm{ncm}} = \arg\max_k \langle \mathrm{norm}(z_\theta(\alpha(u_b))), \mu_{k,t} \rangle . \tag{28}$$

This gradient-free classifier tracks the evolving encoder with negligible additional overhead. Because its decision rule contains neither class-count terms nor learned classifier parameters, it provides a suitable reference for assessing the backbone classifier's preference for frequent classes, as discussed in Sec. K.

K Why cosNCM Provides a Reference for Frequency Bias

A learned parametric classifier is optimized through sample-wise gradients, so classes with more training examples exert greater cumulative influence and class frequency can be encoded in its parameters. In contrast, cosNCM represents each class by one normalized center and applies the same cosine-similarity rule to every class. Given the estimated centers, the decision rule in Eq. (28) contains neither class-count nor class-prior terms. It therefore does not directly favor a class simply because that class contributes more optimization updates, making cosNCM suitable for assessing the backbone classifier's preference for frequent classes.

This property does not make cosNCM generally unaffected by class imbalance. The shared encoder may learn imbalanced feature geometry, and centers estimated from fewer examples may be less accurate for infrequent classes. Rather, cosNCM avoids the direct encoding of class frequency through a learned parametric classifier head. Its reliability as a reference therefore also depends on the learned representation and the quality of its class-center estimates.

To examine the latter, we vary the number of labeled samples used to estimate each class center. As shown in Tab. 13, cosNCM using only three labeled samples per class achieves bACC within 0.3 points of using all labeled samples under $(\gamma_l, \gamma_u) = (100, 100)$ and within 0.2 points under $(100, 1)$. This suggests that center-estimation noise is not a dominant limitation of the reference in the evaluated settings.

L RGLA Dynamics Across Settings

We further examine the training dynamics of the adjustment strength selected by RGLA. Fig. 4 tracks τ_t after the controller is activated at the end of warm-up in four settings that vary the unlabeled distribution, dataset, or base SSL algorithm. All trajectories start from $\tau_0 = 0$.

Table 13: Balanced accuracy (%) of cosNCM with class centers estimated from k labeled samples per class on CIFAR-10-LT. "All" uses all available labeled samples.

(γ_l, γ_u)	$k = 3$	$k = 5$	$k = 10$	$k = 15$	All
$(100, 100)$	76.2	76.3	77.2	77.3	76.5
$(100, 1)$	85.1	84.5	85.3	85.7	85.3

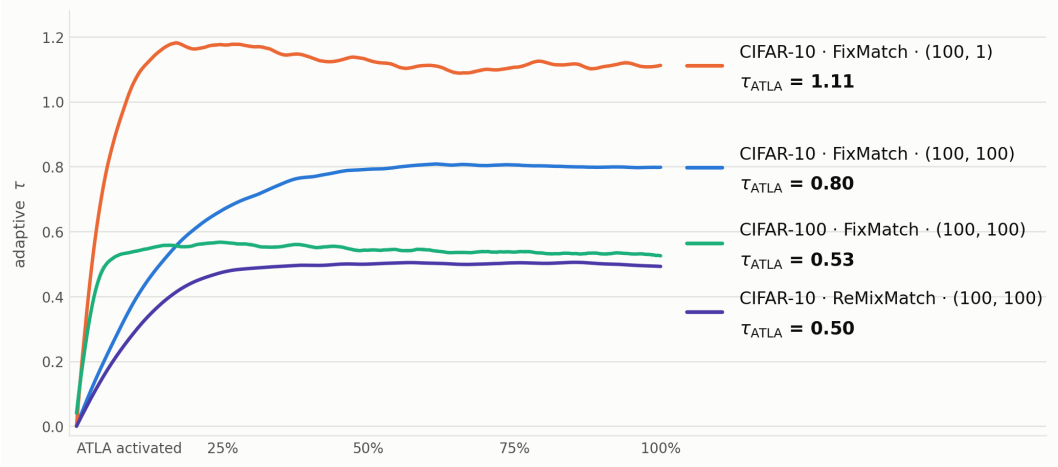


Figure 4: Evolution of the adjustment strength selected by RGLA after warm-up. RGLA rapidly approaches stable but distinct operating values across unlabeled distributions, datasets, and base SSL algorithms.

RGLA makes most of its adjustment early in training and subsequently remains within a narrow range, with only small fluctuations induced by stochastic minibatch marginals. More importantly, the resulting operating value varies substantially across settings even when the labeled imbalance ratio is unchanged. Changing the unlabeled distribution from $(100, 100)$ to $(100, 1)$ increases the selected strength, whereas changing either the dataset or the base SSL algorithm leads to a smaller value. These trajectories complement the transfer results in Tab. 7, showing that RGLA stably identifies a setting-dependent adjustment strength rather than relying on a particular value.

M RGLA VERSUS ORACLE-TUNED LOGIT ADJUSTMENT

We compare RGLA with the post-hoc best fixed adjustment strength selected separately for each setting through a grid search. As shown in Tab. 14, RGLA remains within 0.7 bACC points of the oracle fixed-τ result across all evaluated settings without requiring setting-specific tuning. The best fixed values outperform RGLA on CIFAR-10-LT with FixMatch under $(\gamma_l, \gamma_u) = (100, 100)$ and $(100, 1)$, and with ReMixMatch under $(100, 100)$, whereas on CIFAR-100-LT RGLA instead outperforms the best value identified by the fixed grid. Moreover, the best fixed τ varies from 0.60 to 1.30, and even for the same CIFAR-10-LT $(100, 100)$ setting it changes from 0.90 with FixMatch to 0.60 with ReMixMatch, showing that the appropriate adjustment strength depends not only on the dataset and unlabeled distribution but also on the base SSL algorithm. Overall, RGLA identifies a competitive setting-dependent operating point without requiring a separate grid search for each setting.

N CONTROLLER SENSITIVITY

We evaluate RGLA sensitivity using the FixMatch+RGLA backbone without WVC on CIFAR-10-LT under $(\gamma_l, \gamma_u) = (100, 100)$ and $(100, 1)$, thereby isolating the controller from the weak-view classifier. We independently vary the controller rate η, prediction-marginal EMA rate β, and class-

Table 14: Comparison of backbone balanced accuracy under RGLA and the post-hoc best fixed adjustment strength. The best fixed τ is selected separately for each setting through a grid search.

Dataset	Base SSL	(γ_l, γ_u)	RGLA τ	RGLA bACC	Best fixed τ	Best fixed bACC
CIFAR-10-LT	FixMatch	$(100, 100)$	0.80	83.5	0.90	**84.2**
CIFAR-10-LT	FixMatch	$(100, 1)$	1.11	89.3	1.30	**89.6**
CIFAR-100-LT	FixMatch	$(100, 100)$	0.53	**43.6**	0.60	43.3
CIFAR-10-LT	ReMixMatch	$(100, 100)$	0.50	84.7	0.60	**84.9**

center update rate m, while holding all other hyperparameters fixed. Each value is evaluated over three random seeds, and the tables report the mean backbone bACC and, for η, the mean final τ. As shown in Tabs. 15 to 17, bACC varies by at most 0.6 points under both $(100, 100)$ and 0.6 across all examined values. Varying η also changes the final τ by only 0.03 in both settings. These results show that RGLA reaches stable setting-dependent operating points without replacing direct tuning of τ with sensitive controller hyperparameters.

Table 15: Sensitivity of RGLA to the controller rate η. An asterisk denotes the default value.

(γ_l, γ_u)	Metric	0.10	0.15	0.20^*	0.25	0.30
$(100, 100)$	bACC	83.5	83.5	83.5	83.7	83.8
	Final τ	0.78	0.80	0.80	0.78	0.77
$(100, 1)$	bACC	89.4	89.4	89.3	88.9	89.0
	Final τ	1.13	1.14	1.11	1.12	1.11

Table 16: Sensitivity of RGLA to the prediction-marginal EMA rate β. An asterisk denotes the default value.

(γ_l, γ_u)	0.005	0.01^*	0.02	0.05	0.10
$(100, 100)$	83.5	83.5	83.2	83.8	83.8
$(100, 1)$	89.3	89.3	89.5	89.1	89.0

Table 17: Sensitivity of RGLA to the class-center update rate m. An asterisk denotes the default value.

(γ_l, γ_u)	0.005	0.01^*	0.02	0.05	0.10
$(100, 100)$	83.2	83.5	83.7	83.4	83.2
$(100, 1)$	89.0	89.3	89.2	88.9	89.2

O Computational overhead of WVC

We compare the training throughput of the backbone SSL algorithms with and without WVC on CIFAR-10-LT using an NVIDIA RTX 3090. As shown in Tab. 18, WVC, including RGLA, introduces negligible computational overhead.

Table 18: Training throughput on CIFAR-10-LT.

Algorithm	Iterations/s
FixMatch	16.84
FixMatch + WVC	16.71
ReMixMatch	6.01
ReMixMatch + WVC	5.98

P RESULTS WITH FREEMATCH AND SOFTMATCH AS THE BACKBONE SSL ALGORITHMS

To assess the general applicability of WVC to recent SSL methods, we further conduct experiments with FreeMatch (Wang et al., 2023) and SoftMatch (Chen et al., 2023) as the base SSL framework. The results in Tab. 19 and Tab. 20 show that WVC consistently outperforms the baseline, demonstrating its compatibility with modern SSL algorithms.

Table 19: Comparison of bACC/GM on CIFAR-10-LT when using FreeMatch as the backbone SSL algorithm

CIFAR-10-LT ($\gamma_l = 100$)		
Algorithm	$\gamma_u = 100$	$\gamma_u = 1$
FreeMatch (Wang et al., 2023)	73.6/ 69.4	72.2/ 64.4
FreeMatch+CDMAD (Lee & Kim, 2024)	83.1/ 82.5	85.9/ 85.3
FreeMatch+LLA (Lee & Kim, 2025)	83.8/ 83.4	87.9/ 87.5
FreeMatch+Meta-Experts (Hou & Jia, 2025)	78.1/ 76.4	88.3/ 87.8
FreeMatch+WVC	**85.0/ 84.7**	**89.9/ 89.6**

Table 20: Comparison of bACC/GM on CIFAR-10-LT when using SoftMatch as the backbone SSL algorithm

CIFAR-10-LT ($\gamma_l = 100$)		
Algorithm	$\gamma_u = 100$	$\gamma_u = 1$
SoftMatch (Chen et al., 2023)	75.9/ 73.0	75.3/ 70.9
SoftMatch+CDMAD (Lee & Kim, 2024)	81.8/ 81.1	87.8/ 87.3
SoftMatch+LLA (Lee & Kim, 2025)	83.8/ 83.4	89.3/ 89.1
SoftMatch+Meta-Experts (Hou & Jia, 2025)	78.7/ 77.1	87.8/ 87.2
SoftMatch+WVC	**85.1/ 84.7**	**89.6/ 89.4**