AMDC: Adaptive Multi-Agent Data Curation for Robust Long-Tail Recognition

Shenghan Chen[1], **Yiming Liu**[1], **Yanzhen Wang**[1], **Xiankai Lu**[1], **Yawei Li**[2]

[1]Shandong University
[2]ETH Zürich

Abstract

Long-tail distributions severely hinder robust model generalization, leading to poor performance on minority classes and a trade-off where minority-class improvements degrade majority-class accuracy. Synthetic data augmentation can mitigate this imbalance, but its variable fidelity and randomness introduce a distribution shift between synthetic and real data. Conventional data augmentation techniques, relying on fixed rules, struggle with the dual challenges of data imbalance and randomness of generated samples. To address this problem, we propose Adaptive Multi-Agent Data Curation for Robust Long-Tail Recognition (AMDC). Our framework employs a multi-agent system that actively curates the training set by jointly optimizing the sample selection and the augmentation policy. Specifically, an active learning based filter selects the most informative synthetic samples, thereby mitigating the synthetic-real distribution shift. Meanwhile, LLM agents perceive the classifier's state to dynamically adapt augmentation policies, which effectively addresses the class-imbalance trade-off through distinguishing strategies to head and tail classes. This data curation framework establishes new state-of-the-art performance on CIFAR10-LT, CIFAR100-LT, and ImageNet-LT datasets, while demonstrating exceptional data efficiency. By strategically using just 4.8% of the full synthetic dataset, AMDC achieves 95% of the full-dataset performance and outperforms an equally sized random-sampling baseline by +5.4%. Extensive ablation studies further demonstrate the interpretability of the proposed method.

Introduction

Long-tail distributions in real-world data (Liu et al. 2019) pose a critical challenge for data-driven models. Deep learning architectures are inherently biased towards majority-class due to data imbalance, which severely degrades performance(He and Garcia 2009) and leads to suboptimal recognition of minority-class (Tian et al. 2024).

Existing long-tailed learning strategies include reweighting (Cao et al. 2019; Lin et al. 2017), architecture adjustments (Cui et al. 2023; Ma et al. 2022b) and resampling (Zhang et al. 2021; Shi et al. 2023; Chen and Su 2023) share a common focus on the data rebalancing or classifier elaboration. These approaches often demand manual design and face challenges in generalization ability across different data. Recently, research has shifted to improving the dataset itself by augmenting the training samples. For example, GIF (Ghosh et al. 2020) enriches limited datasets through synthetic sample generation via Stable Diffusion (SD). Meanwhile, DiffLT (Shao et al. 2024) utilizes the CBDM model to generate samples for tail classes. While these generative methods effectively increase data diversity for minority classes, they result in a large-scale and staticcomposite datasets combining real and synthetic samples (Karras, Laine, and Aila 2019). Meanwhile, applying fixed-rules, manual augmentation policies to these hybrid datasets has shown limited inefficiency and poor transferability (Zhang et al. 2020).

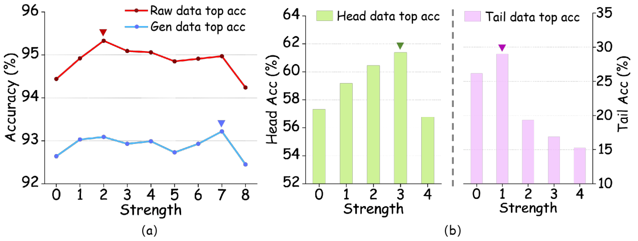

Figure 1: **Limitations of Traditional Data Augmentation.** (a) reveals that optimal data augmentation strength for real data (red line) and synthetic data (blue line) are different. (b) reveals that optimal data augmentation strength for head class (green histogram) and the tail class (pink histogram) are also different. These limitations call for an advanced dynamic augmentation strategy.

Specifically, traditional augmentation strategies fail to overcome two fundamental challenges (Cubuk et al. 2019) (See Fig. 1). **(1) Synthetic-Real Distribution Shift.** The inherent randomness of synthetic data drives their distribution away from that of real data (Karras, Laine, and Aila 2019). Figure 1(a) shows that synthetic and real data sources react unevenly to the same augmentation strategy, underscoring this distribution shift. **(2) Class-Imbalance Trade-off.** The classifier often struggles with a performance trade-off between head and tail classes. That is, optimal augmentation strategies for tail classes may inadvertently degrade the performance for head classes, and vice versa. As revealed by Figure 1(b), head and tail classes require distinct optimal augmentation strengths (Ahn, Ko, and Yun 2023). These observations call for an advanced adaptive augmentation strategy to replace the static one. *And dynamic augmentation becomes impera-*

tive to fully utilize the potential of generative data. To this end, we recast the optimal augmentation as a comprehensive *Data Curation* pipeline that (i) selects informative samples from the noisy synthetic samples, and (ii) applies class-aware, dynamic augmentation to them.

With this aim in mind, our Adaptive Multi-Agent Data Curation for Robust Long-Tail Recognition (AMDC) is designed to execute this pipeline. It consists of two components: an Adaptive Filter module (Filt) and a Multi-Agent Collaboration-based Strategy Generation module (MSG).

In particular, to obtain a compact subset from the massive synthetic data, we combine the classification score and similarity score as the filtering metric. Then, we perform data selection in an active learning manner. Then, based on the selected high quality subset, we design a Multi-Agent Collaboration system to enhance them. The MSG module performs several operations including *classifier state estimation*, *objective analysis*, *policy selection*, and *policy review*. With these collaborative operations, our MSG module realizes a dynamic data augmentation policy adapted to the training state of the classifier, surpassing fixed rule-based schedules such as AutoAugment (Cubuk et al. 2019) and strength tuning of predefined strategies as in CUDA (Ahn, Ko, and Yun 2023). A group of agents continuously monitors key training metrics including the training and validation loss and accuracy for both head and tail classes, and synthesizes class-specific augmentation decisions in real time. This feedback loop dynamically generates new augmentation plans for head and tail classes, thereby mitigating the performance trade-off between head and tail classes during training.

Our main contributions are four-fold:

- We pioneer a new paradigm for generative long-tail recognition with the Adaptive Multi-Agent Data Curation (AMDC) framework, which addresses the dual challenges of synthetic-and-real distribution shift and class-imbalance trade-off.
- We introduce an adaptive filtering module that uses active learning to select high-fidelity synthetic samples, effectively mitigating the synthetic-and-real distribution shift.
- We propose a MSG module, where LLM agents dynamically tailor differentiated augmentation policies for head and tail classes to address the class-imbalance trade-off.
- By using only a small subset of synthetic data, we achieve new state-of-the-art performance on CIFAR10-LT, CIFAR100-LT, and ImageNet-LT, showcasing our method's exceptional data efficiency.

Related Work

Long-tailed Learning and Class-Rebalancing. Most traditional long-tail learning methods include class rebalancing and model restructuring. Class rebalancing includes re-sampling (Zhao et al. 2025a) and re-weighting. Re-sampling strategies aim to achieve a balanced training dataset. They use over-sampling (Buda, Maki, and Mazurowski 2018) to increase the number of instances in tail classes, or under-sampling (He and Garcia 2009)to decrease the number of instances in head classes. However, these methods carry the risk of over-fitting tail classes and impairing model generalization. Re-weighting methods assign weights to the loss function of each class that are negatively correlated with its sample size, aiming to balance the gradient contributions of different classes (Jamal et al. 2020). However, inappropriate weights used in re-weighting methods can lead to problems such as under-fitting or over-fitting. Meanwhile, model restructuring often require careful, task and dataset-specific design(Cai et al. 2025; Narayan, Vs, and Patel 2025), which in turn leads to limited generalization (Ma et al. 2022b).

Data Augmentation (DA). DA is **basically** a cornerstone technique for improving model generalization. Its evolution spans from foundational strategies like Cutout (DeVries and Taylor 2017) and MixUp (Zhang et al. 2018) to the automation of policy discovery with search-based methods like AutoAugment (Cubuk et al. 2019) and really efficient randomized approaches such as RandAugment (Cubuk et al. 2020). To address the challenges of class-imbalanced learning, research has focused on more granular control, including applying differentiated policies for head and tail classes and using generative models to simply synthesize minority samples. The frontier of this field clearly explores dynamic strategies where the augmentation policy evolves during training. Prominent examples of such adaptive methods include Adversarial AutoAugment (Zhang et al. 2020) and Curriculum Augmentation (Soviany et al. 2022; Ahn, Ko, and Yun 2023), which actually adjust policies based on model-in-the-loop feedback or a pre-defined schedule, representing a shift towards state-aware enhancement techniques.

Generative Models and Synthetic Data. The initial research concentrated on leveraging generative adversarial networks (GANs) to generate novel image samples for tail classes, then facilitating the creation of a more balanced dataset for training classifiers (Mariani et al. 2018; Antoniou, Storkey, and Edwards 2017). Recently, with the advent of breakthroughs in image generation quality achieved by diffusion models (Ho, Jain, and Abbeel 2020; Nichol and Dhariwal 2021; Song et al. 2021), some studies have begun exploring the use of diffusion models to generate diverse tail classes for long-tail recognition tasks (Shao et al. 2024). However, the majority of these methods prioritize the generation process itself, while the subsequent enhancement of these newly generated, inherently random data often adopts a fixed strategy, failing to fully capitalize on the potential of synthetic data (Han et al. 2024; Rangwani et al. 2022b).

The Proposed Method

Preliminary and Motivation

For long-tail image recognition task, given a dataset $\mathcal{D} = \{(x_i, y_i), \cdots, (x_N, y_N)\}$, x_i representing an input image and y_i representing the label, with totally N sample and M classes (the set of all classes is C). For each class c_j in C, it has $|c_1| \geq |c_2| \geq ... \geq |c_M|$, where $|c_j|$ denotes the number of samples in class c_j, and the ratio $r = \frac{|c_1|}{|c_M|}$ is defined as the long-tail ratio. Due to the long-tail nature, a few classes possess many samples while other classes only have a few samples, which leads to a class imbalance issue and low

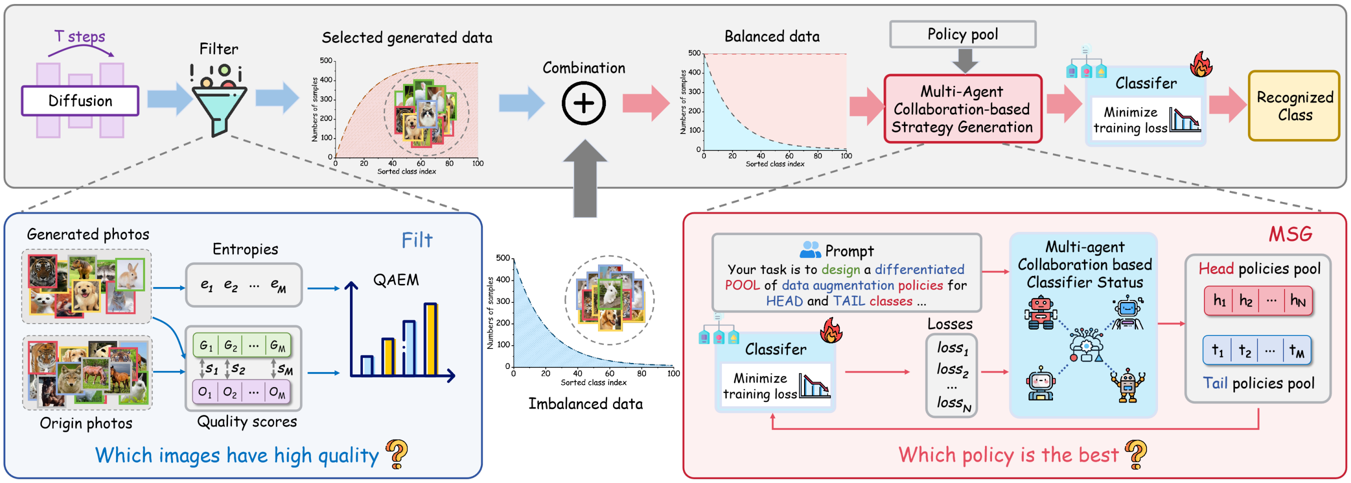

Figure 2: **The overall framework of the proposed AMDC.** AMDC consists of two core components: Active Filter (Filt) module and Multi-Agent Collaboration-based Strategy Generation (MSG) module. In MSG, distinct policy pools are curated for head and tail classes.

precision of the classifier. Thus, long-tail classification aims to learn a function f_{LT} to predict each test sample correctly.

The Randomness Nature of Synthetic Data. Mainstream long-tail recognition methods generally utilize synthetic data to augment their training sets, which is an inherently stochastic process. This randomness is truly fundamental to creating novel and diverse samples. However, it also introduces uncertainty, thereby posing a significant challenge to the subsequent use of synthetic samples.

Taking the denoising diffusion probabilistic model as an example, this model learns to reverse a fixed forward process that gradually adds Gaussian noise to data (Han et al. 2024; Karras et al. 2022). The generation process starts with pure Gaussian noise $\mathbf{x_t} \sim \mathcal{N}(0, \mathbf{I})$, and iteratively refines it by reversing the process over T timesteps. Each step of this reverse trajectory, parameterized by θ as $p_\theta(x_{t-1}|x_t)$, introduces new randomness:

$$\mathbf{x}_{t-1} = \mu_\theta(\mathbf{x}_t, t) + \sigma_t \mathbf{z}, \qquad (1)$$

where $\mathbf{z} \sim \mathcal{N}(0, \mathbf{I})$. The randomness in diffusion models is inherently sequential and constantly accumulated throughout the iterative denoising trajectory.

The strategic dichotomy of head and tail classes. The existing LT solutions often seek a single, fixed optimal augmentation policy $\mathcal{P}$ for each dataset which can be formulated as minimizing the expected loss:

$$\min_\theta \mathbb{E}_{(x,y)\sim\mathcal{D}} \left[\mathcal{L}\left(f_\theta(\mathcal{P}(x)), y\right)\right], \qquad (2)$$

where f_θ means the LT model f_{LT} and $\mathcal{D}$ is the data distribution. Here, the conventional paradigm heavily relies on the assumption that an elaborated data augmentation policy $\mathcal{P}$ which is predefined for all classes.

However, this assumption is challenged by a phenomenon observed in prior experiment: generated and tail-class data are sensitive to the augmentation strength compared to original and head-class data (Ahn, Ko, and Yun 2023).

This differing sensitivity directly fuels the class-imbalance trade-off, where enhancing tail-class performance often compromises head-class results, and vice versa. A fixed-rules augmentation fails to meet the distinct needs of robust head-classes and fragile tail-classes.

Thus, we propose a optimal augmentation policy that incorporates the augmentation strategy into the model learning:

$$\min_{\theta, \mathcal{P}_{head}, \mathcal{P}_{tail}} \mathbb{E}_{(x,y)\sim\mathcal{D}}[\mathcal{L}(f_\theta(\mathcal{P}(x, y)), y)], \qquad (3)$$

where the applied augmentation $\mathcal{P}(x, y)$ is conditionally chosen based on the class label y:

$$\mathcal{P}(x, y) = \begin{cases} \mathcal{P}_{head}(x) & \text{if } y \in \mathcal{C}_{head} \\ \mathcal{P}_{tail}(x) & \text{if } y \in \mathcal{C}_{tail}. \end{cases} \qquad (4)$$

Overall Framework

As shown in Figure 2, we propose a novel LT recognition paradigm called AMDC. Specifically, our AMDC framework consists of an adaptive filter and a multi-agent collaboration-based strategy generation. The workflow is as follows: (1) **Select data:** Given images generated via Diffusion Model, the adaptive filter scores each synthetic sample by computing the Quality Score (S) and Entropy (H), which fuse into a final Quality-Aware Entropy Metric (QAEM) score. This enables the filter to select only the valuable samples (*i.e.* compact subset) for the training set. (2) **Generate dynamical augmentation policy:** Given a text prompt containing a search space of different augmentation policies and classifier state, our multi-agent collaboration-based strategy generation module iteratively refines the augmentation policy pools based on classifier performance feedback during training, ensuring an optimal augmentation strategy is applied to head and tail classes. (3) **Train the classifier:** Due to reusing computation in policy selection and review with classifier retraining, computing cost and time overhead are extremely reduced.

Active Filtering

Given a generated image $(x', y') \in \mathcal{D}_{\text{gen}}$ from the Diffusion Model, we quantify its similarity to real data by finding the nearest neighbor within the training set. Specifically, for each generated image, we compute the L_2 distance to find the closest images in the training set $\mathcal{D}$ with the same label y':

$$d(x', y') = \min_{(x,y)\in\mathcal{D}, y=y'} \|x - x'\|_2. \quad (5)$$

A lower value of $d(x', y')$ suggests higher fidelity of the generated image to the real data distribution.

To measure and select high-quality generated samples, we normalize distance values in Eq. 5:

$$d_{\text{norm}} = \frac{d(x') - \min(d_*)}{\max(d_*) - \min(d_*)}, \quad S = 1 - d_{\text{norm}}, \quad (6)$$

where d_* is the set of all computed minimum distances for the generated samples. The value of d_{norm} is then scaled to the range $[0, 1]$. Consequently, S denotes the quality score, where a higher value of S directly signifies that the generated image is highly similar to a real training sample, thus indicating it has a high perceived quality.

However, we argue that these highly similar samples may not bring the greatest benefit to the classifier (Chen et al. 2009; Ren et al. 2018). To mitigate this problem, drawing inspiration from Active Learning (Settles 2009), we use the classifier's prediction entropy to measure the value of the generated data. The entropy reflects the uncertainty in the classifier's prediction for a given sample:

$$H(p(x')) = -\sum_i p_i(x') \log p_i(x'), \quad (7)$$

where $p(x')$ is the classifier's predicted probability vector for the generated sample x', and $p_i(x')$ is the predicted probability for the i-th class. A higher entropy score indicates that the classifier is uncertain about the sample's class (*i.e.*, hard sample) and can be more valuable for classifier training.

Finally, we take both image quality score (Eq. 6) and entropy (Eq. 7) into consideration and define the QAEM:

$$\text{QAEM}(x', y') = S(x', y')^{\gamma} \cdot H(p(x')), \quad (8)$$

where the hyperparameter γ is used to balance the weight between image quality S and entropy H to selection criterion. With the QAEM score, we can filter out generated samples and construct a compact and high-quality training dataset.

Multi-Agent Collaboration-based Strategy Generation

Besides the data quality, it is meaningful to learn an adaptive data augmentation strategy for head and tail classes.

Inspired by the Chain-of-Thought (CoT) capability of LLMs, we propose a MSG module. It divides the augmentation curation process into multiple sub-stages and constructs corresponding specialized agents for each. Within the system, each agent collaborates and interacts dynamically while strictly performing its specialized tasks. This dynamic interaction ensures that LLMs accurately diagnose the training state and generate appropriate policies, providing significant advantages over traditional monolithic agent reasoning.

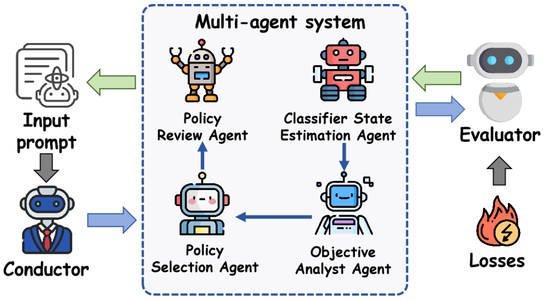

Figure 3: **The pipeline of Multi-Agent Collaboration based Classifier State.** The blue and green arrows indicate forward CoT reasoning and backward feedback processes, and the training and validation losses across multiple epochs are provided by the training network.

Multi-Agent Construction As shown in Figure 3, the constructed Multi-Agent Collab- oration system consists of four agents that aim to address the different augmentation requirements of head and tail classes:

(1) The **Classifier State Estimation Agent** works for the diagnosis of the classifier's overall training state, a task facilitated by the loss history and acc history.

(2) The **Objective Analyst Agent** is tasked with formulating differentiated Augmentation objectives for head and tail classes, fully considering the inherent performance tradeoff between them.

(3) The **Policy Selection Agent** works for selecting different data augmentation policies for the head and tail classes based on objectives.

(4) The **Policy Review Agent** checks the first three steps and ensures the output's format is correct.

The objective of MSG is to generate two policy pools of head classes and tail classes for the subsequent training stage. We construct a director D and an evaluator E to facilitate the coordination of multiple agents. Specifically, the director dynamically guides different agents to iteratively construct the forward CoT reasoning. Subsequently, the evaluator facilitates the backward feedback and utilizes feedback signals to refine the policy pools.

Augmentation Strategy Optimization After constructing the multi-agent system, we perform a deterministic forward pass with a conditionally-triggered dynamic feedback loop to ensure the effectiveness of the augmentation strategy by continuously assessing the classifier's state and adapting the policies accordingly.

Forward CoT Reasoning. The initial input for Forward pass is the pre-defined prompt $\mathcal{P}_p$. Formally, the sequence of agents involved in a forward inference process is defined as:

$$\xi = \{\mathcal{A}_{\delta_1}, \mathcal{A}_{\delta_2}, \dots, \mathcal{A}_{\delta_n}\}, \quad (9)$$

where n is the total number of agents in this sequence and δ_i is the configuration of the i-th agent.

Under the CoT reasoning framework, the set of output for the t-th inference step is denoted as $\mathcal{O}_t$ and the state is denoted as $\mathcal{S}_t = (\mathcal{P}_p, \mathcal{O}_t, t)$. Thus, the sequential forward

CoT reasoning is denoted as:

$$\mathcal{O}_t = \mathcal{A}_{\delta_t}\Big(\mathcal{S}_0, \mathcal{O}_1 = \mathcal{A}_{\delta_1}(\mathcal{S}_0), \ldots, \mathcal{O}_{t-1} = \mathcal{A}_{\delta_{t-1}}(\mathcal{S}_{t-2})\Big). \tag{10}$$

In a forward pass (See Fig. 3), the reasoning chain proceeds through four specialized agents: (1) The Classifier State Estimation Agent analyzes the training and validation loss of head classes and tail classes to output a diagnostic summary; (2) Then, the objective analysis Agent translates this summary into differentiated objectives for head and tail classes; (3) The Policy Selection Agent constructs augmentation policies for head and tail classes based on these objectives; (4) Finally, the Policy Review Agent validates and generates final augmentation policy pools.

Backward Feedback. After that, the backward feedback process (See Fig. 3) is organized by a Director (D) and an Evaluator (E) through an iterative pipeline, which enables a multi-agent system to adjust collaborative behavior by using the evaluator E to evaluate the forward response. Specifically, the result in the forward decision (Eq. 10) is defined as $\mathcal{F}$ (*i.e.*, the selected data augmentation policy pools). The backward feedback process begins with an external feedback f_{raw} usually provided by the target network, which is a classifier in our task, denoted as:

$$f_{raw} = f_{LT}(\mathcal{F}), \tag{11}$$

Via Eq. 11, we can obtain a raw feedback f_{raw}. Based on the feedback and the augmentation policy pools ($\mathcal{F}$), we then use evaluator E to compute the results to evaluate the augmentation policy pools:

$$(f_{eval}, f_{station}) = E(f_{raw}, \mathcal{F}), \tag{12}$$

where f_{eval} indicates whether new augmentation policy pools need to be re-inferred and $f_{station}$ specifies the starting inference step for this re-inference process.

Policy Pool Refinement. Based on the value of f_{eval}, the policy pool would be updated:

$$\mathcal{F}' = \begin{cases} \mathcal{F}, & \text{if } f_{\text{eval}} = \text{True} \\ \Pi_{f_{\text{station}} \to \text{final}}(\mathcal{S}_{f_{station-1}}), & \text{if } f_{\text{eval}} = \text{False} \end{cases} \tag{13}$$

If the f_{eval} of the evaluation is true, the augmentation policy pools $\mathcal{F}$ can be kept, and the next forward CoT decision process is skipped. Otherwise, the evaluator gives a new prompt to the director D.

Then MSG will perform the next forward CoT decision and get a new result $\mathcal{F}'$ which is a new augmentation policy pools. This iterative refinement cycle continues until a policy is approved or a maximum number of attempts N_{max} is reached. To make our algorithm clear and easy to understand, we show the pseudo-code of our algorithm in Alg. 1.

Agent Fine-Tuning with GRPO To further enhance the performance of the Policy Selector Agent, we employ Group Policy Optimization (GRPO) algorithm (DeepSeek-AI et al. 2025) to finetune this agent.

Algorithm 1: Multi-Agent Strategy Generation

```
Input: External feedback $f_{raw}$, State of the $t$-th inference step $\mathcal{S}_t$, Evaluation Agent $E$, Max retries $N_{max}$.
Initialize: Initialize the inference step $t = 1$.
for $retry \leftarrow 1$ to $N_{max}$ do
    $\mathcal{F} \leftarrow$ ForwardPipeline($\mathcal{S}_{t-1}$)
    $(f_{eval}, f_{station}) \leftarrow E(f_{raw}, \mathcal{F})$
    if $f_{eval}$ is true then
        return $\mathcal{F}$
    end if
    $t \leftarrow f_{station}$
end for
Output: Final approved policy pools $(P_{head}, P_{tail})$.
```

The optimization is guided by a composite reward:

$$\mathcal{R}_{\text{total}} = \mathcal{R}_{\text{loss}} + \mathcal{R}_{\text{format}}, \tag{14}$$

which combines two distinct signals. A loss-based reward R_{loss} is designed to compute the difference between instantaneous training loss $\mathcal{L}_t$ and an Exponential Moving Average (EMA) value across multiple epochs (L_{baseline}):

$$\mathcal{R}_{\text{loss}} = \mathcal{L}_t - \mathcal{L}_{\text{baseline}}^{(t)}, \tag{15}$$

$$\mathcal{L}_{\text{baseline}}^{(t)} = \alpha \mathcal{L}_t + (1-\alpha)\mathcal{L}_{\text{baseline}}^{(t-1)}, \tag{16}$$

where α is a factor. Eq. 16 encourages the Policy Selector agent to explore more favorable augmentation policy rather than exploiting the current policy.

Meanwhile, a format reward ($\mathcal{R}_{\text{format}}$) acts as a binary penalty that is responsible for ensuring the agent's output strictly adheres to a predefined structure using `<think>` and `<answer>` tags in its entirety.

$$\mathcal{R}_{\text{format}} = \begin{cases} 0, & \text{if format is correct} \\ 1, & \text{if format is incorrect.} \end{cases} \tag{17}$$

The GRPO fine-tuning process operates on these rewards by comparing multiple candidate policies in a group. At each step, we sample a group of M policies and evaluate each using our total reward function ($\mathcal{R}_{\text{total}}$ in Eq. 14) to get a set of scores $\{\mathcal{R}_1, \ldots, \mathcal{R}_M\}$. These scores are then normalized to calculate a relative advantage $\mathcal{A}_i$ for policy updating, reflecting the quality compared to the group average:

$$\mathcal{A}_i = \frac{\mathcal{R}_i - \text{mean}(\{\mathcal{R}_1, \ldots, \mathcal{R}_M\})}{\text{std}(\{\mathcal{R}_1, \ldots, \mathcal{R}_M\})}. \tag{18}$$

Experiments

Experiment Setup

Datasets. AMDC is evaluated on three long-tailed benchmarks: CIFAR10-LT (Cao et al. 2019), CIFAR100-LT (Cao et al. 2019) and ImageNet-LT (Liu et al. 2019).

Metrics. Following standard evaluation protocols, we report Top-1 accuracy for comparison with state-of-the-art methods. To evaluate AMDC's effectiveness across varying imbalance levels, we report Top-1 accuracy under three imbalance ratios $r \in \{100, 50, 10\}$ for CIFAR10-LT and CIFAR100-LT. Additionally, we report results for "Many" (classes with over 100

Method	Backbone	CIFAR100-LT			CIFAR10-LT			**Statistic (IR 100)**		
		r=100	r=50	r=10	r=100	r=50	r=10	Many	Med.	Few
CE	ResNet-32	38.3	43.9	55.7	70.4	74.8	86.4	65.2	37.1	9.1
Focal Loss (Lin et al. 2017)	ResNet-32	38.4	44.3	55.8	70.4	76.7	86.7	65.3	38.4	8.1
LDAM-DRW (Cao et al. 2019)	ResNet-32	42.0	46.6	58.7	77.0	81.0	88.2	61.5	41.7	20.2
cRT (Kang et al. 2020)	ResNet-32	42.3	46.8	58.1	75.7	80.4	88.3	64.0	44.8	18.1
BBN (Zhou et al. 2020)	ResNet-32	42.6	47.0	59.1	79.8	82.2	88.3	-	-	-
RIDE (3 experts) (Wang et al. 2021)	ResNet-32	48.0	-	-	-	-	-	68.1	49.2	23.9
CAM-BS (Zhang et al. 2021)	ResNet-32	41.7	46.0	-	75.4	81.4	-	-	-	-
DiVE (He, Wu, and Wei 2021)	ResNet-32	45.4	51.1	62.0	-	-	-	-	-	-
SAM (Rangwani et al. 2022a)	ResNet-32	45.4	-	-	81.9	-	-	64.4	46.2	20.8
CUDA (Ahn, Ko, and Yun 2023)	ResNet-32	47.6	51.1	58.4	-	-	-	67.3	50.4	21.4
ADRW (Wang et al. 2023)	ResNet-32	46.4	-	61.9	83.6	-	90.3	-	-	-
H2T (Li et al. 2024)	ResNet-32	48.9	53.8	-	-	-	-	-	-	-
DiffuLT (Shao et al. 2024)	ResNet-32	51.5	56.3	63.8	84.7	86.9	90.7	69.0	51.6	29.7
DiffuLT + BBN (Shao et al. 2024)	ResNet-32	51.9	56.7	64.0	85.0	87.2	90.9	69.5	51.9	30.2
DiffuLT + RIDE (3 experts) (Shao et al. 2024)	ResNet-32	52.4	56.9	64.2	85.3	87.3	90.9	70.3	52.1	30.7
AMDC* (Ours)	ResNet-32	**69.9**	**70.3**	**70.6**	**92.9**	**93.2**	**93.5**	**70.9**	**69.2**	**68.3**
BALLAD* (Ma et al. 2022a)	ViT-B/16	77.8	-	-	-	-	-	84.9	79.7	67.3
VL-LTR* (Tian et al. 2022)	ViT-B/16	-	-	-	-	-	-	-	-	-
LPT* (Dong et al. 2022)	ViT-B/16	89.1	-	-	-	-	-	-	-	-
AMDC* (Ours)	ViT-B/16	**89.3**	-	-	-	-	-	**89.5**	**89.2**	**88.9**

Table 1: Results on CIFAR100-LT and CIFAR10-LT datasets. The imbalance ratio r is set to 100, 50 and 10. Methods marked with $*$ utilize external pre-training data. The highest-performing results are in bold, with the second-best in underline. Additionally, we present the results for different groups ("Many", "Med." and "Few") in CIFAR100-LT with r = 100.

samples), "Med." (classes with 20 to 100 samples), and "Few" (classes with fewer than 20 samples) categories separately to enable in-depth analysis. For ImageNet-LT, we present results with different feature backbones to further verify the effectiveness of AMDC.

Implementation. We employ Qwen2.5-7B- Instruct (Yang et al. 2024) to construct multiple agents in MSG. For reinforcement learning (RL) training, we utilize the Swift framework (Zhao et al. 2025b), where the EMA baseline factor α is set to 0.6. The pretrained diffusion models in Filt consist of EDM (Karras et al. 2022) and Imagen (Saharia et al. 2022). For CIFAR10/100-LT, we create a candidate pool of 100,000 generated images. We use ResNet-32 as the classifier backbone with cross-entropy loss. For ImageNet-LT, we generate a pool of 5,000 candidates for each class. The classifiers were based on ResNet-50 with cross-entropy loss function.

Main Results

CIFAR100-LT and CIFAR10-LT (Cao et al. 2019). As shown in Table 1, our proposed AMDC surpasses recent state-of-the-art (SOTA) approaches across multiple datasets under various imbalance ratios. Notably, under the most challenging imbalance scenario (r=100), AMDC achieves significant improvements over state-of-the-art methods: **+31.6**% on CIFAR100-LT and **+22.5**% on CIFAR10-LT compared to CE (*i.e.,* the baseline method with the naive Cross Entropy loss). Notably, we achieve this by using only **4.8**% of a full synthetic dataset (48,000 images), rather than relying on massive external data (1,000,000 images). This compact subset verifies the important role of Active Filtering in data curation.

Method	Backbone	All	Many	Med.	Few
CE	ResNet50	41.6	64.0	33.8	5.8
Focal Loss (Lin et al. 2017)	ResNet50	-	-	-	-
OLTR (Liu et al. 2019)	ResNet50	-	-	-	-
cRT (Kang et al. 2020)	ResNet50	47.3	58.8	44.0	26.1
RIDE (3 experts) (Wang et al. 2021)	ResNet50	54.9	66.2	51.7	34.9
SAM (Rangwani et al. 2022a)	ResNet50	62.0	52.1	34.8	-
CUDA (Ahn, Ko, and Yun 2023)	ResNet50	51.4	63.1	48.0	31.1
ADRW (Wang et al. 2023)	ResNet50	54.1	62.9	52.6	37.1
DiffuLT (Shao et al. 2024)	ResNet50	56.4	63.3	55.6	39.4
DiffuLT + RIDE (3 experts) (Shao et al. 2024)	ResNet50	56.9	64.1	55.8	39.9
BALLAD* (Ma et al. 2022a)	ResNet50	67.2	71.0	66.3	**59.5**
AMDC* (Ours)	ResNet50	**71.8**	**75.2**	**70.6**	59.1
BALLAD* (Ma et al. 2022a)	ViT-B/16	75.7	79.1	74.5	69.8
VL-LTR* (Tian et al. 2022)	ViT-B/16	77.2	84.5	74.6	59.3
LPT* (Dong et al. 2022)	ViT-B/16	-	-	-	-
AMDC* (Ours)	ViT-B/16	**78.2**	**84.9**	**75.6**	**69.0**

Table 2: Results on ImageNet-LT. Methods marked with $*$ utilize external pre-training data. The best results are in **bold**.

Furthermore, AMDC demonstrates balanced performance gains across all three frequency categories using two different backbones ("Many": 70.9/89.5, "Med.": 69.2/89.2, "Few": 68.3/88.9), effectively addressing the performance trade-off between head and tail classes in long-tailed learning. These results collectively demonstrate AMDC's effectiveness and generalization in handling both extreme class imbalance and the fundamental challenges of long-tailed distributions.

ImageNet-LT (Liu et al. 2019). ImageNet-LT is more challenging as the imbalance factor increases, thus the long-tail distribution degrades the compared methods' performance significantly. Thus, we evaluate AMDC on ImageNet-LT (Liu et al. 2019), featuring more complex data distributions. As demonstrated in Table 2, AMDC achieves significant gains in Top-1 accuracy across all classes (ResNet-50: 67.2% → 71.8%, ViT-B/16: 77.2% → 78.2%), highlighting its effectiveness in mitigating the adverse effects of long-tailed distributions. Again, these gains come from using just 20% of a full synthetic dataset.

Moreover, AMDC yields better performance than all the counterparts on tailed classes while not degrading the performance on head classes. These results comprehensively verify the superiority of AMDC in long-tailed challenges than prior SOTA methods. More experimental results are provided in the supplementary material.

Ablation Experiments

To evaluate out algorithm designs insights, we carry out comprehensive ablation studies on the CIFAR100-LT dataset with an imbalance ratio of $r = 100$, employing ResNet-32 as the classifier backbone. Other settings are consistent with those in the main experiments unless otherwise specified.

Different Modules in the Pipeline. To demonstrate the superiority of AMDC, we first study the importance of each component in Table 3a. We use the CE loss as a vanilla baseline. The CE+Diff configuration serves as a baseline that randomly filters generated data to complement the long-tailed dataset. Our results indicate that applying the multi-agent augmentation strategy improves 4.0% accuracy on CE baseline and 1.1% accuracy on CE+Diff baseline. Furthermore, for the generated data, using our Filter to select valuable samples can improve 4.3% accuracy. This result suggests that both modules are beneficial for achieving superior results.

Method	Acc (%)↑
CE	38.4
CE + MSG	42.4
CE + Diff	64.5
CE + Diff + MSG	65.6
CE + Diff + MSG + Filt	**69.9**

(a) Study on different modules in our pipeline.

γ	Acc (%)↑
1.0	68.3
1.5	68.4
2.0	68.9
2.5	**69.9**
3.0	69.5

(b) Study on hyperparameter γ.

Table 3: Ablation studies on different modules in our pipeline and hyperparameters.

The Importance of Each Agent. We conducted a detailed ablation study to quantify the contribution of different agent combinations within MSG module. The baseline, which utilized no dynamic agent-driven policies, achieved an Overall Accuracy of 69.1%. When the core Policy Selection Agent and Policy Review Agent were employed, the recognition performance notably improved to 69.4%. Further inclusion of the Classifier State Estimation Agent alongside the Policy Selection Agent and Policy Review Agent boosted the performance to 69.7%, demonstrating the benefit of dynamic performance feedback. With the utilization of all agents, the model achieved an Overall Accuracy of 69.9%, highlighting the advantage of MSG.

Classifier State Estimation Agent	Objective Analyst Agent	Policy Selection Agent	Policy Review Agent	Acc. (%)↑
✗	✗	✗	✗	69.1
✗	✗	✓	✓	69.4
✓	✗	✓	✓	69.7
✗	✓	✓	✓	69.3
✓	✓	✓	✓	**69.9**

Table 4: Ablation Study on MSG Agents.

Hyperparameter Sensitivity Analysis of Filts. We also performed an ablation study on the hyperparameter γ in the filter. The results in Tab. 3b show that γ achieves the best performance when set to 2.5, reaching an accuracy of 69.9%.

Method	Overall Acc. (%)↑	Head Acc. (%)↑	Tail Acc. (%)↑
All fixed	38.7±0.4	56.7±0.4	19.4±0.4
MSG in Head	41.1±0.3	**60.3±0.4**	21.2±0.3
MSG in Tail	41.4±0.3	59.2±0.3	23.1±0.4
Full MSG	**42.4±0.5**	59.6±0.4	**25.1±0.4**

Table 5: Comparative analysis of the MSG framework: exploring dynamic augmentation policy in terms of head classes, tail classes and overall performance.

Ablation Study on MSG Framework. We further conduct an ablation study for the MSG model by removing the active filtering module. On CIFAR100-LT, a fixed augmentation policy on CE classifier works as the baseline (Table 5). Compared to the baseline, applying MSG to the head and tail classes indepedently can also brings significantly performance promotion (head class: + 2.4%, + 3.6%, + 1.8%; tail class: + 2.7%, + 2.5%, + 3.7%). Notably, our full model (the last row) yields the best performance for tail classes as well as the overall accuracy.

Conclusion

In this paper, we studied the importance of data augmentation to LT performance and proposed Adaptive Multi-Agent Data Curation. Here, we introduced three key innovations: (1) we formulated the data augmentation in long-tailed recognition as a dynamic optimization process. (2) We propose a filter module that uses a Quality-aware Entropy metric to select the most valuable data. (3) We design a Multi-Agent Collaboration-based strategy generation to fully analyze the condition of classifier then generate a new policy pool for different data subgroups. Extensive experiments on multiple LT datasets have demonstrated the excellent recognition performance of our approach.

References

Ahn, S.; Ko, J.; and Yun, S.-Y. 2023. CUDA: Curriculum of Data Augmentation for Long-tailed Recognition. In *The Eleventh International Conference on Learning Representations*.

Antoniou, A.; Storkey, A.; and Edwards, H. 2017. Data Augmentation Generative Adversarial Networks. *arXiv preprint arXiv:1711.04340*.

Buda, M.; Maki, A.; and Mazurowski, M. A. 2018. A Systematic Study of the Class Imbalance Problem in Convolutional Neural Networks. *Neural networks*, 106: 249–259.

Cai, C.; Yang, L.; Chen, K.; Yang, F.; and Li, X. 2025. Long-Tailed Distribution-Aware Router for Mixture-of-Experts in Large Vision-Language Model. *arXiv preprint arXiv:2507.01351*.

Cao, K.; Wei, C.; Gaidon, A.; Arechiga, N.; and Ma, T. 2019. Learning Imbalanced Datasets with Label-Distribution-Aware Margin Loss. *Advances in Neural Information Processing Systems*, 32.

Chen, J.; and Su, B. 2023. Transfer Knowledge From Head to Tail: Uncertainty Calibration Under Long-Tailed Distribution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 19978–19987.

Chen, Y.; Garcia, E. K.; Gupta, M. R.; Rahimi, A.; and Cazzanti, L. 2009. Similarity-based Classification: Concepts and Algorithms. *Journal of Machine Learning Research*, 10(27): 747–776.

Cubuk, E. D.; Zoph, B.; Mane, D.; Vasudevan, V.; and Le, Q. V. 2019. Autoaugment: Learning Augmentation Strategies From Data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 113–123.

Cubuk, E. D.; Zoph, B.; Shlens, J.; and Le, Q. V. 2020. RandAugment: Practical Automated Data Augmentation with A Reduced Search Space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 702–703.

Cui, J.; Liu, S.; Tian, Z.; Zhong, Z.; and Jia, J. 2023. ResLT: Residual Learning for Long-Tailed Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(3): 3695–3706.

DeepSeek-AI; Daya Guo; Dejian Yang; Haowei Zhang; Junxiao Song; et al. 2025. DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning. arXiv:2501.12948.

DeVries, T.; and Taylor, G. W. 2017. Improved Regularization of Convolutional Neural Networks with Cutout. *arXiv preprint arXiv:1708.04552*.

Dong, B.; Zhou, P.; Yan, S.; and Zuo, W. 2022. LPT: Long-Tailed Prompt Tuning for Image Classification. In *Advances in Neural Information Processing Systems*.

Ghosh, P.; Gupta, P. S.; Uziel, R.; Ranjan, A.; Black, M. J.; and Bolkart, T. 2020. GIF: Generative interpretable faces. In *2020 International Conference on 3D Vision (3DV)*, 868–878. IEEE.

Han, P.; Ye, C.; Zhou, J.; Zhang, J.; Hong, J.; and Li, X. 2024. Latent-based Diffusion Model for Long-tailed Recognition. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2639–2648. IEEE Computer Society.

He, H.; and Garcia, E. A. 2009. Learning from Imbalanced Data. *IEEE Transactions on Knowledge and Data Engineering*, 21(9): 1263–1284.

He, Y.-Y.; Wu, J.; and Wei, X.-S. 2021. Distilling Virtual Examples for Long-tailed Recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 235–244.

Ho, J.; Jain, A.; and Abbeel, P. 2020. Denoising Diffusion Probabilistic Models. *Advances in Neural Information Processing Systems*, 33: 6840–6851.

Jamal, M. A.; Brown, M.; Yang, M.-H.; Wang, L.; and Gong, B. 2020. Rethinking Class-Balanced Methods for Long-tailed Visual Recognition from a Domain Adaptation Perspective. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7610–7619.

Kang, B.; Xie, S.; Rohrbach, M.; Yan, Z.; Gordo, A.; Feng, J.; and Kalantidis, Y. 2020. Decoupling Representation and Classifier for Long-tailed Recognition. In *Eighth International Conference on Learning Representations*.

Karras, T.; Aittala, M.; Aila, T.; and Laine, S. 2022. Elucidating the Design Space of Diffusion-Based Generative Models. In *Proc. NeurIPS*.

Karras, T.; Laine, S.; and Aila, T. 2019. A Style-Based Generator Architecture for Generative Adversarial Networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Li, M.; Zhikai, H.; Lu, Y.; Lan, W.; Cheung, Y.-m.; and Huang, H. 2024. Feature Fusion from Head to Tail for Long-tailed Visual Recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 13581–13589.

Lin, T.-Y.; Goyal, P.; Girshick, R.; He, K.; and Dollár, P. 2017. Focal Loss for Dense Object Detection. In *Proceedings of the IEEE International Conference on Computer Vision*, 2980–2988.

Liu, Z.; Miao, Z.; Zhan, X.; Wang, J.; Gong, B.; and Yu, S. X. 2019. Large-Scale Long-tailed Recognition in an Open World. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2537–2546.

Ma, T.; Geng, S.; Wang, M.; Shao, J.; Lu, J.; Li, H.; Gao, P.; and Qiao, Y. 2022a. A Simple Long-Tailed Recognition Baseline via Vision-Language Model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Ma, T.; Geng, S.; Wang, M.; Xu, S.; Li, H.; Zhang, B.; Gao, P.; and Qiao, Y. 2022b. Unleashing the Potential of Vision-Language Models for Long-Tailed Visual Recognition. In *33rd British Machine Vision Conference*.

Mariani, G.; Scheidegger, F.; Istrate, R.; Bekas, C.; and Malossi, C. 2018. BAGAN: Data Augmentation with Balancing GAN. In *International Conference on Machine Learning*.

Narayan, K.; Vs, V.; and Patel, V. M. 2025. SegFace: Face Segmentation of Long-tail Classes. In *Proceedings of the*

AAAI Conference on Artificial Intelligence, volume 39, 6182–6190.

Nichol, A. Q.; and Dhariwal, P. 2021. Improved Denoising Diffusion Probabilistic Models. In *International Conference on Machine Learning*, 8162–8171. PMLR.

Rangwani, H.; Aithal, S. K.; Mishra, M.; et al. 2022a. Escaping Saddle Points for Effective Generalization on Class-Imbalanced Data. *Advances in Neural Information Processing Systems*, 35: 22791–22805.

Rangwani, H.; Jaswani, N.; Karmali, T.; Jampani, V.; and Babu, R. V. 2022b. Improving GANs for Long-tailed Data through Group Spectral Regularization. In *European Conference on Computer Vision*, 426–442. Springer.

Ren, M.; Zeng, W.; Yang, B.; and Urtasun, R. 2018. Learning to Reweight Examples for Robust Deep Learning. In *International Conference on Machine Learning*, 4334–4343. PMLR.

Saharia, C.; Chan, W.; Saxena, S.; Li, L.; Whang, J.; Denton, E. L.; Ghasemipour, K.; Gontijo Lopes, R.; Karagol Ayan, B.; Salimans, T.; et al. 2022. Photorealistic Text-to-image Diffusion Models with Deep Language Understanding. *Advances in Neural Information Processing Systems*, 35: 36479–36494.

Settles, B. 2009. Active Learning Literature Survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison.

Shao, J.; Zhu, K.; Zhang, H.; and Wu, J. 2024. DiffuLT: Diffusion for Long-tail Recognition Without External Knowledge. *Advances in Neural Information Processing Systems*, 37: 123007–123031.

Shi, J.-X.; Wei, T.; Xiang, Y.; and Li, Y.-F. 2023. How Re-sampling Helps for Long-tail Learning? *Advances in Neural Information Processing Systems*, 36: 75669–75687.

Song, Y.; Sohl-Dickstein, J.; Kingma, D. P.; Kumar, A.; Ermon, S.; and Poole, B. 2021. Score-Based Generative Modeling through Stochastic Differential Equations. In *International Conference on Learning Representations*.

Soviany, P.; Ionescu, R. T.; Rota, P.; and Sebe, N. 2022. Curriculum Learning: A Survey. *International Journal of Computer Vision*, 130(6): 1526–1565.

Tian, C.; Wang, W.; Zhu, X.; Dai, J.; and Qiao, Y. 2022. VL-LTR: Learning Class-wise Visual-Linguistic Representation for Long-Tailed Visual Recognition. In *European Conference on Computer Vision*.

Tian, J.; Jiang, Y.; Zhang, J.; Luo, H.; and Yin, S. 2024. A Novel Data Augmentation Approach to Fault Diagnosis with Class-imbalance Problem. *Reliability Engineering & System Safety*, 243: 109832.

Wang, X.; Lian, L.; Miao, Z.; Liu, Z.; and Yu, S. 2021. Long-tailed Recognition by Routing Diverse Distribution-Aware Experts. In *International Conference on Learning Representations*.

Wang, Z.; Xu, Q.; Yang, Z.; He, Y.; Cao, X.; and Huang, Q. 2023. A Unified Generalization Analysis of Re-weighting and Logit-adjustment for Imbalanced Learning. *Advances in Neural Information Processing Systems*, 36: 48417–48430.

Yang, A.; Yang, B.; Zhang, B.; Hui, B.; et al. 2024. Qwen2.5 Technical Report. *CoRR*, abs/2412.15115.

Zhang, H.; Cisse, M.; Dauphin, Y. N.; and Lopez-Paz, D. 2018. mixup: Beyond Empirical Risk Minimization. In *International Conference on Learning Representations*.

Zhang, X.; Wang, Q.; Zhang, J.; and Zhong, Z. 2020. Adversarial AutoAugment. In *International Conference on Learning Representations*.

Zhang, Y.; Wei, X.-S.; Zhou, B.; and Wu, J. 2021. Bag of Tricks for Long-tailed Visual Recognition with Deep Convolutional Neural Networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, 3447–3455.

Zhao, S.; Wen, X.; Liu, J.; Ma, C.; Yuan, C.; and Qi, X. 2025a. Learning from Neighbors: Category Extrapolation for Long-Tail Learning. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 30483–30492.

Zhao, Y.; Huang, J.; Hu, J.; Wang, X.; Mao, Y.; Zhang, D.; Jiang, Z.; Wu, Z.; Ai, B.; Wang, A.; et al. 2025b. Swift: A Scalable Lightweight Infrastructure for Fine-tuning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 29733–29735.

Zhou, B.; Cui, Q.; Wei, X.-S.; and Chen, Z.-M. 2020. Bbn: Bilateral-branch Network with Cumulative Learning for Long-tailed Visual Recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9719–9728.