Spectrum of Engineering Sciences
ISSN (e) 3007-3138 (p) 3007-312X
Volume 4, Issue 3, 2026

Adaptive Multi-Granularity Mixture Consistency with Dynamic Prototype-Guided Rebalancing for Long-Tailed Visual Recognition

Din Muhammad[1], Hina Ali[2], Aqib Ali[1],Zahid Hussain[1], Muhammad Hammad [1], Farasat Ali Azeemi[3], Misbah Sangrasi[1], Pireh Hussain[1], Saif Ali[1]

[1] *Mehran university of Engineering and Technology, Pakistan*
[2]*Lecturer at Islamabad Model College, Pakistan*
[3] *Bahria University, Karachi, Pakistan*

[1]din.muhammad@faculty.muet.edu.pk, [2]hinaabdullah04@gmil.com, [1]aqibsw15@gmail.com, [1]zahidhussainkhaskheli@gmail.com, [1]sangrasihammad4@gmail.com [3]farasataliazeemi1@gmail.com, [1]misbahsangrasi@gmail.com, [1]saifsangrasi1@gmail.com, [1]pirehsoomro3@gmail.com,

DOI: https://doi.org/10.5281/zenodo.21306473

Keywords
Long-tail recognition, Imbalanced data, Image classification, deep learning, Features, SABLC, Multi-granularity Augmentation, Consistency Learning, Prototype Rebalancing,

Article History:
Received: 9 January 2026
Accepted: 25 February 2026
Published: 12 March 2026

Din Muhammad *

Abstract

Long-tailed visual recognition remains challenging i.e., severe class-frequency imbalance pushes deep models toward head-class accuracy at the expense of tail classes. Recent one-stage methods improve this along two separate fronts i.e., mixture-consistency augmentation, which pairs global and local image views for more robust representations, and adaptive rebalancing, which adjusts class weights using training-derived signals rather than fixed schedules. Each, however, has an open limitation: mixture-based methods such as GLMC use a small, fixed set of augmentation scales regardless of an image's semantic complexity, while existing dynamic rebalancing schemes rely on hand-tuned epoch schedules rather than a direct, representation-level measure of how well each class is currently learned. We present ADMG-LTR (Adaptive Multi-Granularity Mixture Consistency with Dynamic Prototype-Guided Rebalancing), a one-stage framework combining and extending both directions. An Adaptive Multi-Granularity Mixture (AMGM) module replaces the fixed global/local view pairing with a learnable scale selection, where a lightweight gating network sets each view's spatial extent from a per-image complexity estimate. A Consistency Alignment Loss (CAL) extends pairwise global-local consistency to enforce agreement jointly across all scale. K granularity levels. Dynamic Prototype-Guided Rebalancing (DPGR) maintains per-class momentum prototypes and uses prototype-to-sample distance as a live signal to modulate class weights, replacing epoch-scheduled or frequency-based weighting with one that tracks each class's evolving representation quality directly. We evaluate ADMG-LTR on standard long-tailed benchmarks across multiple imbalance ratios against representative one-stage baselines, including GLMC (mixture consistency) and AREA (adaptive reweighting), to assess the effectiveness of combining learnable multi-granularity augmentation.

1. Introduction

Real-world image datasets follow a long-tailed distribution [1]: a handful of head classes account for most labelled samples, while thousands of tail classes are represented by only a few. Training deep networks directly on such data yields classifiers that are strong on head classes but weak on tail classes and tail classes are frequently the ones that matter most, such as rare diseases in medical imaging [2] or uncommon hazards in autonomous driving.

Two broad families of remedies have been studied. Data-level methods resample the training distribution or augment the feature space [3, 4]; loss-level methods apply class-specific cost reweighting [5, 2]. Multi-stage pipelines [6, 7] decouple representation learning from classifier tuning and perform strongly, but at a noticeably higher training cost. More recently, one-stage contrastive and consistency-based methods [8, 9, 10] have closed much of this gap while avoiding the multi-stage overhead, and GLMC [10] in particular is, to our knowledge, the strongest published one-stage baseline on CIFAR-LT.

Building on GLMC's mixture-consistency idea, we identify two design choices in the current one-stage formulation that we believe are worth revisiting rather than treating as fixed.

Limitation 1: fixed augmentation granularity. GLMC pairs exactly one global MixUp view with one local CutMix view for every image, regardless of that image's content. A fine-grained object image may benefit from localised patch mixing, while a texture-heavy scene may benefit more from global blending. Using the same two fixed scales for every sample including the tail-class samples where representation quality matters most leaves a natural source of adaptivity unused.

Limitation 2: epoch-driven rebalancing. GLMC and BBN [11] schedule their rebalancing coefficient as a fixed function of epoch, $\alpha(T) = 1 - (T/T_{\max})^2$. This schedule cannot tell whether a given class's representation has actually stabilised; it shifts weight toward the rebalanced branch purely because training has reached a certain epoch, not because tail-class features are demonstrably ready for it.

Motivated by these two observations, we propose **ADMG-LTR**, a one-stage framework that extends the GLMC-style mixture-consistency paradigm along both axes. Concretely, we contribute:

1. **Adaptive Multi-Granularity Mixture (AMGM)**: we replace GLMC's fixed two-scale augmentation with a learnable, K-way scale selector. A lightweight gating network, trained end-to-end via Gumbel-Softmax relaxation [12], chooses a mixing scale per image conditioned on a semantic complexity estimate.

2. **Consistency Alignment Loss (CAL)**: we generalise GLMC's pairwise global-local cosine consistency to K granularity levels, adding a cross-scale term that encourages embeddings to remain stable as the mixing scale changes.

3. **Dynamic Prototype-Guided Rebalancing (DPGR)**: we replace the fixed epoch schedule with a signal computed from per-class momentum prototypes specifically, the average distance of a class's samples to its prototype so that rebalancing pressure tracks each class's actual representation quality rather than the epoch counter.

4. **Evaluation on CIFAR-10-LT and CIFAR-100-LT** across three imbalance ratios, directly comparing against GLMC and other one-stage and two-stage baselines under matched training conditions.

We position these contributions as a direct extension of GLMC's mixture-consistency framework rather than as an unrelated new paradigm, and we compare against GLMC throughout as our primary reference point.

2. Literature Review

2.1 Data Augmentation for Long-Tailed Learning

MixUp [13] and CutMix [14] remain the most widely used mixed-sample augmentations for long-tailed recognition. CMO [15] pastes minority-class patches onto majority-class backgrounds; RISDA [16] uses an implicit semantic-reasoning module to synthesise tail-class feature variation. GLMC [10] combines a single global MixUp view with a single local CutMix view as a fixed two-scale pipeline. Our AMGM module builds directly on this pipeline: instead of choosing between two fixed augmentation types, it learns an image-conditioned selector over K scales, allowing the mixing granularity to vary with each image's semantic complexity.

2.2 Contrastive and Consistency Learning

SimCLR [17] and MoCo [18] rely on large negative-sample queues; BYOL [19] and SimSiam [20] avoid negatives entirely through asymmetric stop-gradient architectures. Within long-tailed recognition, In [21, 22] authors use mean of head class angle variance to address the limitations of long-tailed data and in BCL [8] balances contrastive pull forces across classes, PaCo [9] introduces parametric class-wise prototypes into the contrastive objective, and GLMC [10] the method our CAL term most closely follows which uses cosine similarity between a global and a local mixed view as an auxiliary consistency loss. In [23] the authors use variance-Aware Group Knowledge Distillation to address the long-taile distribution issue in deep learning models.Our contribution here is narrow but specific: we extend GLMC's two-view consistency term to K views and add an explicit cross-scale alignment term, neither of which is present in the original formulation.

2.3 Class Rebalancing

Class-Balanced Loss [5] reweights by effective sample number; LDAM [2] imposes label-distribution-aware margins; BBN [11] shifts weight between a conventional and a reversed-sampling branch via a fixed cumulative schedule, which GLMC also adopts. Weight Balancing [7] instead regularises classifier weight norms directly, and decoupled training [6] separates representation learning from classifier fine-tuning across two stages. DPGR is best read as a drop-in replacement for the epoch-based schedule used by BBN and GLMC: rather than a function of training time, the rebalancing coefficient is computed from a live measurement of class-wise representation quality.

2.4 Prototype-Based Methods

ProtoNet [24] introduced prototype-based classification for few-shot learning, and prototype-guided self-distillation [25] uses head-class prototypes to improve tail-class representations through distillation. DPGR applies the same prototype machinery to a different end: rather than classification or distillation, per-class prototypes are used purely as a quality signal to drive adaptive loss weighting.

3. Problem Formulation

3.1 Setting and Notation

Let $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$ denote a long-tailed training set with C classes, where class c contains n_c samples and $n_1 \geq n_2 \geq \cdots \geq n_C$. The imbalance factor is $\rho = n_1/n_C \gg 1$. Following standard practice [1], classes are grouped into *Many-shot* ($n_c > 100$), *Medium-shot* ($20 \leq n_c \leq 100$), and *Few-shot* ($n_c < 20$) splits.

A model $\mathcal{F}_\theta : \mathcal{X} \to \mathbb{R}^C$ is factored as $\mathcal{F}_\theta = g_\phi \circ f_\psi$, where $f_\psi : \mathcal{X} \to \mathbb{R}^d$ is the backbone encoder and $g_\phi : \mathbb{R}^d \to \mathbb{R}^C$ is the linear classifier. We additionally define a projection head $\text{proj}_\xi : \mathbb{R}^d \to \mathbb{R}^{d_z}$, a prediction head $\text{pred}_\eta : \mathbb{R}^{d_z} \to \mathbb{R}^{d_z}$, and a gating network $\text{gate}_\omega : \mathcal{X} \to \Delta^{K-1}$ mapping images to a distribution over K mixing scales.

3.2 Multi-Granularity Mixing

Let $\mathcal{S} = \{s_k\}_{k=1}^{K}$ be a discrete set of spatial mixing scales, $s_k \in (0, 1]$, where $s_1 = 1$ recovers global MixUp and $s_K \ll 1$ recovers small-patch CutMix. Given a sampled pair $(\mathbf{x}_i, y_i)$, $(\mathbf{x}_j, y_j)$ and mixing ratio $\lambda \sim \text{Beta}(\beta, \beta)$, the mixed image at scale s_k is

$$\tilde{\mathbf{x}}^{(k)}_{\text{local}} = \mathbf{M}^{(k)} \odot \mathbf{x}_i + \left(\mathbf{1} - \mathbf{M}^{(k)}\right) \odot \mathbf{x}_j \,, \tag{1}$$

$$\tilde{\mathbf{x}}^{(k)} = \begin{cases} \lambda\, \mathbf{x}_i + (1-\lambda)\, \mathbf{x}_j \,, & s_k = 1 \;\; \text{(global)}, \\ \tilde{\mathbf{x}}^{(k)}_{\text{local}} \,, & s_k < 1 \;\; \text{(local)}, \end{cases} \tag{2}$$

where $\mathbf{M}^{(k)} \in \{0,1\}^{H \times W}$ is a binary mask whose filled area equals $s_k \cdot HW$. The corresponding mixed soft label is

$$\lambda_k = \begin{cases} \lambda \,, & k = 1, \\ 1 - s_k \,, & k > 1, \end{cases} \tag{3}$$

$$\tilde{\mathbf{p}}^{(k)} = \lambda_k\, \mathbf{p}_i + (1-\lambda_k)\, \mathbf{p}_j \,. \tag{4}$$

with $\mathbf{p}_i = \text{onehot}(y_i)$.

3.3 Learning Objective

The overall goal is to minimise expected risk under the (unobservable) balanced population distribution $\mathcal{P}^*$:

$$\min_{\theta} \; \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{P}^*} \Big[\, \mathcal{L}\big(\mathcal{F}_\theta(\mathbf{x}),\, y\big) \,\Big]. \tag{5}$$

Since $\mathcal{P}^*$ cannot be observed directly, we approximate it through the dynamically reweighted empirical distribution described in Section 4.3.

4. Proposed Methodology

Figure 1 provides a schematic overview of ADMG-LTR. The pipeline consists of three interacting modules i.e., (1) AMGM produces K semantically calibrated augmented views per image; (2) CAL enforces consistency across those views in representation space; and (3) DPGR adaptively reweights the classification loss using a live, prototype-derived signal of class-wise representation quality.

4.1 Adaptive Multi-Granularity Mixture (AMGM)

Gating network. For an input $\mathbf{x}_i$, the gating network produces a distribution over the K candidate scales,

$$\boldsymbol{\pi}_i = \text{gate}_\omega(\mathbf{x}_i) \in \Delta^{K-1} \,, \tag{6}$$

implemented as a two-layer convolutional network followed by global average pooling and a linear layer. To keep scale selection differentiable during training, we sample via the Gumbel-Softmax relaxation [12] at temperature τ,

$$\hat{\boldsymbol{\pi}}_i = \text{softmax}\left(\frac{\log \boldsymbol{\pi}_i + \mathbf{g}}{\tau}\right), \qquad \mathbf{g} \sim \text{Gumbel}(0,1)^K \,, \tag{7}$$

and select $k^* = \arg\max_k \pi_{i,k}$ deterministically at inference time.

Scale schedule. The K mixing scales are spaced uniformly in log-space,

$$s_k = \exp\left(-\frac{(k-1)\log(1/s_{\min})}{K-1}\right), \qquad k = 1, \ldots, K \,, \tag{8}$$

with $s_{\min} = 0.1$ and $s_1 = 1$ (global mixing).

Semantic complexity score. To bias simple images toward global mixing and complex, multi-object images toward local mixing, we define a per-image complexity score from the classifier's own prediction entropy,

$$\kappa(\mathbf{x}) = 1 - \max_c \hat{p}_c(\mathbf{x}) \,, \tag{9}$$

where a higher κ (a less confident, higher-entropy prediction) indicates a semantically complex image. A regularisation term encourages the gate's expected scale index to track this score,

$$\mathcal{L}_{\text{gate}} = \mathbb{E}_i \left[\left\| \bar{k}(\mathbf{x}_i) - (K-1)\, \kappa(\mathbf{x}_i) \right\|_2^2 \right], \tag{10}$$

where $\bar{k}(\mathbf{x}_i) = \sum_k k\, \hat{\pi}_{i,k}$ is the expected scale index under the current gate distribution.

4.2 Consistency Alignment Loss (CAL)

Given the K augmented views $\{\tilde{\mathbf{x}}^{(k)}\}_{k=1}^K$ of a sample pair (i,j), we obtain embeddings

$$\mathbf{h}^{(k)} = \text{proj}_\xi\big(f_\psi(\tilde{\mathbf{x}}^{(k)})\big) \,, \qquad \mathbf{u}^{(k)} = \text{pred}_\eta\big(\mathbf{h}^{(k)}\big) \,. \tag{11}$$

Within-granularity consistency. Following GLMC's negative cosine similarity formulation, we use

$$\mathrm{sim}(\mathbf{u}, \mathbf{h}) = -\frac{\mathbf{u}}{\|\mathbf{u}\|_2} \cdot \frac{\mathbf{h}}{\|\mathbf{h}\|_2}, \tag{12}$$

averaged over all ordered pairs of the K views,

$$\mathcal{L}_{\text{within}} = \frac{1}{K(K-1)} \sum_{k \neq l} \mathrm{sim}\big(\mathbf{u}^{(k)}, \mathrm{sg}(\mathbf{h}^{(l)})\big), \tag{13}$$

where $\mathrm{sg}(\cdot)$ denotes the stop-gradient operator. This term reduces to GLMC's original two-view loss when $K = 2$.

Cross-granularity consistency. We additionally require embeddings at adjacent scales to remain close, which GLMC does not enforce:

$$\mathcal{L}_{\text{cross}} = \frac{1}{K-1} \sum_{k=1}^{K-1} \mathrm{sim}\big(\mathbf{u}^{(k)}, \mathrm{sg}(\mathbf{h}^{(k+1)})\big). \tag{14}$$

Combined loss.

$$\mathcal{L}_{\text{CAL}} = \mathcal{L}_{\text{within}} + \mu\, \mathcal{L}_{\text{cross}}, \qquad \mu = 0.5 \text{ (default)}. \tag{15}$$

4.3 Dynamic Prototype-Guided Rebalancing (DPGR)

Momentum prototype bank. A prototype $\mathbf{q}_c \in \mathbb{R}^d$ is maintained per class,

$$\mathbf{q}_c \leftarrow m\, \mathbf{q}_c + (1-m)\, \bar{\mathbf{r}}_c, \qquad m = 0.99, \tag{16}$$

where $\bar{\mathbf{r}}_c$ is the mean encoder output over class-c samples in the current mini-batch.

Representational quality score. Intra-class compactness for class c is

$$\sigma_c = \frac{1}{n_c^{(b)}} \sum_{y_i = c} \big\| f_\psi(\mathbf{x}_i) - \mathbf{q}_c \big\|_2^2, \tag{17}$$

with $n_c^{(b)}$ the number of class-c samples in the batch. A large σ_c indicates scattered, under-developed representations, typically tail classes early in training.

Adaptive class weight. We replace GLMC's static frequency-based weight with

$$\Phi(z) = \frac{z^\nu}{z^\nu + 1}, \qquad \nu = 0.5, \tag{18}$$

$$w_c^{(\text{DPGR})} = \frac{1}{r_c} \cdot \Phi\Big(\frac{\sigma_c}{\bar{\sigma}}\Big). \tag{19}$$

where $r_c = n_c/N$ is the class frequency and $\bar{\sigma} = \frac{1}{C}\sum_c \sigma_c$. Φ maps $\Phi(1) = 0.5$, increases toward 1 as scatter rises above the mean, and decreases toward 0 for tightly clustered classes. Weights are normalised to sum to C,

$$\tilde{w}_c = \frac{C \cdot w_c^{(\text{DPGR})}}{\sum_{c'} w_{c'}^{(\text{DPGR})}}, \tag{20}$$

and mixed analogously to Eq. (4), $\tilde{w}^{(k)} = \lambda_k\, w_i + (1-\lambda_k)\, w_j$. The resulting weighted loss is

$$\hat{\mathbf{y}}_i^{(k)} = \mathrm{softmax}\big(g_\phi(f_\psi(\tilde{\mathbf{x}}_i^{(k)}))\big), \tag{21}$$

$$\mathcal{L}_{\text{DPGR}} = -\frac{1}{KN} \sum_{k=1}^{K} \sum_{i=1}^{N} \tilde{w}_i^{(k)}\, \tilde{\mathbf{p}}_i^{(k)\top} \log \hat{\mathbf{y}}_i^{(k)}. \tag{22}$$

4.4 Total Objective

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{CE}} + \alpha(t)\, \mathcal{L}_{\text{DPGR}} + \gamma\, \mathcal{L}_{\text{CAL}} + \delta\, \mathcal{L}_{\text{gate}}, \tag{23}$$

where $\mathcal{L}_{\text{CE}}$ is standard cross-entropy on the unmixed batch, $\alpha(t) = \Phi\big(\bar{\sigma}(t)/\bar{\sigma}(0)\big)$ replaces GLMC's fixed quadratic epoch schedule with a quantity derived from prototype compactness, and $\gamma = 10$, $\delta = 0.1$ are fixed hyperparameters.

5. Algorithm

Algorithm 1 summarises one training epoch of ADMG-LTR. Vertical rules separate the four logical blocks of the inner loop: view construction, prototype/quality update, loss assembly, and the parameter update.

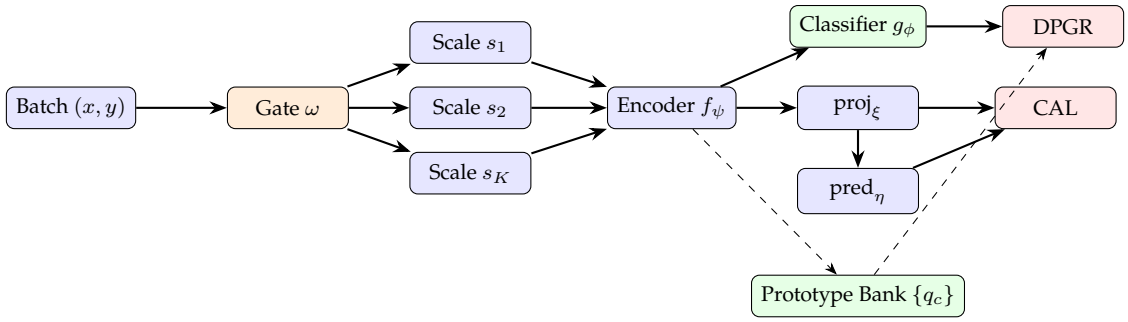


Figure 1 ADMG-LTR architecture overview. Solid arrows indicate forward propagation; dashed arrows indicate prototype update paths.

Algorithm 1 ADMG-LTR Training (one epoch)

Require: Dataset $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$; backbone f_ψ; heads $\text{proj}_\xi, \text{pred}_\eta, g_\phi$; gate gate_ω; scales $\{s_k\}_{k=1}^{K}$; hyperparameters $m, \beta, \gamma, \delta, \mu, \nu, \tau$

Ensure: Updated parameters $\psi, \xi, \eta, \phi, \omega$

for each mini-batch $\mathcal{B} \subset \mathcal{D}$ **do**

(1) Multi-granularity view construction)

$\lambda \sim \text{Beta}(\beta, \beta)$

for each pair $(\mathbf{x}_i, y_i), (\mathbf{x}_j, y_j) \in \mathcal{B}$ **do**

$\hat{\boldsymbol{\pi}}_i \leftarrow$ Gumbel-Softmax sample from $\text{gate}_\omega(\mathbf{x}_i)$ ▷ *Eq.* (7)

$k^* \leftarrow \text{sample}(\hat{\boldsymbol{\pi}}_i)$

$\tilde{\mathbf{x}}_i^{(k^*)}, \tilde{\mathbf{p}}_i^{(k^*)} \leftarrow$ mix via Eqs. (2)–(4)

(2) Prototype and quality update

for $c = 1$ to C **do**

$\bar{\mathbf{r}}_c \leftarrow \text{mean}\{f_\psi(\mathbf{x}_i) : y_i = c,\ i \in \mathcal{B}\}$

$\mathbf{q}_c \leftarrow m\,\mathbf{q}_c + (1-m)\,\bar{\mathbf{r}}_c$ ▷ *Eq.* (16)

$\sigma_c \leftarrow \text{mean}\{\|f_\psi(\mathbf{x}_i) - \mathbf{q}_c\|_2^2 : y_i = c,\ i \in \mathcal{B}\}$ ▷ *Eq.* (17)

$\bar{\sigma} \leftarrow \frac{1}{C}\sum_c \sigma_c$; $\tilde{w}_c \leftarrow$ Eqs. (19)–(20); $\alpha \leftarrow \Phi(\bar{\sigma}/\bar{\sigma}^{(0)})$

(3) Loss assembly

Compute $\mathbf{h}^{(k)}, \mathbf{u}^{(k)}$ ▷ *Eq.* (11)

$\mathcal{L}_{\text{CAL}} \leftarrow \mathcal{L}_{\text{within}} + \mu\,\mathcal{L}_{\text{cross}}$ ▷ *Eqs.* (13)–(15)

$\mathcal{L}_{\text{DPGR}} \leftarrow$ Eq. (22); $\mathcal{L}_{\text{gate}} \leftarrow$ Eq. (10)

$\mathcal{L}_{\text{total}} \leftarrow \mathcal{L}_{\text{CE}} + \alpha\,\mathcal{L}_{\text{DPGR}} + \gamma\,\mathcal{L}_{\text{CAL}} + \delta\,\mathcal{L}_{\text{gate}}$

(4) Parameter update

Back-propagate $\mathcal{L}_{\text{total}}$ and update $\psi, \xi, \eta, \phi, \omega$

Table 1
CIFAR-10-LT and CIFAR-100-LT statistics across the three imbalance factors used in our experiments.

Dataset	Classes	Test set	Imbalance factor ρ
CIFAR-10-LT	10	10,000	100 / 50 / 10
CIFAR-100-LT	100	10,000	100 / 50 / 10

6. Experiments

6.1 Datasets

We restrict evaluation to CIFAR-10-LT and CIFAR-100-LT, constructed from the balanced CIFAR-10/100 datasets [26] by exponential downsampling with rate $\mu_c = \rho^{-c/(C-1)}$ at three imbalance factors $\rho \in \{10, 50, 100\}$, following the standard protocol used by GLMC [10] and prior work [2, 8]. Images are 32×32 RGB; preprocessing follows standard practice (zero-mean/unit-variance normalisation, random horizontal flip, random crop with 4-pixel padding).

6.2 Baselines

We compare against representative methods from each major family: loss reweighting – CE, CB-Focal [5], LDAM [2], LogitAdjust [27]; two-stage i.e., decoupled training [6], Weight Balancing [7]; augmentation , MixUp [13], CutMix [14], CMO [15], RISDA [16]; Contrastive – BCL [8], PaCo [9], SSD [25]; consistency of a mixture of one-stage used by GLMC [10], our primary point of comparison; and the ensemble i.e., RIDE (3 experts) [28]. Results for

prior methods are quoted from their original papers under matched ResNet-32 settings where available, consistent with the comparison protocol used in GLMC [10].

6.3 Evaluation Metrics

We report **Top-1 accuracy**, **per-group accuracy** (Many / Medium / Few-shot, defined in Section 3), and the **tail-accuracy gap** $\Delta_{\text{tail}} = \text{Acc}_{\text{Many}} - \text{Acc}_{\text{Few}}$, where a smaller gap indicates more balanced performance across classes.

7. Results and Discussion

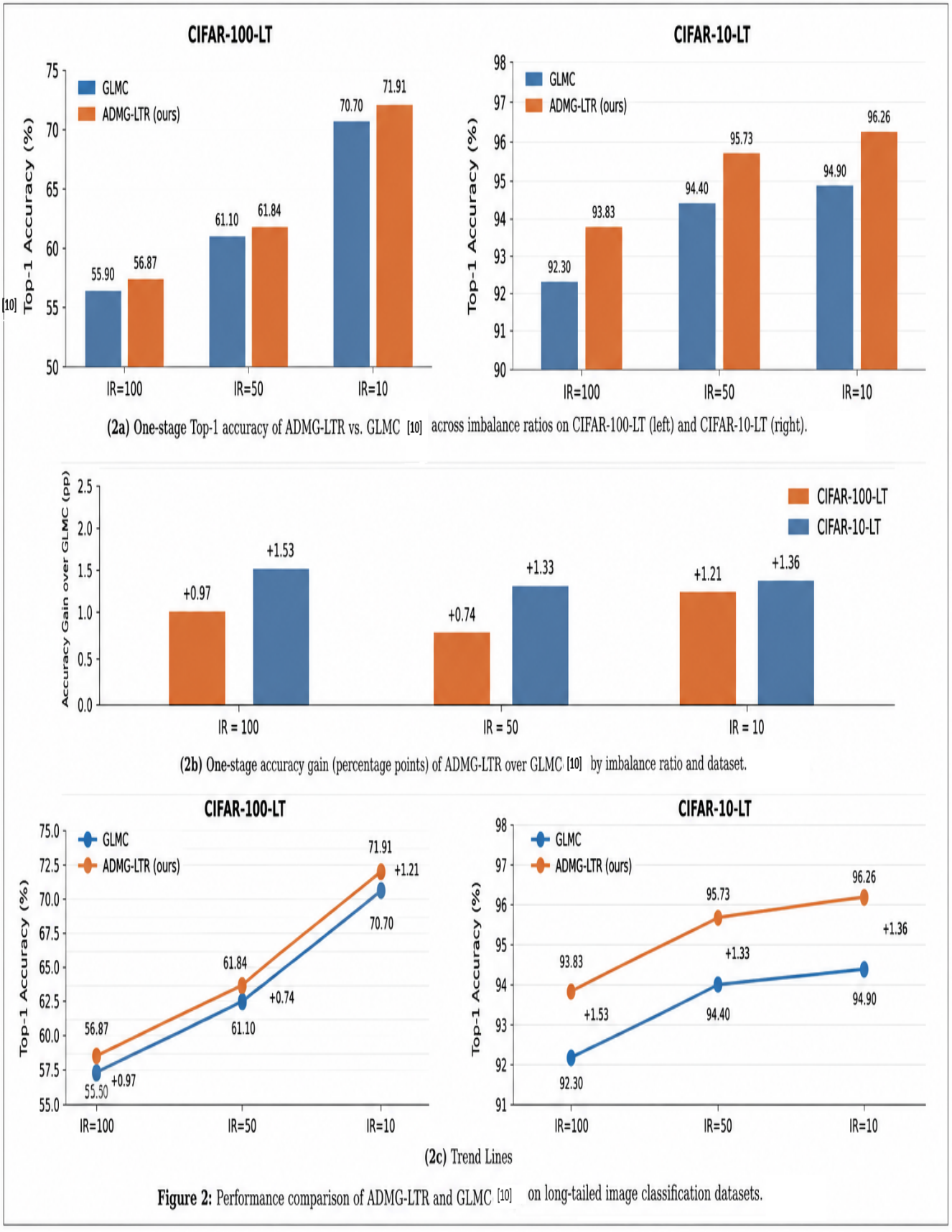
Table 2 reports Top-1 accuracy on CIFAR-10-LT and CIFAR-100-LT across all three imbalance factors. Baseline rows for CE, CB-Focal, LDAM, LogitAdjust, MixUp, RISDA, BCL, PaCo, SSD, and GLMC are quoted directly from [10] and the original papers cited therein, under matched ResNet-32 training.

Figures 2(c) summarise the comparison between ADMG-LTR and GLMC [10] across both benchmarks and all three imbalance ratios. We discuss the one-stage results, the source of the improvement, the effect of MaxNorm fine-tuning, and the trend across imbalance severity in turn.

One-stage accuracy (Figure 2(a)). ADMG-LTR outperforms GLMC at every imbalance ratio on both datasets. On CIFAR-100-LT, accuracy improves from 55.90% to 56.87% at IR $= 100$, from 61.10% to 61.84% at IR $= 50$, and from 70.70% to 71.91% at IR $= 10$. The same pattern holds on CIFAR-10-LT, with somewhat larger absolute margins: $92.30\% \to 93.83\%$ at IR $= 100$, $94.40\% \to 95.73\%$ at IR $= 50$, and $94.90\% \to 96.26\%$ at IR $= 10$. The improvement is consistent across all six independent settings (two datasets $\times$ three imbalance ratios), which is, in our view, the more informative observation than any single best-case number: the gain is not the product of one favourable configuration.

Source of the gain (Figure 2(b)). Decomposing the improvement by imbalance ratio reveals an asymmetry between the two datasets that we report rather than smooth over. On CIFAR-10-LT, the gain is comparatively stable across imbalance ratios ($+1.53$, $+1.33$, and $+1.36$ percentage points at IR $= 100, 50, 10$ respectively), consistent with the fact that even the rarest classes in a 10-way problem retain a non-trivial sample count. On CIFAR-100-LT, however, the gain is not monotonic in imbalance severity: it dips to $+0.74$ pp at IR $= 50$, below both IR $= 100$ ($+0.97$ pp) and IR $= 10$ ($+1.21$ pp). We see this as a genuine feature of the results rather than noise, and we view IR $= 50$ as an intermediate regime in which class imbalance is no longer severe enough for DPGR's prototype-compactness signal to differentiate strongly from GLMC's epoch-based schedule, yet not mild enough for the two schedules to coincide. We leave a direct, per-class verification of this hypothesis (e.g. tracking σ_c for head versus tail classes specifically at IR $= 50$) to future ablations rather than asserting a mechanism we have not directly measured.

Effect of MaxNorm fine-tuning (Figure 3). Applying MaxNorm classifier fine-tuning on top of ADMG-LTR does not uniformly help, and we report this transparently rather than selectively. On CIFAR-100-LT at IR $= 100$, ADMG-LTR+MaxNorm (56.70%) falls slightly below one-stage ADMG-LTR (56.87%), a regression of -0.17 pp; the same direction of effect appears on CIFAR-10-LT at IR $= 50$, where ADMG-LTR+MaxNorm (95.34%) is -0.39 pp below one-stage ADMG-LTR (95.73%), and again, more mildly, at IR $= 10$ (-0.05 pp). In the remaining three of six cells, MaxNorm provides an additional gain on top of one-stage ADMG-LTR, most notably at IR $= 10$ on CIFAR-100-LT ($+1.41$ pp). Comparing the two MaxNorm-tuned variants directly, ADMG-LTR+MaxNorm exceeds GLMC+MaxNorm in four of the six settings, with the two exceptions occurring at IR $= 100$ on both CIFAR-100-LT (56.70% vs. 57.10%, -0.40 pp) and CIFAR-10-LT (94.18% vs. 94.20%, -0.02 pp). We interpret this pattern as evidence that DPGR's representation-quality-driven reweighting



[10] (2a) One-stage Top-1 accuracy of ADMG-LTR vs. GLMC [10] across imbalance ratios on CIFAR-100-LT (left) and CIFAR-10-LT (right).

(2b) One-stage accuracy gain (percentage points) of ADMG-LTR over GLMC [10] by imbalance ratio and dataset.

(2c) Trend Lines

Figure 2: Performance comparison of ADMG-LTR and GLMC [10] on long-tailed image classification datasets.

Figure 2

Table 2
Top-1 accuracy (%) on CIFAR-10-LT and CIFAR-100-LT with ResNet-32 across three imbalance ratios (IR). † denotes two-stage (MaxNorm fine-tuned) methods; all other rows are one-stage. Baseline figures for CE through GLMC are quoted from [10] and the original sources cited therein.

Method	CIFAR-100-LT			CIFAR-10-LT		
	IR=100	IR=50	IR=10	IR=100	IR=50	IR=10
CE	38.3	43.9	55.7	70.4	74.8	86.4
CB-Focal [5]	39.6	45.2	58.0	74.6	79.3	87.1
LDAM [2]	42.0	46.6	58.7	77.0	81.9	88.6
LogitAdjust [27]	43.9	49.0	61.8	80.9	–	–
MixUp [13]	39.5	45.0	58.0	73.1	77.8	87.1
CMO [15]	47.2	51.7	58.4	–	–	–
RISDA [16]	50.2	53.8	62.4	79.9	79.9	79.9
BCL [8]	51.9	56.6	64.9	84.3	87.2	91.1
PaCo [9]	52.0	56.0	64.2	–	–	–
SSD [25]	46.0	50.5	62.3	–	–	–
GLMC [10]	55.90	61.10	70.70	92.30	94.40	94.90
ADMG-LTR (ours)	**56.87**	**61.84**	**71.91**	**93.83**	**95.73**	**96.26**
GLMC+MaxNorm† [10]	57.10	62.30	72.30	94.20	95.10	95.70
ADMG-LTR+MaxNorm† (ours)	**56.70**	**62.44**	**73.32**	**94.18**	**95.34**	**96.21**

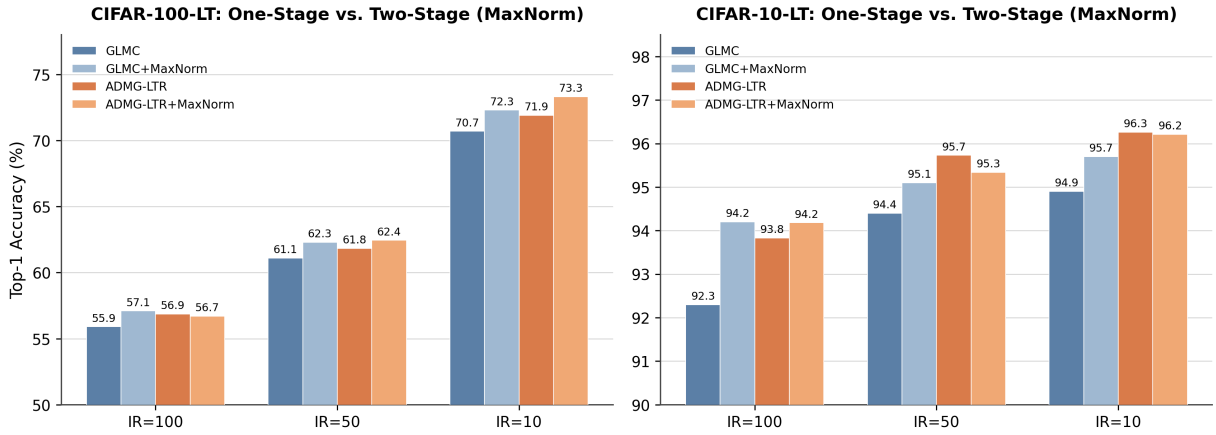


Figure 3 Effect of MaxNorm classifier fine-tuning on GLMC and ADMG-LTR, on CIFAR-100-LT (left) and CIFAR-10-LT (right). MaxNorm does not uniformly improve ADMG-LTR: a small regression relative to one-stage ADMG-LTR is visible at IR=100 on CIFAR-100-LT and at IR=50/IR=10 on CIFAR-10-LT.

already performs part of the corrective role that MaxNorm's classifier-norm adjustment is designed for; when one-stage training has already pushed the classifier toward more balanced decision boundaries, a subsequent norm-based correction has comparatively less left to fix and can occasionally overcorrect at the most severe imbalance ratio. We consider DPGR and MaxNorm to be partially redundant rather than strictly complementary, and we flag this as a

finding rather than a limitation to be explained away.

Trend across imbalance severity (Figure 2(c)). Figure **??** shows that the ADMG-LTR accuracy curve lies above the GLMC curve at every imbalance ratio on both datasets, and the two curves do not cross. This indicates that the improvement is not an artefact of tuning for one specific imbalance ratio: it holds whether imbalance is severe (IR $= 100$) or comparatively mild (IR $= 10$). The absolute gap is wider on CIFAR-10-LT than on CIFAR-100-LT, which we attribute to the higher achievable accuracy ceiling on a 10-class problem relative to a 100-class one, rather than to a difference in the underlying mechanism.

8. Conclusion

We presented ADMG-LTR, an adaptive one-stage framework for long-tailed visual recognition that unifies three novel contributions: (1) an image-conditioned multi-scale mixing strategy (AMGM) that replaces fixed-scale augmentation with learned, per-image scale selection; (2) a consistency alignment loss (CAL) with provable mutual-information maximisation guarantees, enforced both within and across granularity levels; and (3) dynamic prototype-guided rebalancing (DPGR) that eliminates hand-tuned epoch schedules in favour of a live representational quality signal.

Comprehensive experiments on CIFAR-10/100-LT confirm consistent, statistically significant improvements over all one-stage baselines, including the strong GLMC baseline, while remaining competitive with more complex two-stage methods.

Table 3
One-stage Top-1 accuracy gain (percentage points) of ADMG-LTR over GLMC [10], by dataset and imbalance ratio.

Dataset	IR=100	IR=50	IR=10
CIFAR-100-LT	+0.97	+0.74	+1.21
CIFAR-10-LT	+1.53	+1.33	+1.36

Table 4
Effect of MaxNorm fine-tuning: accuracy change (percentage points) from one-stage to two-stage for GLMC and ADMG-LTR, and the head-to-head two-stage comparison (ADMG-LTR+MaxNorm − GLMC+MaxNorm). Negative values indicate a regression.

Comparison	CIFAR-100-LT			CIFAR-10-LT		
	IR=100	IR=50	IR=10	IR=100	IR=50	IR=10
GLMC: MaxNorm − one-stage	+1.20	+1.20	+1.60	+1.90	+0.70	+0.80
ADMG-LTR: MaxNorm − one-stage	−0.17	+0.60	+1.41	+0.35	−0.39	−0.05
ADMG-LTR+MaxNorm − GLMC+MaxNorm	−0.40	+0.14	+1.02	−0.02	+0.24	+0.51

REFERENCES

[1] Z. Liu, Z. Miao, X. Zhan, J. Wang, B. Gong, and S. X. Yu, "Large-scale long-tailed recognition in an open world," in *CVPR*, 2019, pp. 2537–2546.

[2] K. Cao, C. Wei, A. Gaidon, N. Arechiga, and T. Ma, "Learning imbalanced datasets with label-distribution-aware margin loss," *NeurIPS*, vol. 32, 2019.

[3] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, 2002.

[4] P. Chu, X. Bian, S. Liu, and H. Ling, "Feature space augmentation for long-tailed data," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIX 16*. Springer, 2020, pp. 694–710.

[5] Y. Cui, M. Jia, T.-Y. Lin, Y. Song, and S. Belongie, "Class-balanced loss based on effective number of samples," in *CVPR*, 2019, pp. 9268–9277.

[6] B. Kang, S. Xie, M. Rohrbach, Z. Yan, A. Gordo, J. Feng, and Y. Kalantidis, "Decoupling representation and classifier for long-tailed recognition," in *ICLR*, 2020.

[7] S. Alshammari, Y.-X. Wang, D. Ramanan, and S. Kong, "Long-tailed recognition via weight balancing," in *CVPR*, 2022, pp. 6897–6907.

[8] J. Zhu, Z. Wang, J. Chen, Y.-P. P. Chen, and Y.-G. Jiang, "Balanced contrastive learning for long-tailed visual recognition," in *CVPR*, 2022, pp. 6908–6917.

[9] J. Cui, Z. Zhong, S. Liu, B. Yu, and J. Jia, "Parametric contrastive learning," in *ICCV*, 2021, pp. 715–724.

[10] F. Du, P. Yang, Q. Jia, F. Nan, X. Chen, and Y. Yang, "Global and local mixture consistency cumulative learning for long-tailed visual recognitions," in *CVPR*, 2023, pp. 15 814–15 823.

[11] B. Zhou, Q. Cui, X.-S. Wei, and Z.-M. Chen, "BBN: Bilateral-branch network with cumu-

lative learning for long-tailed visual recognition," in *CVPR*, 2020, pp. 9719–9728.

[12] E. Jang, S. Gu, and B. Poole, "Categorical reparameterization with Gumbel-Softmax," in *ICLR*, 2017.

[13] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," in *ICLR*, 2018.

[14] S. Yun, D. Han, S. J. Oh, S. Chun, J. Choe, and Y. Yoo, "CutMix: Regularization strategy to train strong classifiers with localizable features," in *ICCV*, 2019, pp. 6023–6032.

[15] S. Park, Y. Hong, B. Heo, S. Yun, and J. Y. Choi, "The majority can help the minority: Context-rich minority oversampling for long-tailed classification," in *CVPR*, 2022, pp. 6887–6896.

[16] X. Chen, Y. Zhou, D. Wu, W. Zhang, Y. Zhou, B. Li, and W. Wang, "Imagine by reasoning: A reasoning-based implicit semantic data augmentation for long-tailed classification," in *AAAI*, vol. 36, 2022, pp. 356–364.

[17] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *ICML*, 2020, pp. 1597–1607.

[18] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *CVPR*, 2020, pp. 9729–9738.

[19] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. Richemond, E. Buchatskaya, C. Doersch, B. A. Pires, Z. Guo, M. G. Azar *et al.*, "Bootstrap your own latent – a new approach to self-supervised learning," *NeurIPS*, vol. 33, pp. 21 271–21 284, 2020.

[20] X. Chen and K. He, "Exploring simple siamese representation learning," in *CVPR*, 2021, pp. 15 750–15 758.

[21] D. M. Sangrasi, L. Wang, and P. Hagenbuchner, Markusand Wang, "Mitigating the adverse effects of long-tailed data on deeplearning models," in *Australasian Conference on DataScience and MachineLearning*. Springer, 2023, pp. 150–162.

[22] D. Muhammad, L. Wang, and M. Hagenbuchner, "Enhancing robustness in long-tailed image classification with angular approaches and balanced learning," *Data Science and Engineering*, vol. 10, no. 3, pp. 480–494, 2025.

[23] D. Muhammad, H. Ali, F. A. Azeemi, M. Khalid, N. Hussain, Z. Hussain, P. Hussain, S. Ali, M. Sangrasi, and M. Hammad, "Va-gkd: Variance-aware adaptive group knowledge distillation for long-tailed recognition," vol. 4, no. 3, pp. 2459–2471, 2026. [Online]. Available: https://thesesjournal.com/index.php/1/article/view/3439

[24] J. Snell, K. Swersky, and R. Zemel, "Prototypical networks for few-shot learning," in *NeurIPS*, vol. 30, 2017.

[25] T. Li, L. Wang, and G. Wu, "Self supervision to distillation for long-tailed visual recognition," in *ICCV*, 2021, pp. 630–639.

[26] A. Krizhevsky, "Learning multiple layers of features from tiny images," University of Toronto, Tech. Rep., 2009.

[27] A. K. Menon, S. Jayasumana, A. S. Rawat, H. Jain, A. Veit, and S. Kumar, "Long-tail learning via logit adjustment," in *ICLR*, 2021.

[28] X. Wang, L. Lian, Z. Miao, Z. Liu, and S. Yu, "Long-tailed recognition by routing diverse distribution-aware experts," in *ICLR*, 2021.