Synthesizing Minority Samples for Long-tailed Classification via Distribution Matching

Anonymous authors
Paper under double-blind review

Abstract

In many real-world applications, deep neural networks (DNNs) often perform poorly on datasets with long-tailed distributions. To address this issue, a promising approach is to propose an optimization objective to transform real majority samples into synthetic minority samples. However, this objective is designed only from the classification perspective. To this end, we propose a novel framework that synthesizes minority samples from the majority by considering both classification and distribution matching. Specifically, our method adjusts the distribution of synthetic minority samples to closely align with that of the true minority class, while enforcing the synthetic samples to learn more generalizable and discriminative features of the minority class. Experimental results on several standard benchmark datasets demonstrate the effectiveness of our method in both long-tailed classification and synthesizing high-quality synthetic minority samples.

1 Introduction

The success of deep learning for supervised learning relies on high-quality large-scale datasets, which are often assumed to have nearly balanced numbers of samples for each class (Russakovsky et al., 2015). However, real-world datasets usually suffer from a long-tailed problem, where a few majority classes occupy most data while many minority classes have very few samples (Zhou et al., 2017; Liu et al., 2015). Deep neural networks (DNNs) trained on long-tailed datasets have poor generalization performance, especially in minority classes (Zhou et al., 2020; Liu et al., 2019). Therefore, it is of practical importance to develop methods for mitigating the long-tailed problem.

To alleviate the imbalanced issue, several kinds of methods have been proposed in the past decade, in which the data-level approach has received significant attention due to the simplicity and effectiveness (Yang et al., 2022). This approach usually aims to achieve a balanced training data distribution via re-sampling (*i.e.*, under-sampling (He & Garcia, 2009), over-sampling (Van Hulse et al., 2007; Gao et al., 2023)) or data augmentation (Chu et al., 2020; Hong et al., 2022; Li et al., 2021; Ahn et al., 2023; Gao et al., 2024b). In the context of re-sampling, one representative method is the Synthetic Minority Over-sampling Technique (SMOTE) (Chawla et al., 2002), which synthesizes minority samples by interpolating between existing minority samples and their nearest neighbor samples. Recently, Kim et al. (2020) have revisited the over-sampling framework and proposed a new way of to synthesize minority samples, called Major-to-minor (M2m). A unique advantage of M2m over SMOTE-based methods is that M2m utilizes the majority samples to generate minority samples via an optimization process, thus can "cook with much more raw materials". In this way, M2m is able to "transform" majority samples into minority samples to achieve a balanced dataset.

Despite its initial success, M2m only focuses on optimizing the synthetic minority samples from the classification perspective, whose similarity with the real samples in the concerned minority class is overlooked. In other words, a synthetic sample $\hat{x}$ initialized from a majority class k_0 is viewed as a sample of the minority class k if a pre-trained classifier g identifies it as class k confidently, and a target classifier f has low confidence about it on k_0, ignoring whether the synthetic sample upholds the genuine characteristics (i.e., feature distribution) of the class k.

In this work, we propose a novel framework for synthesizing minority samples via distribution matching. Our insight is that *a desired synthetic minority sample should not only satisfy the classification constraints about* $g(\hat{x})$ *and* $f(\hat{x})$*, but also be distributionally close to real samples in the target minority class.* To satisfy these, we introduce a principled approach that optimizes the synthetic minority samples by enforcing them to satisfy the classification constraints and being close to the distribution of real samples, by minimizing the optimal transport (OT) distance (Peyré et al., 2019). Moreover, to mitigate the harmfulness of unreliable synthetic samples, we define a sample rejection criteria based on the distance between synthetic minority samples and real minority samples.

In order to enhance the generality of our method, we introduce an additional regularization term concerning the "confusing class" within the minority class, which accounts for instances where minority samples are frequently misclassified into a specific class rather than other classes. In this way, we relax the label requirement of majority samples and as a result, our proposed method can translate not only In-Distribution (ID) majority samples but also Out-of-Distribution (OOD) samples (Wei et al., 2022) into synthetic minority samples, making ours more applicable in practice. Similar to M2m, our method can be used as a plug-in approach to enhance the performance of other methods, e.g., reweighting loss. Moreover, we conduct extensive experiments on standard benchmark datasets and our methods achieves improved long-tailed classification performance. In conclusion, our contributions are summarized as follows:

1. To address long-tailed classification problem, we propose a general framework for synthesizing minority samples via distribution matching, where we formulate real samples and synthetic ones as two distributions.
2. We optimize synthetic minority samples by enforcing them to satisfy the classification constraints and keep close to the real representation distribution by minimizing the OT distance.
3. By introducing a novel constraint on confusion classes and a general sample rejection criteria based on feature distance, our framework can simultaneously optimize ID and OOD samples to achieve effective over-sampling in minority classes.
4. Extensive experiments on standard benchmarks demonstrate the effectiveness of our method, showing that is a promising over-sampling framework for long-tailed classification problem.

2 Preliminaries

Optimal transport. OT is a widely used measurement for comparing distributions (Peyré et al., 2019), where we only focus on the discrete situation that is more related to our framework. Assuming we have two sets of points (features), we can formulate the discrete distributions as $P = \sum_{n=1}^{N} u_n \delta_{x_n}$ and $Q = \sum_{m=1}^{M} v_m \delta_{y_m}$, where δ is Dirac function and $\boldsymbol{u} \in \Delta^N$ and $\boldsymbol{v} \in \Delta^M$ are the discrete probability vectors that sum to 1. The discrete OT distance between distribution P and Q can be formulated as:

$$\min_{\mathbf{T} \in \Pi(P,Q)} \langle \mathbf{T}, \mathbf{C} \rangle = \sum_{n}^{N} \sum_{m}^{M} T_{nm} C_{nm}, \tag{1}$$

where $\mathbf{C} \in \mathbb{R}_{>0}^{n \times m}$ is the cost matrix whose each point denotes the distance between x_n and y_m and transport probability matrix $\mathbf{T} \in \mathbb{R}_{>0}^{n \times m}$ satisfies $\Pi(P,Q) := \left\{ \mathbf{T} | \sum_{n=1}^{N} T_{nm} = v_m, \sum_{m=1}^{M} T_{nm} = u_n \right\}$. As directly optimizing 1 is always time-expensive, Sinkhorn algorithm (Cuturi, 2013) introduces an entropic constraint, *i.e.*, $H(\mathbf{T}) = -\sum_{nm} T_{nm} \ln T_{nm}$ for fast optimization.

Long-tailed classification. Assume a training dataset $\mathcal{D}_{\text{train}} = \{(x_i, y_i)\}_{i=1}^{N}$, where $x_i \in \mathbb{R}^d$ denotes the i-th input and y_i means its corresponding label over K classes. Let N denote the number of the entire training data and N_k is that of class k, where we assume $N_1 \geq N_2 \geq ... \geq N_K$ without loss of generality. Denote $f : \mathbb{R}^d \rightarrow \mathbb{R}^K$ as the target classifier, which can be learned by empirical risk minimization (ERM) over the training set with an appropriate loss function $\mathcal{L}(f)$:

$$\min_{f} \mathbb{E}_{(x,y) \sim \mathcal{D}_{\text{train}}} [\mathcal{L}(f; x, y)]. \tag{2}$$

Eq. 2 optimized f usually performs poorly on the minority classes.

3 Method

3.1 Motivation

To motivate our method, we first review the most related work in the line of over-sampling, called Major-to-minor (M2m) (Kim et al., 2020). M2m aims to construct a new balanced dataset $\mathcal{D}_{\text{bal}}$ from the original dataset $\mathcal{D}_{\text{train}}$ by generating $N_1 - N_k$ synthetic samples for each class k, where the concerned classifier f trained on $\mathcal{D}_{\text{bal}}$ is expected to perform better than that trained on $\mathcal{D}_{\text{train}}$. Note that $N_1 \geq N_2 \geq ... \geq N_K$. Therefore, generating $N_1 - N_k$ synthetic samples for each class k enables a balanced dataset. Here, synthetic samples in minority classes are generated by translating from other samples in majority classes.

In addition to the to-be-learned classifier f trained on $\mathcal{D}_{\text{bal}}$, M2m assumes a baseline classifier g pre-trained on the imbalanced dataset $\mathcal{D}_{\text{train}}$ with standard ERM training, where f and g have the same structure. Although g may not achieve the optimal performance, it is expected to achieve reasonable performance on the imbalanced training dataset. To obtain a synthetic sample $\hat{x}$ for a minority class k, M2m uses a training sample x_0 of a major class k_0 in $\mathcal{D}_{\text{train}}$, where $k_0 < k$, and then optimizes $\hat{x}$ based on the gradient ascent.

Although M2m can achieve promising results, it translates $\hat{x}$ from x_0 purely in the view of classification and ignores the similarity between $\hat{x}$ and the corresponding real samples in the concerned minority class k. In this scenario, the synthetic sample might mislead the model and cause inaccurate predictions.

3.2 Learning Synthetic Minority Samples with Distribution Matching

To address the above issue, we aim to learn high-quality synthetic samples that not only satisfy the classification constraints about $g(\hat{x})$ and $f(\hat{x})$ but also follow the distribution of real samples in the target minority class k. Following M2m, we achieve the first goal by solving the optimization problem below:

$$\arg\min_{\hat{x}:=x_0+\epsilon} \mathcal{L}(g(\hat{x}), k) + \lambda f_{k_0}(\hat{x}), \tag{3}$$

where $\hat{x}$ is initialized with $x_0 + \epsilon$ and ϵ is standard Gaussian noise. Next, we focus on the second goal. Taking the k-th class in $\mathcal{D}_{\text{train}}$ as an example, we denote $\mathcal{D}_k = \{(x_n, y_n)\}_{n=1}^{N_k}$ as the set of real samples, and $\hat{\mathcal{D}}_k = \{(\hat{x}_m, \hat{y}_m)\}_{m=1}^{M_k}$ as the to-be-learned synthetic set, where $M_k = N_1 - N_k$ is the number of synthetic samples of class k. Then the empirical distributions of $\mathcal{D}_k$ and $\hat{\mathcal{D}}_k$ can be formulated as:

$$P_k = \sum_{n=1}^{N_k} \frac{1}{N_k} \delta_{x_n}, \qquad Q_k = \sum_{m=1}^{M_k} \frac{1}{M_k} \delta_{\hat{x}_m}. \tag{4}$$

Note that label is omitted since $\hat{y}_m = y_n = k$. Moving beyond Eq. 3, which only utilizes the classification loss to learn minority synthetic samples, we further introduce a distribution matching loss to enforce the to-be-learned distribution Q_k to stay close to the real distribution P_k of class k. Let $\text{Dist}(P_k, Q_k)$ denote the distance between the distributions P_k and Q_k. Here we adopt the principled approach of OT to define $\text{Dist}(P_k, Q_k)$, although other approaches are also available, such as Maximum Mean Discrepancy (MMD) (Gretton et al., 2012) and Energy Distance (ED) (Rizzo & Székely, 2016). We defer the implementation of $\text{Dist}(P_k, Q_k)$ with other measures such as Maximum Mean Discrepancy (MMD) (Gretton et al., 2012) and Energy Distance (ED) (Rizzo & Székely, 2016) to Appendix A.

Since the training images are highly dimensional, minimizing the distribution distance in the image space is expensive and inaccurate. Therefore, we assume an embedding function $\psi_\theta : \mathbb{R}^d \to \mathbb{R}^{d'}$ parameterized with θ and compute the distribution distance $\text{Dist}_\theta(P_k, Q_k)$ in the feature space. Specifically, we define it using the entropic OT:

$$\text{Dist}_\theta(P_k, Q_k) = \min_{\mathbf{T} \in \Pi(P_k, Q_k)} \langle \mathbf{T}, \mathbf{C} \rangle - \gamma H(\mathbf{T}), \tag{5}$$

where $\gamma > 0$ is a hyper-parameter for the entropy constraint $H(\mathbf{T})$. The transport plan satisfies:

$$\Pi(P_k, Q_k) := \left\{ \mathbf{T} | \sum_{n=1}^{N_k} T_{nm} = 1/M_k, \sum_{m=1}^{M_k} T_{nm} = 1/N_k \right\}, \tag{6}$$

and the cost function C_{nm} measures the distance between the real sample x_n and synthetic sample $\hat{x}_m$. C_{nm} can be viewed as a distance metric in the embedding space. Although theoretically it is possible to use any reasonable distance metric, we use the cosine similarity, *i.e.*, $C_{nm} = 1 - \cos(\psi_\theta(x_n), \psi_\theta(\hat{x}_m))$, which gives the best performance in this work.

3.3 Embedding Function ψ_θ and Optimization Problem

In order to achieve efficient computation of distribution distance, the parameterization of an embedding function ψ_θ is necessary and important. Commonly, we can employ the feature extractor in g or f as the embedding function. However, g is a biased model and f is learned during each training iteration, whose parameters may not be optimal for computing the $\text{Dist}_\theta(P_k, Q_k)$. Motivated by Zhao & Bilen (2023) that computes feature distance based on a family of models, we also match P_k and Q_k in many sampled embedding spaces. Specifically, when computing $\text{Dist}_\theta(P_k, Q_k)$ each time, we randomly sample the initialized network parameter, *i.e.*, $\theta \sim P_\theta$, where P_θ denotes the probability distribution of θ, then use θ to parameterize the embedding function. Moreover, we experimentally validate that the family of randomly initialized embedding spaces can produce better results than using one embedding space.

To summarize, we can minimize $\mathbb{E}_{\theta \sim P_\theta} [\text{Dist}_\theta(P_k, Q_k)]$ such that the synthetic samples are optimized to match the original data distribution in various embedding spaces. The overall optimization objective is formulated as:

$$\underset{\{\hat{x}_m := x_{0,m} + \epsilon_m\}_{m=1}^{M_k}}{\arg\min} \sum_{m=1}^{M_k} \left[\mathcal{L}(g(\hat{x}_m), k) + \lambda_1 f_{k_0}(\hat{x}_m) \right] + \lambda_2 \mathbb{E}_{\theta \sim P_\theta} \left[\text{Dist}_\theta(P_k, Q_k) \right], \tag{7}$$

where $x_{0,m}$ denotes the randomly sampled image from the major class k_0 for $\hat{x}_m$ and ϵ_m is the randomly sampled Gaussian noise.

3.4 Leveraging Out-of-Distribution Data

Beyond translating majority samples in the ID setting (e.g., samples in $\mathcal{D}_{\text{train}}$) to achieve efficient over-sampling and data augmentation for minority classes, it is more practical and valuable by leveraging OOD data to achieve the balance of a long-tailed dataset. Let $\mathcal{D}_{\text{ood}} = \{x_i\}_{i=1}^{N_o}$ denote the OOD dataset, where $x_i \in \mathbb{R}^d$ denotes i-th sample. We assume that the OOD dataset is unlabeled or the label information is not useful due to the large distribution shift from the ID dataset $\mathcal{D}_{\text{train}}$. Now we can initialize the minority sample as $\hat{x}_m := x_{\text{ood},m} + \epsilon_m$, where $x_{\text{ood},m}$ is a randomly sampled image from $\mathcal{D}_{\text{ood}}$ for $\hat{x}_m$.

Recall that M2m restricts the target classifier f to have lower confidence on the original class k_0 of x_0 by adding a regularization term in Eq. 3. However, the introduction of the OOD dataset brings a challenge as there is no corresponding k_0 for each $x_{\text{ood},m}$ in $\mathcal{D}_{\text{ood}}$. Therefore, we replace the constraint of the synthetic samples about k_0 by introducing a confusing class k_c. Specifically, we obtain the confusion matrix $\mathbf{A} \in \mathbb{R}^{K \times K}$ using a randomly sampled balanced subset from $\mathcal{D}_{\text{train}}$ and the pre-trained classifier g, whose element A_{ij} denotes the probability that a sample belongs to class i but is predicted as class j. Then, for the target minority class k, k_c is its most confusing class if $A_{k,k_c} \geq A_{k,i}$, where $i \in [1, K]$ and $i \neq k$. Finally, we design the constraint on the confusing class for an optimized sample $\hat{x}_m$ as $f_{k_c}(\hat{x}_m)$ and rewrite Eq. 7 as:

$$\underset{\{\hat{x}_m := x_{\text{ood},m} + \epsilon_m\}_{m=1}^{M_k}}{\arg\min} \sum_{m=1}^{M_k} \left[\mathcal{L}(g(\hat{x}_m), k) + \lambda_1 f_{k_c}(\hat{x}_m) \right] + \lambda_2 \mathbb{E}_{\theta \sim P_\theta} \left[\text{Dist}_\theta(P_k, Q_k) \right], \tag{8}$$

where $f_{k_c}(\hat{x}_m)$ restricts f to have lower confidence on the confusing class k_c. That is to say, we should avoid the synthetic samples to contain significant information of the confusing class in the viewpoint of target classifier f. In addition to addressing the issue of exploiting $\mathcal{D}_{\text{ood}}$, this regularization term can not only address the issue of exploiting OOD but also be added to the training loss in the ID scenario.

Algorithm 1: Oversampling Minority Samples via Our Method (In-Distribution).

Input : $\mathcal{D}_{\text{train}}$, classifier f, pre-trained classifier g and hyper-parameters.

Initialize $\mathcal{D}_{\text{bal}} \leftarrow \mathcal{D}_{\text{train}}$;
for $k = 2, ..., K$ **do**
  Compute $M_k \leftarrow N_1 - N_k$;
  Initialize $\hat{\mathcal{D}}_k \leftarrow \emptyset$;
  for $m = 1, ..., M_k$ **do** `// Step 1. Sample selection`
    Sample a majority class k_0 with $p = 1 - \beta^{(N_{k_0} - N_k)^+}$;
    Sample a $x_{0,m}$ from k_0 ;
    Initialize $\hat{x}_m \leftarrow x_{0,m} + \epsilon_m$ with a Gaussian noise ϵ_m.;
    Update $\hat{\mathcal{D}}_k \leftarrow \hat{\mathcal{D}}_k \cup \{(\hat{x}_m, k)\}$;
  end
  Build $Q_k = \sum_{m=1}^{M_k} \frac{1}{M_k} \delta_{\hat{x}_m}$ and $P_k = \sum_{n=1}^{N_k} \frac{1}{N_k} \delta_{x_n}$ according to Eq. 4;
  for $t = 1, ..., T$ **do** `// Step 2. Optimize x̂`
    Update $\hat{\mathcal{D}}_k$ by solving $\underset{\{\hat{x}_m := x_{0,m} + \epsilon_m\}_{m=1}^{M_k}}{\arg\min} \sum_{m=1}^{M_k} \left[\mathcal{L}(g(\hat{x}_m), k) + \lambda_1 f_{k_0}(\hat{x}_m)\right] + \lambda_2 \mathbb{E}_{\theta \sim P_\theta} \left[\text{Dist}_\theta(P_k, Q_k)\right]$ according to Eq. 7;
  end
  for $\hat{x}_m$ *in* $\hat{\mathcal{D}}_k$ **do** `// Step 3. Sample rejection for x̂`
    if $\mathcal{L}(g(\hat{x}_m), k) \geq \tau$ *or Reject* $= 1$ **then**
      $\hat{x}_m \leftarrow$ with a random sample from class k in $\mathcal{D}_{\text{train}}$;
    end
    Update $\mathcal{D}_{\text{bal}} \leftarrow \mathcal{D}_{\text{bal}} \cup \{(\hat{x}_m, k)\}$;
  end
end

3.5 Implementation Details

Mini-batch learning. We adopt the stochastic gradient descent (SGD) (Ruder, 2016) to learn the target classifier f and optimize the synthetic samples based on a batch-wise re-sampling following M2m. More specifically, we use a standard over-sampling (Huang et al., 2016) to obtain a class-balanced mini-batch $\{(x_i, y_i)_{i=1}^B\}$. To stimulate the generation of $N_1 - N_k$ samples for any k, for each sample x_i in the mini-batch, we use probability $\frac{N_1 - N_{y_i}}{N_1}$ to decide whether learning a synthetic sample $\hat{x}_i$ to replace x_i.

Sample selection criteria for x_0. We choose a seed sample x_0 to learn $\hat{x}_i$ for x_i with class k. In an OOD setting, we just randomly sample an image from $\mathcal{D}_{\text{ood}}$ as x_0. In ID setting, we first choose k_0 with the probability $k_0 \sim 1 - \beta^{(N_0 - N_k)^+}$ in the current mini-batch, where $(\cdot)^+ := \max(\cdot, 0)$, and $\beta \in [0, 1)$ is a hyper-parameter. After that, x_0 is sampled uniformly among samples in class k_0. Once we choose the seed sample x_0 for the minority class k, we start to learn $\hat{x}$. Rather than using all N_k samples within class k, we randomly sample a subset of the real samples from class k to construct P_k in each iteration for saving cost consumption. Besides, we use the to-be-optimized samples for class k in the current mini-batch to build Q_k. Finally, we optimize $\hat{x}$ using Eq. 7 or Eq. 8 by performing T iterations with a step size of η, depending on ID or OOD, respectively.

Sample rejection criteria for $\hat{x}$. To reduce the harmfulness of unreliable synthetic samples, it is necessary to design sample rejection criteria to discard unsatisfactory synthetic samples. Here, we consider two conditions that can determine a reliable synthetic sample. Following M2m, the first one is setting a threshold $\tau > 0$ and rejecting the resultant synthetic sample for k-th class if $\mathcal{L}\left(g; \hat{x}, k\right) > \tau$ for stability. For the second factor, M2m designs the rejection probability as $\mathbb{P}\left(\text{Reject } \hat{x} \mid k_0, k\right) \propto \beta^{\left(N_{k_0} - N_k\right)^+}$.

Different from M2m that utilizes the class frequency of k_0 and target class k to decide the reliability of $\hat{x}$, we introduce a more general sample-level criteria to reject $\hat{x}$ (*i.e.*, Reject $\hat{x} = 1$) if it satisfies:

$$\frac{1}{N_k} \sum_{n=1}^{N_k} d(\psi_\theta(\hat{x}), \psi_\theta(x_n)) > \frac{1}{{N_k}^2} \sum_{n=1}^{N_k} \sum_{m=1}^{N_k} d(\psi_\theta(x_n), \psi_\theta(x_m)), \tag{9}$$

Table 1: Test top-1 errors (%) of ResNet-32 on CIFAR-LT-10 / CIFAR-LT-100 under different imbalance factors on the ID setting, where †, ‡ and * denote the results from the original paper, our reproduction and MetaSAug (Li et al., 2021), respectively. Results of SMOTE are from Kim et al. (2020). The methods are trained with CE loss unless otherwise stated.

Method	CIFAR-LT-10				CIFAR-LT-100			
Imbalance Factor	200	100	50	10	200	100	50	10
CE Loss*	34.13	29.86	25.06	13.82	65.30	61.54	55.98	44.27
Focal Loss* (Lin et al., 2017)	34.71	29.62	23.29	13.34	64.38	61.59	55.68	44.22
CB,CE Loss* (Cui et al., 2019)	31.23	27.32	21.87	13.10	64.44	61.23	55.21	42.43
LDAM-DRW* (Cao et al., 2019)	25.26	21.88	18.73	11.63	61.55	57.11	52.03	41.22
MetaSAug† (Li et al., 2021)	23.11	19.46	15.97	10.56	**60.06**	53.13	48.10	38.27
RSG‡ (Wang et al., 2021)	-	20.04	17.2	-	-	55.4	51.5	-
MBJ‡ (Liu et al., 2022)	-	19.0	13.4	11.2	-	54.2	47.4	39.3
CB-SAFA† (Hong et al., 2022)	27.18	23.68	19.79	12.07	60.34	54.13	52.04	39.77
CUDA† (Ahn et al., 2023)	-	-	-	-	-	57.3±0.4	52.8±0.4	40.4±0.6
CMO‡ (Park et al., 2022)	25.43	19.59	16.47	11.50	63.47	56.13	51.71	40.49
OTMix‡ (Gao et al., 2024b)	-	21.7	16.6	9.8	-	53.6	49.3	38.4
M2m† (Kim et al., 2020)	25.34±0.46‡	21.7±0.16	18.81±0.76‡	12.5±0.15‡	63.77±0.33‡	57.1±0.16	50.48±0.43‡	44.8±0.05‡
OURS	**22.85±0.12**	**18.20±0.21**	**12.96±0.11**	**9.34±0.11**	61.28±0.21	**52.95±0.18**	**47.02±0.26**	**37.57±0.32**

where $d(\psi_\theta(\hat{x}), \psi_\theta(x_n))$ indicates the distance between $\hat{x}$ and x_n and can be defined by cosine similarity, *i.e.*, $1 - \cos(\psi_\theta(\hat{x}), \psi_\theta(x_n))$. This rejection criterion can avoid the requirement for N_{k_0}, which can also be applied to the OOD scenario. The underlying intuition is that the synthetic samples are expected to have a smaller distance with real samples in class k than the intra-class distance. We replace x_i in the current mini-batch by $\hat{x}$ if it satisfies the above two factors. We summarize the synthetic process for the ID setting in Algorithm 1.

4 Experiments

In this section, we present experimental results to show the effectiveness of the proposed method. The detailed experiment settings and hyper-parameters are provided in Appendix C.1.

Datasets. We evaluate our method on CIFAR-LT-10 / CIFAR-LT-100, ImageNet-LT and Places-LT. We build CIFAR-LT-10 / CIFAR-LT-100 from the standard CIFAR-10/CIFAR-100 datasets (Krizhevsky et al., 2009) with IF $\in \{50, 100, 200\}$ (Kim et al., 2020; Kang et al., 2019; Li et al., 2021). ImageNet-LT is a subset of the ImageNet-2012 dataset (Deng et al., 2009) with 1000 classes and IF $= 1280/5$ (Kim et al., 2020; Ren et al., 2020). Places-LT is a subset from the Places-365 dataset (Zhou et al., 2017) with 365 classes and IF $= 4980/5$ (Cao et al., 2019; Ren et al., 2020).

Baselines. We compare with five types of baselines: (1) ***Cross-entropy (CE)***. (2) ***Re-weighting loss***, including Focal loss (Lin et al., 2017), Class-Balanced (CB) loss (Cui et al., 2019), Balanced-Softmax (BS) loss (Ren et al., 2020) and LDAM-DRW loss (Cao et al., 2019). (3) ***Feature based augmentation methods***, including MetaSAug (Li et al., 2021), SAFA (Hong et al., 2022), CUDA (Ahn et al., 2023), RSG (Wang et al., 2021) and MBJ (Liu et al., 2022). (4) ***Minority over-sampling methods***, including SMOTE (Chawla et al., 2002), M2m (Kim et al., 2020) and CMO (Park et al., 2022). (5) **OOD methods**, i.e., Open-Sampling (Wei et al., 2022).

4.1 Experiments on Long-tailed CIFAR

Results with the ID setting. Table 1 summarizes the average results of our method for three independent runs with standard deviation on CIFAR-LT-10 / CIFAR-LT-100 under different settings. We find that our method outperforms the CE baseline and re-weighting methods by a large margin. Moreover, our method achieves a significant improvement than both feature- and sample- based data augmentation methods, except for IF = 200 with CIFAR-LT-100 when compared with MetaSAug. Remarkably, the comparison between ours and the minority sample synthetic method, *i.e.*, M2m, confirms the validity of introducing the distribution matching loss when transferring the majority samples to the minority classes. Besides, we use MMD and ED to implement Dist_θ and report results on CIFAR-LT-10 and time complexity in Section **??**.

Results with the OOD setting. To validate whether our proposed method can translate OOD instances, we employ 300,000 random images[1] (Hendrycks et al., 2018) as the OOD dataset $\mathcal{D}_{\text{ood}}$ for CIFAR-LT-10 / CIFAR-LT-100 by following Open-Sampling (Wei et al., 2022). We report the performance in the case of IF = 100 and IF = 50. We find that our method is significantly better than Open-Sampling, which utilizes open-set noisy labels to re-balance the long-tailed training dataset. It is reasonable since we optimize the OOD samples from the view of classification and distribution matching rather than endowing them with noisy labels without re-labeling OOD samples. Besides, we perform experiments by combining both OOD and ID settings, which produce a better performance than that in the ID setting. The detailed results are provided in the supplementary materials.

Table 2: Test top-1 errors (%) of ResNet-32 on CIFAR-LT-10 / CIFAR-LT-100 with different imbalance factors on the OOD setting, where † denotes the results from the original paper, *OS* indicates Open-Sampling (Wei et al., 2022).

Method	CIFAR-LT-10		CIFAR-LT-100	
IF	100	50	100	50
OS†	22.38±0.28	18.24±0.51	59.74±0.65	55.23±0.25
OURS	**20.03±0.17**	**16.39±0.22**	**55.09±0.23**	**49.38±0.27**

Boosting other methods. To investigate whether our method can be combined with other long-tailed methods under ID and OOD settings, we consider several classical re-weighting losses, including CB loss (Cui et al., 2019) and BS loss (Ren et al., 2020). As shown in Tab. 3, our method significantly improves the performance of re-weighting methods under the ID setting and performs better than M2m. Under the OOD setting, M2m is not usable, while the performance of Open-Sampling is worse than our method combined with different re-weighting losses. These results indicate the effectiveness and flexibility of our method when combined with other methods under both ID and OOD settings.

Table 3: Test top-1 errors (%) of ResNet-32 on CIFAR-LT-10 dataset under both ID and OOD settings when combined with different re-weighting methods.

Method	CIFAR-LT-10 (ID)			CIFAR-LT-10 (OOD)		
IF	200	100	50	200	100	50
CE	34.13	29.86	25.06	-	-	-
+ M2m	25.34	21.70	18.81	-	-	-
+ OS	-	-	-	28.28	22.38	18.24
+ OURS	22.85	18.20	15.96	23.43	20.03	16.39
Δ	↓ 2.49	↓ 3.50	↓ 2.15	↓ 4.85	↓ 2.35	↓ 1.85
CB-DRW	31.23	27.32	21.87	-	-	-
+ M2m	25.24	19.33	18.25	-	-	-
+ OS	-	-	-	29.77	24.23	19.90
+ OURS	21.19	18.07	16.30	22.69	20.16	16.70
Δ	↓ 4.95	↓ 1.26	↓ 1.95	↓ 7.08	↓ 4.07	↓ 3.20
BS	-	21.97	18.37	-	-	-
+ M2m	25.16	23.43	19.96	-	-	-
+ OS	-	-	-	28.59	20.95	17.24
+ OURS	20.98	16.13	14.22	23.08	19.81	17.06
Δ	↓ 4.18	↓ 7.30	↓ 5.74	↓ 5.51	↓ 1.14	↓ 0.08

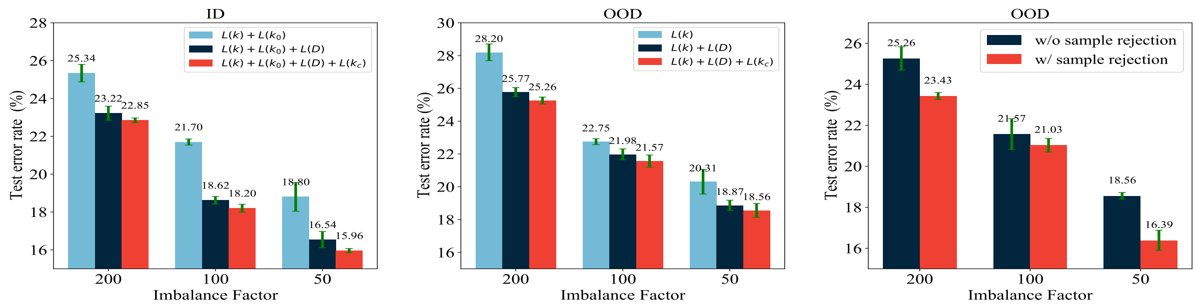
4.2 Experiments on ImageNet-LT and Places-LT

Results. As summarized in Tab. 4, we perform experiments on ImageNet-LT and Places-LT. We can see that our method using CE loss outperforms the vanilla CE, over-sampling and M2m, which indicates the effectiveness of generating the minority samples from the view of the distribution matching. Furthermore, our method can also be combined with other losses, where we take the BS loss as an example and obtain improvements by 1.69% and 0.93% compared with the BS loss on ImageNet-LT and Places-LT, respectively. These results show that our proposed data augmentation method is effective on large-scale complicated long-tailed datasets.

Table 4: Test top-1 errors (%) of ResNet-50 on ImageNet-LT and ResNet-152 on Places-LT.

Method	ImageNet-LT	Places-LT	Method	ImageNet-LT	Places-LT
CE	58.4	70.1	FSA	-	63.6
Focal Loss	-	65.4	MBJ	-	61.9
LDAM-DRW	50.2	-	CMO + RIDE	43.8	-
BS	49.0	61.3	OTMix+CE	48.0	-
RIDE (3 experts)	45.1	-	OTMix+BS	44.4	-
BCL	44.0	-	OTMix+RIDE	42.7	-
M2m + CE	55.40	63.27	**OURS+CE**	53.77	61.68
Over-Sampling + CE*	55.34	64.27	**OURS+BS**	**47.31**	**60.37**
RSG + LDAM-DRW	-	60.7	**OURS+RIDE**	**43.22**	-
MisLAS	47.3	-	**OURS+BCL**	**41.89**	-

[1]https://github.com/hendrycks/outlier-exposure.



(a) ID setting (b) OOD setting (c) OOD setting

Figure 1: Ablation experiments of our method on CIFAR-LT-10 with IF=100.

4.3 Ablation Study

We conduct a series of ablation studies on CIFAR-LT-10. Specifically, we use $L(k)$, $L(k_0)$, $L(k_c)$ and $L(D)$ to represent the $\mathcal{L}(g(\hat{x}_m), k)$, $f_{k_0}(\hat{x}_m)$, $f_{k_0}(\hat{x}_m)$ and $\mathbb{E}_{\theta\sim P_\theta}[\mathrm{Dist}_\theta(P_k, Q_k)]$, respectively.

Distribution matching loss $L(D)$. To verify the effect of our proposed distribution matching loss, we first compare M2m with $L(k)$+$L(k_0)$ and our method with $L(k)$+$L(k_0)$+$L(D)$ in Fig. 1(a) under the ID setting. It is clear that our method consistently outperforms M2m with different imbalance factors. The reason is that using distribution matching loss ensures the distribution of the synthetic samples to be close the distribution of the real samples. Thereby, our method can generate more effective synthetic samples. Besides, considering M2m cannot use OOD samples due to the regularization term $L(k_0)$, we adopt $L(k)$, $L(k)$+$L(D)$ to learn the minority samples, where the latter achieves better performance as shown in Fig. 1(b).

Regularization about the confusing class $L(k_c)$. Recall that M2m is only available in the ID setting and our method is applicable to the OOD setting by introducing the $L(k_c)$, where k_c is the confusing class of minority class k. As shown in Fig. 1(b), our method with $L(k)$+ $L(D)$+$L(k_c)$ can achieve better performance than using $L(k)$+$L(D)$, under the OOD setting with different imbalance factors. Besides, it is worth noting that the constraint can also be applied to the ID setting. As shown in Fig. 1(a), our method obtains additional performance improvements with the help of $L(k_c)$. The phenomenon illustrates that the regularization term on the confusing class k_c can make our method applicable to the OOD setting and perform better in the ID setting. The reason is that high-quality synthetic minority samples can be generated by ensuring that these samples have low confidence in the confusing class k_c.

Sample rejection in the OOD setting. As specified in previous section, the unreliable generation quality of synthetic samples urges us to propose the sample rejection criteria, especially in the OOD setting. To validate our proposed rejection strategy Eq. 9, we perform an ablation study on the CIFAR-LT-10 with IF = 10. Results in Fig. 1(c) present that using our proposed rejection strategy consistently improves the performance when leveraging the OOD samples for CIFAR-LT-10 with different imbalance factors. That is to say, our method can fully use OOD samples to generate minority samples while alleviating the toxicity of the distribution shift brought about by OOD samples.

Embedding spaces $\mathbb{E}_{\theta\sim P_\theta}[\cdot]$. We investigate the effect of the embedding for computing the distribution matching loss described in Eq. 7. We use the encoder in the to-be-learned model f and that in the pre-trained model g as the baselines, where we also discuss the performance of a randomly initialized encoder and a CLIP vision encoder (Radford et al., 2021). All encoders have the same architecture for a fair comparison, except for CLIP. As summarized in Tab. 5, the encoder in f serves as the worst embedding function. The possible reason behind this phenomenon is the coupling between the loss of distribution matching that optimizes synthetic minority samples and the training loss of the to-be-learned classifier f in each training iteration. Besides, using the encoder in the imperfect pre-trained g also achieves inferior results. It is reasonable since g is a biased

Table 5: Test top-1 errors (%) with different embeddings.

Feature Extractor	CIFAR-LT-10(ID)		
	200	100	50
Model g	25.60	21.24	16.55
Model f	25.72	20.03	16.88
A randomly initialized encoder	25.32	19.78	16.57
CLIP (RN50)	23.32	18.94	16.21
The family of random encoders	**22.85**	**18.20**	**15.96**

model and cannot extract satisfactory features. Interestingly, we find that only using a randomly initialized encoder during the entire training process can produce acceptable performance, proving the effectiveness of the appropriate and unbiased embedding function for the loss of distribution matching. Moreover, randomly initializing the embedding function at each training iteration outperforms other settings. It shows that the family of embedding spaces can be obtained by sampling randomly initialized DNNs, and is effective in computing the distance between real and synthetic samples. Moreover, we find that even the CLIP (Radford et al., 2021) visual encoder can be used for our distribution matching purposes.

4.4 Detailed ablation study on sample selection and rejection criteria

As shown in the left part of Table 6, the model performs worst under the ID setting when using only the k_0 selection criteria and no rejection criteria, regardless of IF values equals to 50 or 100. However, the introduction of $\mathrm{L}(p)$ as a quality metric for synthesized samples results in a marked improvement, with test errors decreasing from 31.16% to 20.73% for IF $= 100$ and from 27.25% to 20.86% for IF $= 50$. This finding underscores the efficacy of $\mathrm{L}(p)$ in discerning and selecting more reliable synthesized samples, leading to improved model performance. Furthermore, the subsequent incorporation of $\mathrm{L}(d)$ contributes to an even more pronounced decrease in test errors, underscoring its effectiveness in refining the selection of high-quality synthetic samples. These conclusions are corroborated by the OOD setting, which demonstrates that our proposed sample rejection criteria effectively identify high-quality, credible samples for model training, thereby catalyzing performance gains across different distribution settings. Finally, when maintaining $\mathrm{L}(p)+\mathrm{L}(d)$ as rejection criteria and evaluating the influence of different sample selection criteria, we observe that $k_0 \sim 1 - \beta^(N_0, N_k)^+$ demonstrates superior performance, surpassing the alternative approach of randomly selecting a seed sample for initialization.

Table 6: Test top-1 errors (%) on CIFAR-LT-10 with IF $\in [100, 50]$ under the ID and OOD settings. For the left part, We evaluate the influence of sample rejection criteria and sample selection criteria, where $\tau = 0.9$ and $\beta = 0.999$. For the right part, we evaluate the influence of different τ and β. All the experiments are conducted by using sample selection criteria and sample rejection criteria.

-	Selection	Rejection	100	50	-	τ	β	100	50
ID	k_0	-	31.16	27.25	ID	0.9	0.999	18.20	15.96
ID	k_0	$\mathrm{L}(p)$	20.73	20.86	ID	0.6	0.999	18.69	16.33
ID	k_0	$\mathrm{L}(p)+\mathrm{L}(d)$	18.20	15.96	ID	0.3	0.999	19.12	16.82
ID	Random	$\mathrm{L}(p)+\mathrm{L}(d)$	20.07	16.32	ID	0.9	0.888	18.53	16.27
OOD	Random	$\mathrm{L}(p)$	20.32	16.56	ID	0.9	0.777	18.97	16.59
OOD	Random	$\mathrm{L}(p)+\mathrm{L}(d)$	20.03	16.39	ID	0.9	0.666	18.77	16.30

As shown in the right part of Table 6, we evaluate the influence of different τ and β on the performance using $\mathrm{L}(p)+\mathrm{L}(d)$ as rejection criteria. A larger τ results in more low-quality synthetic samples being used to train the network, consequently leading to suboptimal classification performance. β controls the probability of selecting k_0, thereby influencing the diversity of the initialization of synthetic samples. When the β is fixed by 0.999, the model gives lower test errors with the increase of τ from 0.3 to 0.9. This observation aligns with the theoretical analysis of the impact of τ, indicating that our proposed sample rejection method, both theoretically and experimentally, effectively filters the quality of synthetic samples. For the selection of β, we can observe that the larger β gives the best performance when τ is fixed by 0.9, which indicates that the introduction of more diversity for the initialization will help the model generalize on the long-tailed dataset.

4.5 Combining OOD and ID.

Beyond leveraging the OOD samples to replace the majority samples in our framework, further, we explore whether combining the OOD setting and ID setting can produce better performance. To this end, we conduct experiments on CIFAR-LT-10 with IF $\in \{200, 100, 50\}$ on ResNet-32. Specifically, we firstly optimize

Table 7: Test top-1 errors (%) of ResNet-32 on CIFAR-LT-10 under different imbalance factors, where †and ‡denote the results from the original paper and our reproduction, respectively. The methods are trained with CE loss unless otherwise stated.

Distribution	Method	CIFAR-LT-10		
	Imbalance Factor	200	100	50
ID	M2m	25.34‡	21.7†	18.81†
OOD	Open-Sampling	28.28‡	22.38†	18.24†
OOD	OURS	23.43	20.03	16.39
ID	OURS	22.85	18.20	15.96
OOD to ID	OURS	21.83$\downarrow$ 1.02	17.96$\downarrow$ 0.24	15.32$\downarrow$ 0.64

synthetic samples and train the target classifier f using $\mathcal{D}_{ood}$. Then we save the best checkpoint and employ an additional 20 epochs to further train f under the ID setting, where we initialize the minority samples with majority samples, using the Alg. 1. In other words, we first use the OOD setting to train the target classifier f and then further train f under ID setting.

As shown in Tab. 7, our method in ID and OOD settings outperform the M2m and Open-Sampling, respectively. Furthermore, introducing the OOD dataset into the ID setting, our method further achieves 1.02%, 0.24% and 0.64% gains with IF $\in \{200, 100, 50\}$, respectively. These demonstrate that our framework in ID or OOD setting can achieve better performance than corresponding baselines. Besides, the OOD samples can be utilized to further improve the performance of our proposed method under the ID setting.

4.6 Visualizations

Visualization of synthetic samples in feature space. As shown in Fig. 2, we visualize synthetic minority samples and real minority samples in CIFAR-LT-10 (IF = 100) using t-SNE (Van der Maaten & Hinton, 2008) in feature space. We show classes 7, 8, 9, and 10 (descend ranked by the numbers of their samples), each of which has 232, 139, 83, and 50 real samples. We randomly select 50 synthetic samples for each class after using the sample rejection criteria. In terms of M2m, synthetic samples from each class are difficult to capture the corresponding real distribution. Besides, synthetic samples from different classes are seriously coupled together. As expected, our synthetic samples can effectively capture the real distribution of each class. Therefore, it reveals why our method can generate more beneficial synthetic minority samples than M2m.

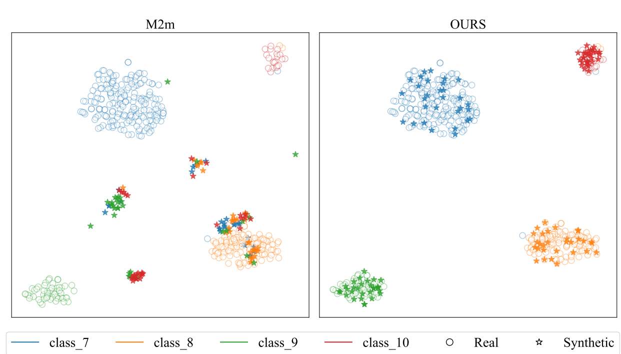


Figure 2: Visualization of the features of synthetic samples and real samples on CIFAR-LT-10 (IF = 100) with ResNet-32. where 'o' and the star indicate real and synthetic samples in the same class, respectively.

Visualization of synthetic samples in pixel space. We use M2m and ours to optimize a sample x_0 with k_0 = car to k = deer on the same pre-trained model g. Figure 3 shows that x_0 is correctly classified as a car with a probability of 0.99 on g. Then, after optimization, ours and M2m produce different synthetic samples $\hat{x}$ and corresponding noise, even though the two $\hat{x}$ are visually indistinguishable. At this time, the probability of $\hat{x}$ optimized by ours being classified as its original class k_0 on g is 0.07, and the probability of being classified to k is 0.91, while the corresponding probabilities of M2m are 0.19 and 0.73. This shows that ours successfully pushes the synthesized sample away from its original label k_0 on g, and makes it closer to our target label k. At the same time, ours also makes f believe that $\hat{x}$ is a sample from k with a higher probability (0.57, 0.32 larger than M2m) on the target classifier f, and its classification probability on k_0 is significantly lower than

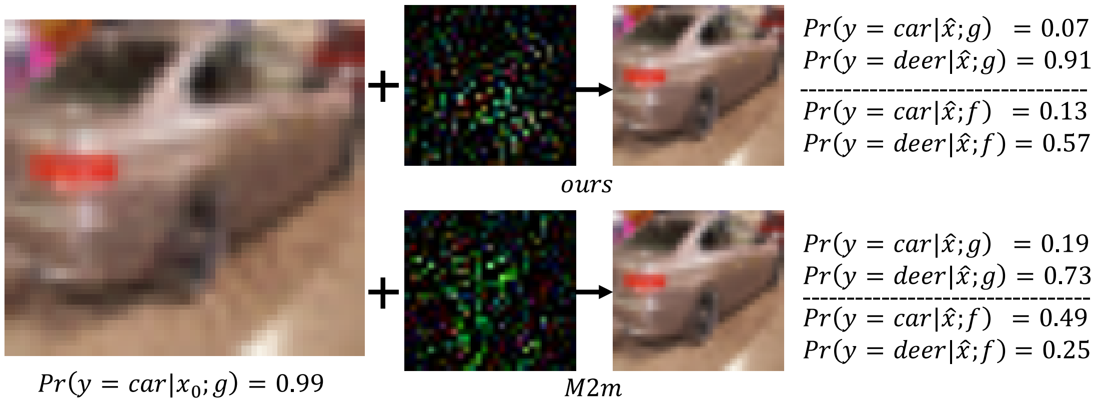


Figure 3: An illustration of a synthetic minority sample by our method and M2m, where g is assumed to be ResNet-32 trained by standard ERM. The noise image is amplified by 20 for better visibility.

the corresponding results of M2m by 0.30. This shows that the samples synthesized by our method are more credible and realistic for f.

In fact, we do not optimize a majority sample x_0 into a picture that is similar to a minority sample in the pixel space, like a generator. Instead, we simply optimize the sample x_0 directly to confuse our models g and f, making the models believe that our synthetic samples $\hat{x}$ are indeed from real minority class, which helps the network to generalize on the minority class. Recall the visualization of feature space, which shows that the feature distributions of our synthetic samples $\hat{x}$ are closer to real samples x of class k. This indicates that the features of our synthetic samples are more realistic and credible, where f regards the features of the real samples and the synthetic samples are from the same distribution. On the other hand, Section C.6 shows that the probability of our synthetic samples being correctly classified as the target class k on the target classifier f is significantly higher than that of M2m. From the classification perspective, our method can produce better synthetic minority samples and reduce the difficulty of the model learning from synthetic samples.

In summary, our starting point is to generate synthetic samples that are more realistic and credible for the model. From the perspective of pixel space, our method and M2m have no obvious difference. However, from the classification probability and feature matching perspectives, our synthetic samples are more realistic and credible for the network and, therefore, more effective than M2m.

4.7 Comparison of different implements of $\mathrm{Dist}_\theta(P_k, Q_k)$

To prove the generality and effectiveness of our method, we conduct experiments on CIFAR-LT-10 and CIFAR-LT-100 with different imbalanced factors (IF) by using different implements of $\mathrm{Dist}_\theta(P_k, Q_k)$, where the results of different methods are summarized in Tab. 8. We implement $\mathrm{Dist}_\theta(P_k, Q_k)$ using MMD, ED and OT, denoted as ours+MMD, ours+ED and ours+OT, respectively. We report the average results of our method for three runs with standard deviation independently. We can find that all of them have superior performance compared to the M2m baseline by a large margin. Besides, we can observe that ours+OT performs best, which might benefit to the more accurate characterization and measurement of the distance between distributions brought by OT. In other words, OT learns an optimal transport plan which endows each cost element with corresponding importance T_{ij}. These results demonstrate the effectiveness and generality of our proposed method.

4.8 Convergence and time complexity

To fairly compare training time-consuming, we conduct an experiment on CIFAR-LT-10 with IF = 100 on ResNet-32 in the same device environment with one Tesla-V100 GPU. As shown in Fig.**??**, although our

Table 8: Test top-1 errors (%) of ResNet-32 on CIFAR-LT-10 / CIFAR-LT-100 under different imbalance factors, where †and ‡denote the results from the original paper and our reproduction, respectively. The methods are trained with CE loss unless otherwise stated.

Method	CIFAR-LT-10			CIFAR-LT-100		
Imbalance Factor	200	100	50	200	100	50
M2m†	25.34±0.46‡	21.7±0.16	18.81±0.76‡	63.77±0.33‡	57.1±0.16	50.48±0.43‡
OURS+MMD	23.01±0.37	18.91±0.29	16.42±0.15	62.28±0.27	53.84±0.15	47.25±0.17
OURS+ED	22.93±0.24	18.75±0.15	16.16±0.22	61.97±0.10	53.01±0.22	47.33±0.28
OURS+OT	**22.85±0.12**	**18.20±0.21**	**15.96±0.11**	**61.28±0.21**	**52.95±0.18**	**47.02±0.26**

method takes a little bit more time than M2m. However, no matter using OT, MMD or ED for $\text{Dist}_\theta(P_k, Q_k)$, our proposed method with ID and OOD samples can outperform M2m given the same training time of M2m.

4.9 Additional Analysis

We analyze the influence of different OOD datasets in Appendix C.5, classification confidence of synthetic samples on the target classifier f in Appendix C.6, influence of the pre-trained model g in Appendix C.7, and visualization of the confusion matrix in Appendix C.8.

5 Related Work

Over-sampling methods for long-tailed problem. Data-based methods aim to solve the imbalance problem by building relatively balanced classes from the perspective of data, including under-sampling majority samples (He & Garcia, 2009; Drummond & Holte, 2003), over-sampling minority samples (Shen et al., 2016; Buda et al., 2018; Barandela et al., 2004) and data augmentation (Ahn et al., 2023; Park et al., 2022; Yan et al., 2019; Kim et al., 2020; Gao et al., 2023; 2024a; Li et al., 2025; Guo et al., 2022b). Our method has a close connection with minority over-sampling methods. A related work is Optimal transport over-sampling (OTOS) (Yan et al., 2019), which maps the noise to synthetic ones based on the Wasserstein barycenter. Different from OTOS which generates samples by a mapping matrix and is limited to a binary classification, we provide a more general and direct optimization objective for generating synthetic samples. By minimizing this objective, we can obtain synthetic samples with reliable classification confidence and high representation similarity, where we can handle multi-class classification task and leverage more practical OOD setting. Another related work, M2m (Kim et al., 2020), translates majority samples to the target minority class by maximizing the prediction probability. However, in our work, we optimize synthetic minority samples from both perspectives of classification confidence and distribution matching, where we extend the ID to OOD setting for further versatility.

Utilizing auxiliary dataset for long-tailed problem. In imbalanced learning, Yang & Xu (2020) leverage unlabeled ID data as additional samples to compensate for the minority classes, while Su et al. (2021) adopts a semi-supervised learning framework to incorporate out-of-class samples from related classes. Open-Sampling (Wei et al., 2022) explores the benefit of using OOD data in the long-tailed problem. The major difference between ours and Open-Sampling is that we translate OOD samples by introducing an optimization phase and introducing a sample rejection strategy but Open-Sampling assigns a noisy label to each OOD sample using a pre-defined label distribution without filtering the OOD data.

6 Conclusion

To address the long-tailed classification issue, we propose a novel framework for translating majority samples into synthetic minority samples by leveraging classification confidence and distribution matching. Our method optimizes the synthetic minority samples by enforcing them to satisfy the classification constraints and being close to the distribution of real samples in the target minority class. In addition, we introduce an effective regularization term for confusing classes, enabling our framework to better utilize available and rich OOD data to synthesize minority classes. Extensive experiments on benchmark datasets demonstrate that our framework can generate effective minority samples and achieve the desired long-tailed classification performance.

References

Sumyeong Ahn, Jongwoo Ko, and Se-Young Yun. Cuda: Curriculum of data augmentation for long-tailed recognition. *arXiv preprint arXiv:2302.05499*, 2023.

R. Barandela, R. M. Valdovinos, JS Sánchez, and F. J. Ferri. The imbalanced training sample problem: Under or over sampling? In *Structural, Syntactic, and Statistical Pattern Recognition, Joint IAPR International Workshops, SSPR 2004 and SPR 2004, Lisbon, Portugal, August 18-20, 2004 Proceedings*, 2004.

Patrick Bilic, Patrick Christ, Hongwei Bran Li, Eugene Vorontsov, Avi Ben-Cohen, Georgios Kaissis, Adi Szeskin, Colin Jacobs, Gabriel Efrain Humpire Mamani, Gabriel Chartrand, et al. The liver tumor segmentation benchmark (lits). *Medical Image Analysis*, 84:102680, 2023.

Mateusz Buda, Atsuto Maki, and Maciej A Mazurowski. A systematic study of the class imbalance problem in convolutional neural networks. *Neural networks*, 106:249–259, 2018.

Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Arechiga, and Tengyu Ma. Learning imbalanced datasets with label-distribution-aware margin loss. *Advances in neural information processing systems*, 32, 2019.

Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. SMOTE: synthetic minority over-sampling technique. *J. Artif. Intell. Res.*, 16:321–357, 2002. doi: 10.1613/jair.953. URL `https://doi.org/10.1613/jair.953`.

Peng Chu, Xiao Bian, Shaopeng Liu, and Haibin 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*, pp. 694–710. Springer, 2020.

Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. Class-balanced loss based on effective number of samples. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9268–9277, 2019.

Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in neural information processing systems*, 26, 2013.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

C. Drummond and R. C. Holte. C4.5, class imbalance, and cost sensitivity: Why under-sampling beats over-sampling. *proc of the icml workshop on learning from imbalanced datasets ii*, 2003.

Jintong Gao, He Zhao, Zhuo Li, and Dandan Guo. Enhancing minority classes by mixing: An adaptative optimal transport approach for long-tailed classification. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 60329–60348. Curran Associates, Inc., 2023. URL `https://proceedings.neurips.cc/paper_files/paper/2023/file/bdabb5d4262bcfb6a1d529d690a6c82b-Paper-Conference.pdf`.

Jintong Gao, He Zhao, Dan dan Guo, and Hongyuan Zha. Distribution alignment optimization through neural collapse for long-tailed classification. In *Forty-first International Conference on Machine Learning*, 2024a. URL `https://openreview.net/forum?id=Hjwx3H6Vci`.

Jintong Gao, He Zhao, Zhuo Li, and Dandan Guo. Enhancing minority classes by mixing: An adaptative optimal transport approach for long-tailed classification. *Advances in Neural Information Processing Systems*, 36, 2024b.

Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *The Journal of Machine Learning Research*, 13(1):723–773, 2012.

Dandan Guo, Zhuo Li, He Zhao, Mingyuan Zhou, Hongyuan Zha, et al. Learning to re-weight examples with optimal transport for imbalanced classification. *Advances in Neural Information Processing Systems*, 35: 25517–25530, 2022a.

Dandan Guo, Zhuo Li, meixi zheng, He Zhao, Mingyuan Zhou, and Hongyuan Zha. Learning to re-weight examples with optimal transport for imbalanced classification. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 25517–25530. Curran Associates, Inc., 2022b. URL `https://proceedings.neurips.cc/paper_files/paper/2022/file/a39a9aceda771cded859ae7560530e09-Paper-Conference.pdf`.

Haibo He and Edwardo A Garcia. Learning from imbalanced data. *IEEE Transactions on knowledge and data engineering*, 21(9):1263–1284, 2009.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Dan Hendrycks, Mantas Mazeika, and Thomas Dietterich. Deep anomaly detection with outlier exposure. *arXiv preprint arXiv:1812.04606*, 2018.

Yan Hong, Jianfu Zhang, Zhongyi Sun, and Ke Yan. Safa: Sample-adaptive feature augmentation for long-tailed image classification. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXIV*, pp. 587–603. Springer, 2022.

Chen Huang, Yining Li, Chen Change Loy, and Xiaoou Tang. Learning deep representation for imbalanced classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5375–5384, 2016.

Bingyi Kang, Saining Xie, Marcus Rohrbach, Zhicheng Yan, Albert Gordo, Jiashi Feng, and Yannis Kalantidis. Decoupling representation and classifier for long-tailed recognition. *arXiv preprint arXiv:1910.09217*, 2019.

Jaehyung Kim, Jongheon Jeong, and Jinwoo Shin. M2m: Imbalanced classification via major-to-minor translation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 13896–13905, 2020.

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

Shuang Li, Kaixiong Gong, Chi Harold Liu, Yulin Wang, Feng Qiao, and Xinjing Cheng. Metasaug: Meta semantic augmentation for long-tailed visual recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 5212–5221, 2021.

Zhuo Li, He Zhao, Anningzhe Gao, Dandan Guo, Tsung-Hui Chang, and Xiang Wan. Prototype-oriented clean subset extraction for noisy long-tailed classification. *IEEE Transactions on Circuits and Systems for Video Technology*, pp. 1–1, 2025. doi: 10.1109/TCSVT.2025.3546031.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pp. 2980–2988, 2017.

Jialun Liu, Wenhui Li, and Yifan Sun. Memory-based jitter: Improving visual recognition on long-tailed data with diversity in memory. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 1720–1728, 2022.

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of the IEEE international conference on computer vision*, pp. 3730–3738, 2015.

Ziwei Liu, Zhongqi Miao, Xiaohang Zhan, Jiayun Wang, Boqing Gong, and Stella X Yu. Large-scale long-tailed recognition in an open world. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2537–2546, 2019.

Seulki Park, Youngkyu Hong, Byeongho Heo, Sangdoo Yun, and Jin Young Choi. The majority can help the minority: Context-rich minority oversampling for long-tailed classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6887–6896, 2022.

Gabriel Peyré, Marco Cuturi, et al. Computational optimal transport: With applications to data science. *Foundations and Trends® in Machine Learning*, 11(5-6):355–607, 2019.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PMLR, 2021.

Jiawei Ren, Cunjun Yu, Xiao Ma, Haiyu Zhao, Shuai Yi, et al. Balanced meta-softmax for long-tailed visual recognition. *Advances in neural information processing systems*, 33:4175–4186, 2020.

Maria L Rizzo and Gábor J Székely. Energy distance. *wiley interdisciplinary reviews: Computational statistics*, 8(1):27–38, 2016.

Sebastian Ruder. An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*, 2016.

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*, 115:211–252, 2015.

Li Shen, Zhouchen Lin, and Qingming Huang. Relay backpropagation for effective learning of deep convolutional neural networks. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part VII 14*, pp. 467–482. Springer, 2016.

Jong-Chyi Su, Zezhou Cheng, and Subhransu Maji. A realistic evaluation of semi-supervised learning for fine-grained classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12966–12975, 2021.

Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

Jason Van Hulse, Taghi M Khoshgoftaar, and Amri Napolitano. Experimental perspectives on learning from imbalanced data. In *Proceedings of the 24th international conference on Machine learning*, pp. 935–942, 2007.

Jianfeng Wang, Thomas Lukasiewicz, Xiaolin Hu, Jianfei Cai, and Zhenghua Xu. Rsg: A simple but effective module for learning imbalanced datasets. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3784–3793, 2021.

Hongxin Wei, Lue Tao, Renchunzi Xie, Lei Feng, and Bo An. Open-sampling: Exploring out-of-distribution data for re-balancing long-tailed datasets. In *International Conference on Machine Learning*, pp. 23615–23630. PMLR, 2022.

Yuguang Yan, Mingkui Tan, Yanwu Xu, Jiezhang Cao, Michael Ng, Huaqing Min, and Qingyao Wu. Oversampling for imbalanced data via optimal transport. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 5605–5612, 2019.

Lu Yang, He Jiang, Qing Song, and Jun Guo. A survey on long-tailed visual recognition. *International Journal of Computer Vision*, 130(7):1837–1872, 2022.

Yuzhe Yang and Zhi Xu. Rethinking the value of labels for improving class-imbalanced learning. *Advances in neural information processing systems*, 33:19290–19301, 2020.

Bo Zhao and Hakan Bilen. Dataset condensation with distribution matching. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 6514–6523, 2023.

Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE transactions on pattern analysis and machine intelligence*, 40(6): 1452–1464, 2017.

Boyan Zhou, Quan Cui, Xiu-Shen Wei, and Zhao-Min Chen. Bbn: Bilateral-branch network with cumulative learning for long-tailed visual recognition. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Aug 2020. doi: 10.1109/cvpr42600.2020.00974. URL `http://dx.doi.org/10.1109/cvpr42600.2020.00974`.

Appendix For Synthesizing Minority Samples for Long-tailed Classification via Distribution Matching

A Alternatives for $\mathbf{Dist}_\theta(P_k, Q_k)$

A.1 Maximum Mean Discrepancy (MMD)

Preliminaries. MMD is an effective non-parametric metric for comparing the distributions based on two sets of data (Gretton et al., 2012), where the general MMD between two distributions P and Q is defined as

$$\mathrm{MMD}^2(P, Q) = \sup_{\|\phi\|_{\mathcal{H}} \leq 1} \left\| \mathbb{E}_{x\sim P}\left[\phi\left(x\right)\right] - \mathbb{E}_{y\sim Q}\left[\phi\left(y\right)\right] \right\|_{\mathcal{H}}^2, \tag{10}$$

where $\mathbb{E}_{x\sim P}[\cdot]$ denotes the expectation with regard to the distribution P, ϕ is the embedding function, and $\|\phi\|_{\mathcal{H}} \leq 1$ defines a set of functions in the unit ball of a reproducing kernel Hilbert space (RKHS) $\mathcal{H}$.

Define $\mathbf{Dist}_\theta(P_k, Q_k)$ with MMD. As we do not have access to ground-truth data distributions for synthetic and real samples shown in Eq. 10, we can use a biased empirical estimate of the MMD by replacing the population expectations with empirical expectations (Gretton et al., 2012), which are computed on the synthetic and real samples in P_k and Q_k and denoted as

$$\mathrm{Dist}_\theta(P_k, Q_k) = \left\| \frac{1}{N_k} \sum_{n=1}^{N_k} \psi_\theta\left(x_n\right) - \frac{1}{M_k} \sum_{m=1}^{M_k} \psi_\theta\left(\hat{x}_m\right) \right\|^2 \tag{11}$$

A.2 Energy Distance (ED)

Preliminaries. Drawing inspiration from the concept of potential energy between objects in a gravitational field, Energy Distance (ED) (Rizzo & Székely, 2016) measures the similarity between two probability distributions, P and Q. This can be mathematically expressed as follows:

$$\begin{aligned} \mathrm{ED}^2(P, Q) = {} & 2\mathbb{E}_{x\sim P, y\sim Q}||\phi(x) - \phi(y)|| \\ & - \mathbb{E}_{x\sim P}||\phi(x) - \phi(x')|| \\ & - \mathbb{E}_{y\sim Q}||\phi(y) - \phi(y')||, \end{aligned} \tag{12}$$

where $\mathbb{E}_{x\sim P}[\cdot]$ denotes the expectation with respect to the distribution P and $||\cdot||$ denotes the Euclidean norm (length) of its argument. In addition, x' and y' are independent copies of x and y, respectively.

Define $\mathbf{Dist}_\theta(P_k, Q_k)$ with ED. Here, we can define $\mathrm{Dist}_\theta(P_k, Q_k)$ based on the energy distance (ED) as follows:

$$\begin{aligned} \mathrm{Dist}_\theta(P_k, Q_k) = {} & \frac{2}{N_k M_k} \sum_{n=1}^{N_k} \sum_{m=1}^{M_k} ||\psi_\theta(x_n) - \psi_\theta(\hat{x}_m)||^2 \\ & - \frac{1}{{N_k}^2} \sum_{n,m=1}^{N_k} ||\psi_\theta(x_n) - \psi_\theta(x_m)||^2 \\ & - \frac{1}{{M_k}^2} \sum_{n,m=1}^{M_k} ||\psi_\theta(\hat{x}_n) - \psi_\theta(\hat{x}_m)||^2 \end{aligned} \tag{13}$$

Algorithm 2: Oversampling minority samples via our framework (Out-of-Distribution).

Input : Dataset $\mathcal{D}_{\text{train}}$ and $\mathcal{D}_{\text{ood}}$, classifier f and pre-trained classifier g, a confusion matrix $\mathbf{A}$, hyper-parameters:$\{\lambda_1,\ \lambda_2,\ \gamma,\ \eta,\ T,\ \tau,\ \beta\}$

Output: A class-balanced dataset $\mathcal{D}_{\text{bal}}$

Initialize $\mathcal{D}_{\text{bal}} \leftarrow \mathcal{D}_{\text{train}}$;

Randomly sample a balanced subset from $\mathcal{D}_{\text{train}}$ and obtain confusion matrix A by evaluate the pre-trained model g using this subset;

for $k = 2, ..., K$ **do**

    Compute $M_k \leftarrow N_1 - N_k$; Initialize $\hat{\mathcal{D}}_k \leftarrow \emptyset$;

    for $m = 1, ..., M_k$ **do** `// Step 1. Sample selection for` x_0

        Sample a x_0 from $\mathcal{D}_{\text{ood}}$ randomly;

        Initialize $\hat{x}_m \leftarrow x_{0,m} + \epsilon_m$ with a standard Gaussian noise ϵ_m, then update $\hat{\mathcal{D}}_k \leftarrow \hat{\mathcal{D}}_k \cup \{(\hat{x}_m, k)\}$;

    end

    Use $\hat{\mathcal{D}}_k$ to build Q_k and real training samples D_k in $\mathcal{D}_{\text{train}}$ to build P_k ;

    Obtain k_c if $A_{k,k_c} \geq A_{k,i}$ where $i \in [1, K]$ and $i \neq k$;

    for $t = 1, ..., T$ **do** `// Step 2. Optimization for` $\hat{x}$

        Update $\hat{\mathcal{D}}_k$ by minimizing $\sum_{m=1}^{M_k} [\mathcal{L}(g(\hat{x}_m), k) + \lambda_1 \cdot f_{k_c}(\hat{x}_m)] + \lambda_2 \cdot \text{Dist}_\theta(P_k, Q_k)$, where $\theta \sim P_\theta$;

    end

    for $\hat{x}_m$ *in* $\hat{\mathcal{D}}_k$ **do** `// Step 3. Sample rejection for` $\hat{x}$

        if $\mathcal{L}(g(\hat{x}_m), k) \geq \tau$ *or Reject* $= 1$ **then**

            $\hat{x}_m \leftarrow$ with a random sample from class k in $\mathcal{D}_{\text{train}}$;

        end

        Update $\mathcal{D}_{\text{bal}} \leftarrow \mathcal{D}_{\text{bal}} \cup \{(\hat{x}_m, k)\}$;

    end

end

B Algorithms of our framework

In this section, we give the algorithm processes 2 of our method under OOD settings as shown in Alg. 2.

C More details about datasets and experiments

C.1 Settings and Training details

Unless otherwise stated, we set the imbalance factor as IF $= N_1/N_K$ and use $T = 5$ iterations with a step size of $\eta = 0.1$ to optimize the synthetic samples at each training iteration. The hyper-parameter for the OT entropy constraint is $\gamma = 0.1$ and the maximum iteration number in the Sinkhorn algorithm is 200. We use SGD with momentum 0.9 and weight decay $5e^{-4}$ and conduct all the experiments on 8 Tesla-V100 GPUs.

C.2 CIFAR-10 and CIFAR-100 datasets

CIFAR-LT-10 / CIFAR-LT-100. The original CIFAR-10/CIFAR-100 datasets (Krizhevsky et al., 2009) include 60,000 images and 10/100 classes with a size of 32×32, where there are 50,000 images for training and 10,000 for testing. By following (Kim et al., 2020), we create CIFAR-LT-10 and CIFAR-LT-100 by randomly under-sampling in the original datasets with IF $= \{200, 100, 50\}$. We use the original test dataset to evaluate our method. **Training details.** Following (Kim et al., 2020; Li et al., 2021; Guo et al., 2022a), we use ResNet-32 (He et al., 2016) as the backbone. We employ 200 epochs for training f with an initial learning rate α of 0.1, which is decayed by $1e^{-2}$ at 160-th epoch and 180-th epoch. We set batch size as 32 and start our method at 160-th epoch, where we set λ_1 and λ_2 as 0.5, β as 0.999 and τ as 0.9.

C.3 ImageNet-LT and Places-LT

ImageNet-LT. The original ImageNet-2012 dataset (Deng et al., 2009) includes 1,281,167 images and 1000 classes with a max size of 1300×732. By following (Kim et al., 2020; Li et al., 2021; Liu et al., 2019), we create ImageNet-LT with 115.8K samples in 1000 classes and IF = 1280/5. We adopt the original validation dataset to test our method.

Places-LT The original Places-365 dataset(Zhou et al., 2017) includes 1,803,460 images and 365 classes with a max size of 5000×3068. By following (Liu et al., 2019), we create Places-LT with 62.5K samples in 1000 classes and the imbalance factor IF = 4980/5. We adopt the original test dataset to test our method.
Training details. Following previous works (Kim et al., 2020; Li et al., 2021; Kang et al., 2019), we use ResNet-50 as the backbone for ImageNet-LT. We employ 200 epochs for training f with an initial learning rate α as 0.1, which will be decayed by $1e^{-1}$ at the 160-th epoch and 180-th epoch. For Places-LT, we employ ResNet-152 pre-trained on the full ImageNet dataset (Russakovsky et al., 2015) as the backbone following (Guo et al., 2022a; Li et al., 2021). We set 200 epochs for training f with an initial learning rate α as 0.1, decayed by $1e^{-1}$ every 40 epochs. We start our method at 160-th epoch for ImageNet-LT and 90-th for Places-LT. We set λ_1 and λ_2 as 0.5, β as 0.999 and τ as 0.3. For all experiments, we initialize batch size as 64 and set it as 32 after deploying our method for training stability.

C.4 Training details about pre-trained model g

CIFAR-LT-10 / CIFAR-LT-100. For CIFAR-LT-10 / CIFAR-LT-100, we use ResNet-32 (He et al., 2016) as backbone network for pre-training. We employ 200 epochs for training g with an initial learning rate α of 0.1, which will be decayed by $1e^{-2}$ at 160th epoch and 180th epoch. We use SGD with momentum 0.9 and weight decay $5e^{-4}$ and set batch size as 128. In the first 160 epochs, we use the original imbalanced dataset to train the model g. For the last 40 epochs, we use the vanilla over-sample technique by inverse class frequency to further train the model g. We save the best checkpoint as our pre-trained model g.

ImageNet-LT & Places-LT For ImageNet-LT, we use ResNet-50 (He et al., 2016) as backbone network for pre-training. We employ 200 epochs for training g with an initial learning rate α of 0.1, which will be decayed by $1e^{-2}$ at 160th epoch and 180th epoch. For Places-LT, we employ ResNet-152 pre-trained on the full ImageNet dataset. We use 120 epochs for training g with an initial learning rate α as 0.1, which is decayed by $1e^{-1}$ every 10 epochs. For both datasets, we use SGD with momentum 0.9 and weight decay $5e^{-4}$ and set batch size as 512. Similar to CIFAR-LT-10 / CIFAR-LT-100, before the $160-th$ and $90-th$ epoch on ImageNet-LT and Places-LT, we use the original imbalanced dataset to train model g. For the last epochs, we adopt the vanilla over-sample technique by inverse class frequency (Drummond & Holte, 2003) to further train g. We save the best checkpoint as our pre-trained model g.

C.5 Different choice of OOD dataset

We further examine the impact of domain gap on Out-of-Distribution (OOD) scenarios, using medical images and pure noise as synthetic sample initializations. We use OrganAMNIST (Bilic et al., 2023) as the OOD dataset for CIFAR-LT-10. Results show that medical images outperformed the vanilla cross-entropy baseline, confirming our method's effectiveness. However, they underperform than the pure noise, likely due to the large domain gap between medical images and CIFAR-LT-10. Both are inferior to random images, indicating that the final performance is influenced by the domain gap between the OOD and target datasets. We find natural images, which are not only closer to our target dataset but also more accessible than medical images, to be a more effective choice. When we use natural images as the initialization, we achieve significant performance improvements, demonstrating that a suitable choice of OOD dataset can enhance performance. In our case, the downstream task in the long-tailed recognition benchmark usually is natural image classification, making it beneficial to use an OOD dataset similar to natural images. We believe that natural images from the OOD dataset, beyond initialization, share similar textures, styles, and color information, aiding the model's generalization ability by furnishing minority classes with their lacked information.

Table 9: Test top-1 errors (%) of ResNet-32 on CIFAR-LT-10 with IF $\in [200, 100, 50]$ and different OOD datasets. *Distribution* means the initialization of to-be-learned synthetic samples $\hat{x}$, e.g., *OOD* denotes we initialize $\hat{x}$ with OOD samples. *Domain* indicates the corresponding domain of OOD dataset.

Distribution	Domain	200	100	50
-	CE-Baseline	34.13	29.86	25.06
OOD	Pure Noise	23.93	20.40	16.75
OOD	Medical	23.95	20.95	17.17
OOD	Natural	**23.43**	**20.03**	**16.39**

C.6 Classification confidence of synthetic sample on target classifier f.

To prove the effectiveness of our method in enhancing the generation of high-quality synthetic samples, we compare the classification performance of the synthetic samples on the target classifier f for CIFAR-LT-10 (ID) with IF = 100 on ResNet-32. Specifically, we use the target classifier to output the probability of the synthetic samples in class k being correctly predicted as the k class, where we consider $k \in \{2, \cdots, 10\}$. Then we compute the average probability for all samples and express it as a percentage. Compared to M2m, Fig. 4 illustrates that the synthetic samples generated by our method (based on OT) have higher classification confidence for the corresponding concerned class during the training of target classifier f, in both ID and OOD settings. This finding suggests that the our synthetic samples can be predicted correctly. To prove our effectiveness in enhancing the generation of high-quality synthetic samples, we compare the classification performance of the synthetic samples on the target classifier f for CIFAR-LT-10 (ID) with IF = 100 on ResNet-32. Specifically, we use the target classifier to output the probability of the synthetic samples in class k being correctly predicted as the k class, where we consider $k \in \{2, \cdots, 10\}$. Then we compute the average probability for all samples and express it as a percentage. Compared to M2m, Fig. 4 illustrates that the synthetic samples generated by our method (based on OT) have higher classification confidence for the corresponding concerned class during the training of target classifier f, in both ID and OOD settings. This finding suggests that the generated samples by ours are more credible.

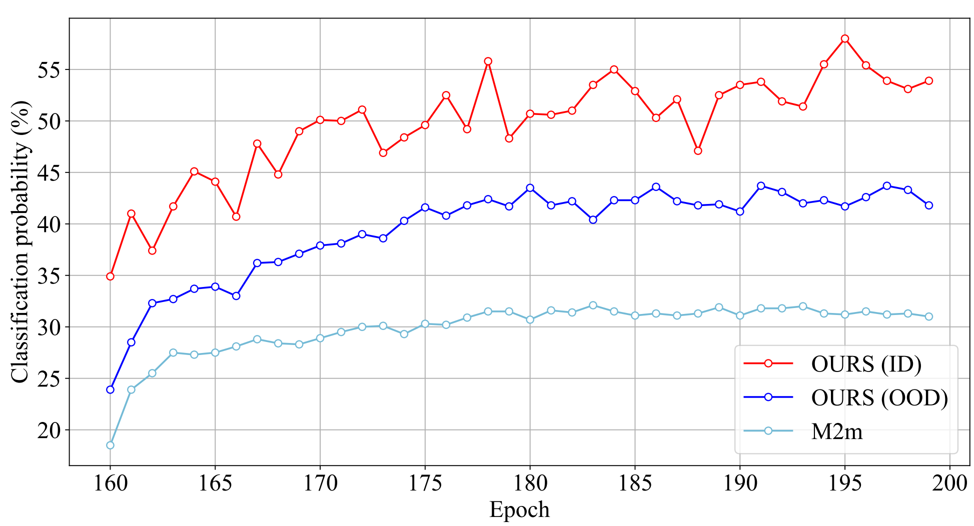


Figure 4: Probability of synthetic samples that are correctly classified as the concerned class k on target classifier f.

C.7 Influence of pre-trained model g

In this section, we investigate whether the effectiveness of the proposed method is affected by the performance of pre-trained g. We can see that the better the performance of the pre-trained classifier, the better the final classification result of ours in general. Although the pre-trained classifier in the 100th epoch and the best-performed classifier have different classification performance, they have similar impact on the final classification result. Besides, the ensemble of three best-performed classifiers achieves better performance than only using one best-performed classifier. In our work, we perform the experiments only using one best-performed classifier. It demonstrates that the final classification results will increase if we use the ensemble of the pre-trained classifiers to optimize the synthetic samples.

Table 10: Test top-1 errors (%) on CIFAR-LT-10 with IF = 100 under the in-distribution setting. *Pre-trained performance* indicates the overall performance on the g and *Final performance* is the corresponding final classification result.

g	Pre-trained performance	Final performance
1-th epoch	66.51	21.16
3-th epoch	59.75	20.23
10-th epoch	44.26	20.17
20-th epoch	41.71	19.07
100-th epoch	30.95	18.33
Best epoch	28.17	18.37
Ensemble	28.77, 30.23, 29.01	**18.01**

C.8 Visualization of confusion matrix

To demonstrate the effectiveness of our method in improving the performance of minority classes, we visualize the confusion matrices of CE, M2m and OURS on CIFAR-LT-10 with IF=200. As shown in Fig.5, CE has poor classification performance in minority classes. Therefore, it is necessary to solve the long-tailed problem. Although M2m mitigates the problem, it still performs poorly on the rarest classes. Our proposed method achieves better performance than CE and M2m. In particular, ours is superior to strong baseline M2m for almost every class, thereby alleviating the imbalanced classification problem.

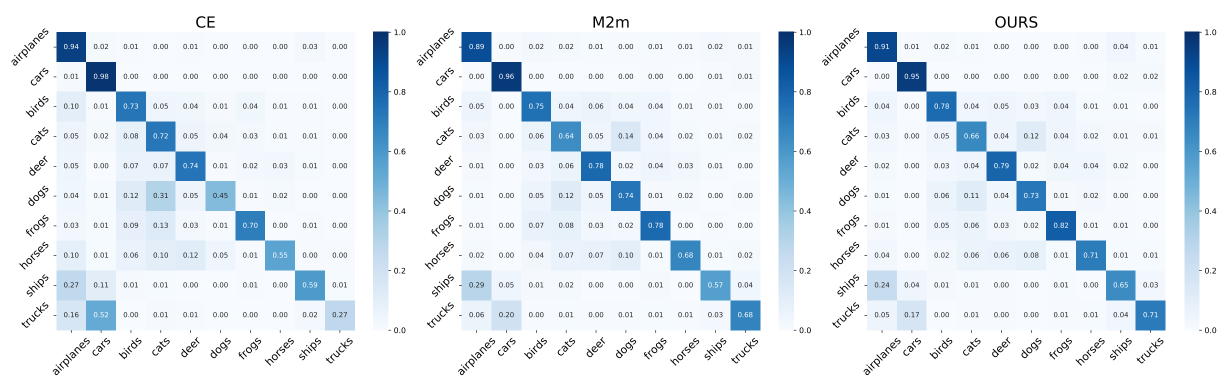


Figure 5: Confusion matrices of the CE, M2m and OURS on CIFAR-LT-10 with the imbalance factor 200.