Supplementary Materials

A. Full demonstration for CBDM

Proposition 1. *When training a diffusion model parameterized with θ on a class-imbalanced dataset, its conditional reverse distribution $p_\theta\left(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, y\right)$ can be corrected with an adjustment schema:*

$$p_\theta^\star(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, y) = p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, y)\frac{p_\theta(\boldsymbol{x}_{t-1})}{p_\theta^\star(\boldsymbol{x}_{t-1})}\frac{q^\star(\boldsymbol{x}_t)}{q(\boldsymbol{x}_t)} \tag{1}$$

Proof. The starting point for this derivation comes from [1]. By noting the conditional prior distribution given the image label as $\hat{q}$, we can write the reverse conditional probability $\hat{q}\left(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, y\right)$ as

$$\hat{q}\left(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, y\right) = \frac{q\left(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t\right)\hat{q}\left(y|\boldsymbol{x}_{t-1}\right)}{\hat{q}\left(y|\boldsymbol{x}_t\right)} \tag{2}$$

With the Bayesian formula, the equation can be transformed into

$$\hat{q}(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, y) = \frac{q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)\hat{q}(y|\boldsymbol{x}_{t-1})}{\hat{q}(y|\boldsymbol{x}_t)} \tag{3}$$

$$= \frac{q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)\hat{q}(\boldsymbol{x}_{t-1}|y)\hat{q}(y)\hat{q}(\boldsymbol{x}_t)}{\hat{q}(\boldsymbol{x}_t|y)\hat{q}(y)\hat{q}(\boldsymbol{x}_{t-1})} \tag{4}$$

$$= \frac{q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)\hat{q}(\boldsymbol{x}_{t-1}|y)\hat{q}(\boldsymbol{x}_t)}{\hat{q}(\boldsymbol{x}_t|y)\hat{q}(\boldsymbol{x}_{t-1})} \tag{5}$$

In the above equation, $q(\boldsymbol{x}_t)$ is equal to $\hat{q}(\boldsymbol{x}_t)$ according to the derivation of [1]. Moreover, since the conditional diffusion model is trained to fit a prior distribution with known conditions by definition, we can approximate $\hat{q}(\boldsymbol{x}_{t-1})$ with $p_\theta(\boldsymbol{x}_{t-1})$ in the following calculations. Thus, we have

$$p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, y) = \frac{q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)\hat{q}(\boldsymbol{x}_{t-1}|y)q(\boldsymbol{x}_t)}{\hat{q}(\boldsymbol{x}_t|y)p_\theta(\boldsymbol{x}_{t-1})} \tag{6}$$

We use $\star$ to stand for an optimal distribution, for instance, when the categories are evenly distributed and $\frac{q(y)}{p_\theta(y)}$ is correctly estimated. In this case, we have:

$$p_\theta^\star(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, y) = \frac{q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)\hat{q}^\star(\boldsymbol{x}_{t-1}|y)q^\star(\boldsymbol{x}_t)}{\hat{q}^\star(\boldsymbol{x}_t|y)p_\theta^\star(\boldsymbol{x}_{t-1})} \tag{7}$$

Here in Eqn. (6), $\hat{q}\left(\boldsymbol{x}_t|y\right)$ and $\hat{q}(\boldsymbol{x}_{t-1}|y)$ are conditional on the distribution of the label y and thus are not affected by the label distribution. Then, the only term that is under the influence of imbalanced label distribution is $p_\theta\left(\boldsymbol{x}_{t-1}\right)$ and $q\left(\boldsymbol{x}_t\right)$. For the optimal label distribution defined with $\star$, we follow the same deduction. Dividing the two equations yields:

$$p_\theta^\star(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, y) = p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, y)\frac{p_\theta(\boldsymbol{x}_{t-1})}{p_\theta^\star(\boldsymbol{x}_{t-1})}\frac{q^\star(\boldsymbol{x}_t)}{q(\boldsymbol{x}_t)} \tag{8}$$

□

From post-hoc adjustment to training with adjustment The above Eqn. (8) shows how to transform the original model estimation $p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, y)$ into $p^\star_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, y)$ by adding the adjustment term $\frac{p_\theta(\boldsymbol{x}_{t-1})}{p^\star_\theta(\boldsymbol{x}_{t-1})}\frac{q^\star(\boldsymbol{x}_t)}{q(\boldsymbol{x}_t)}$. Naturally, the most direct way to adjust the distribution is to implement this term in a post-hoc manner during sampling.

First, we note that $\frac{p_\theta(\boldsymbol{x}_{t-1})}{p^\star_\theta(\boldsymbol{x}_{t-1})}$ cannot be obtained directly, but can be approximated by the conditional expectation of the model. For example, we can rewrite it as the conditional expectation of $p^\star_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, y)$ successively about the labels y and $\boldsymbol{x}_t$, and later estimate it by Monte Carlo sampling. However, the use of Monte Carlo sampling during image generation imposes much more computational burden, and the image generation will also be inaccurate due to the large estimation error caused by the difficulty to sample uniformly on $\boldsymbol{x}_t$ and on y. But the most problematic issue is that the adjustment term also contains the true probability of the data distribution $\frac{q^\star(\boldsymbol{x}_t)}{q(\boldsymbol{x}_t)}$, which obviously cannot be estimated. So the presence of this term makes the post hoc adjustment method impossible to implement. In contrast, following the idea of trainable logit adjustment in the long-tail recognition task [4], we found that these two problems can be solved if p_θ is adjusted during training.

First, $\frac{q^\star(\boldsymbol{x}_t)}{q(\boldsymbol{x}_t)}$ is independent of model parameters, and can be thus neglected in the following calculation, which yields a simpler result:

$$p^\star_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, y) = p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, y)\frac{p_\theta(\boldsymbol{x}_{t-1})}{p^\star_\theta(\boldsymbol{x}_{t-1})} \tag{9}$$

Second, similar to the logit adjustment in long-tail recognition, our first step in adjusting the distribution during training lies in reversing the sign before the adjustment term. Since we are considering the gradient of the log probability, changing the positive and negative sign is actually equivalent to reversing the adjustment term $\frac{p_\theta(\boldsymbol{x}_{t-1})}{p^\star_\theta(\boldsymbol{x}_{t-1})}$. Thus, we have instead:

$$p^\star_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, y) = p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, y)\frac{p^\star_\theta(\boldsymbol{x}_{t-1})}{p_\theta(\boldsymbol{x}_{t-1})} \tag{10}$$

Proposition 2. *For the adjusted loss $\mathcal{L}^\star_{DM} = \sum_{t=1}^{T}\mathcal{L}^\star_{t-1}$, an upper-bound of the target training objective to calibrate at timestep t (i.e. $\mathcal{L}^\star_{t-1}$) can be derived as:*

$$\begin{aligned}\sum_{t\geq 1}\mathcal{L}^\star_{t-1} &= \sum_{t\geq 1} D_{\mathrm{KL}}[q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0)\ ||\ p^\star_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, y)] \\ &\leq \sum_{t\geq 1}[\underbrace{D_{\mathrm{KL}}[q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0)\ ||\ p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, y)]}_{\textit{Diffusion model loss } \mathcal{L}_{DM}} \\ &\quad + \underbrace{t\mathbb{E}_{y'\sim q^\star_{\mathcal{Y}}}[D_{\mathrm{KL}}[p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)||p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, y')]}_{\textit{Distribution adjustment loss } \mathcal{L}_r}]],\end{aligned}$$

where $q^\star_{\mathcal{Y}}$ is the target label distribution to adjust, *e.g.,* a class-balanced label distribution.

Proof. In the proposition, we admit that the adjustment weight (i.e. the regularization weight) is 1. In the deduction, we further denote τ as the regularization weight, which makes the adjusted probability becomes:

$$p^\star_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t) = p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, y)\frac{p^\star_\theta(\boldsymbol{x}_{t-1})^\tau}{p_\theta(\boldsymbol{x}_{t-1})^\tau} \tag{11}$$

Thereafter, by bringing $p^\star_\theta$ into L_{t-1} of the DDPM and simplifying the formula, we have.

$$\sum_{t\geq 1}L^\star_{t-1} = \mathbb{E}_q[-\sum_{t\geq 1} log\frac{p^\star_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, y)}{q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)}] \tag{12}$$

$$= \mathbb{E}_q[-\sum_{t\geq 1} log\frac{p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, y)}{q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)}\frac{p^\star_\theta(\boldsymbol{x}_{t-1})^\tau}{p_\theta(\boldsymbol{x}_{t-1})^\tau}] \tag{13}$$

$$= \mathbb{E}_q[-\sum_{t\geq 1}(log\frac{p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, y)}{q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)} - log\frac{p^\star_\theta(\boldsymbol{x}_{t-1})^\tau}{p_\theta(\boldsymbol{x}_{t-1})^\tau})] \tag{14}$$

$$= \mathbb{E}_q[-\sum_{t\geq 1} log\frac{p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, y)}{q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)}] + \tau\mathbb{E}_q[-\sum_{t\geq 1} log\frac{p^\star_\theta(\boldsymbol{x}_{t-1})}{p_\theta(\boldsymbol{x}_{t-1})}] \tag{15}$$

In the above derivation: from Eqn. (13) to equation Eqn. (14), we split the product in log into the summation; from Eqn. (14) to Eqn. (15), the summation is apportioned into two terms, after which the exponent τ in log is extracted. We note that the expectation of the divisor of the two probabilities in the log-likelihood is equal to the KL divergence of both, i.e:

$$\mathbb{E}_q[-log\frac{p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t,y)}{q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)}] = D_{KL}[q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t,\boldsymbol{x}_0)\ ||\ p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t,y)] \tag{16}$$

Thus, we have:

$$\sum_{t\geq 1} L^\star_{t-1} = \mathbb{E}_q[-\sum_{t\geq 1} log\frac{p^\star_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t,y)}{q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)}] \tag{17}$$

$$= \sum_{t\geq 1} D_{KL}[q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t,\boldsymbol{x}_0)\ ||\ p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t,y)] + \tau\sum_{t\geq 1}\mathbb{E}_q[-log\frac{p^\star_\theta(\boldsymbol{x}_{t-1})}{p_\theta(\boldsymbol{x}_{t-1})}] \tag{18}$$

$$= \sum_{t\geq 1} D_{KL}[q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t,\boldsymbol{x}_0)\ ||\ p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t,y)] + \tau\sum_{t\geq 1}\mathbb{E}_q[-\sum_{t'\geq t} log\frac{p^\star_\theta(\boldsymbol{x}_{t'-1}|\boldsymbol{x}_{t'})}{p_\theta(\boldsymbol{x}_{t'-1}|\boldsymbol{x}_{t'})}] \tag{19}$$

$$= \sum_{t\geq 1} D_{KL}[q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t,\boldsymbol{x}_0)\ ||\ p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t,y)] + \tau\sum_{t\geq 1} t\mathbb{E}_q[-log\frac{p^\star_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)}{p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)}] \tag{20}$$

$$= \sum_{t\geq 1} D_{KL}[q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t,\boldsymbol{x}_0)\ ||\ p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t,y)] + \tau t D_{KL}(p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)||p^\star_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)) \tag{21}$$

where, from Eqn. (17) to Eqn. (18), we rewrite the first term in the form of KL divergence. From Eqn. (18) to Eqn. (19), we decompose $p_\theta(\boldsymbol{x}_{t-1})$ into $q(\boldsymbol{x}_t)\Pi_{t<t'\leq T}p_\theta(\boldsymbol{x}_{t'-1}|\boldsymbol{x}_{t'})$ in the form of a Markov chain, while the continuous multiplication is converted into the form of a logit sum; thereafter, we shift the summation symbols inside the expectation outward and rewrite the expectation again into the form of KL divergence. For the derivation of Eqn. (18) to Eqn. (19), we further describe in detail as follows:

$$\mathbb{E}_q[-log\frac{p^\star_\theta(\boldsymbol{x}_{t-1})}{p_\theta(\boldsymbol{x}_{t-1})}] = \mathbb{E}_q[-log\frac{q(\boldsymbol{x}_T)\Pi_{t<t'\leq T}p^\star_\theta(\boldsymbol{x}_{t'-1}|\boldsymbol{x}_{t'})}{q(\boldsymbol{x}_T)\Pi_{t<t'\leq T}p_\theta(\boldsymbol{x}_{t'-1}|\boldsymbol{x}_{t'})}] \tag{22}$$

$$= \mathbb{E}_q[-\sum_{t<t'\leq T} log\frac{p^\star_\theta(\boldsymbol{x}_{t'-1}|\boldsymbol{x}_{t'})}{p_\theta(\boldsymbol{x}_{t'-1}|\boldsymbol{x}_{t'})}] \tag{23}$$

$$\tag{24}$$

When $\tau = 1$, we obtain the proposition itself; else, we obtain the CBDM algorithm with the hyper-parameter regularization weight τ. □

Loss function Based on the above propositions, we derive the final loss form as follows:

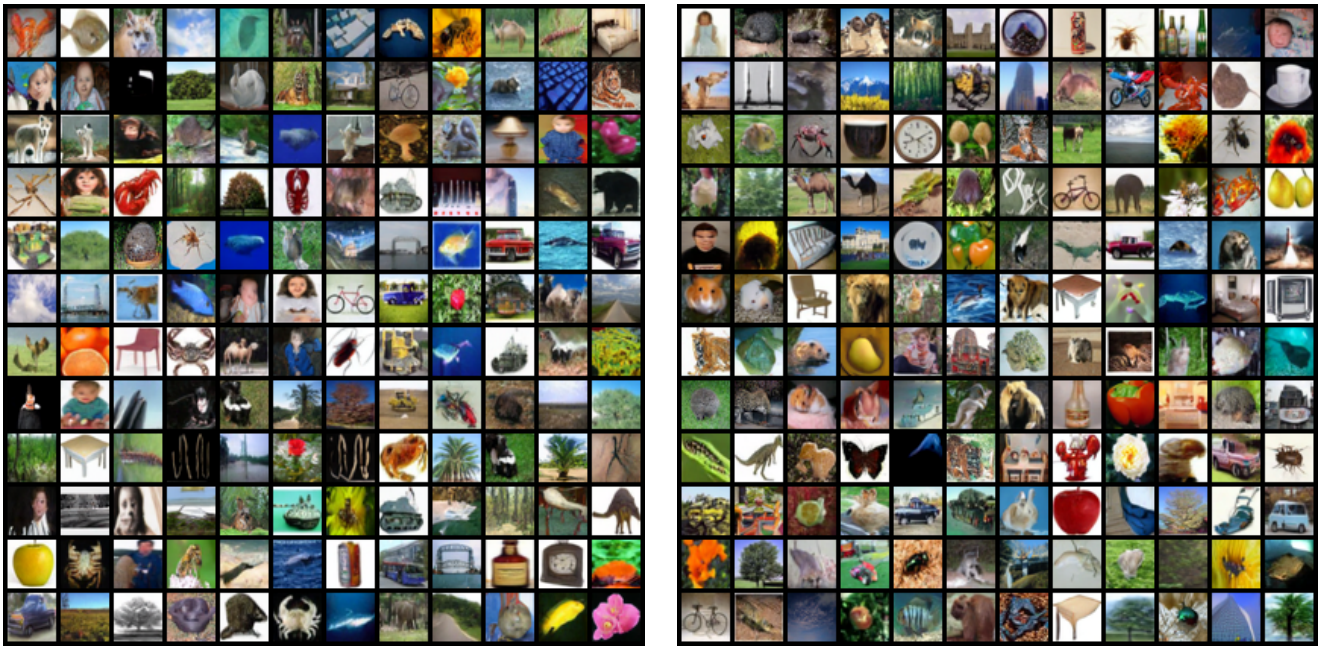
$$\mathcal{L}_{CBDM} = \mathcal{L}_{DM} + \mathcal{L}_r \tag{25}$$

where:

$$\begin{cases} \mathcal{L}_{DM} = & D_{KL}(q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t,\boldsymbol{x}_0)\ ||\ p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)) \\ \mathcal{L}_r = & \tau t D_{KL}(p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)||p^\star_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)) \end{cases} \tag{26}$$

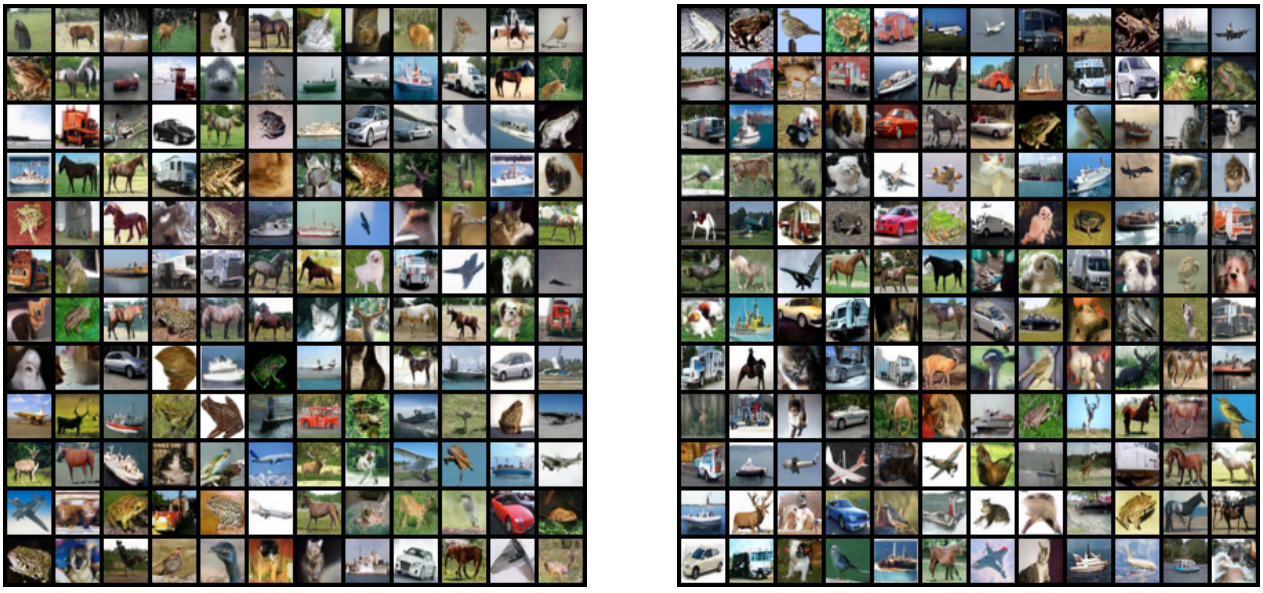
Again, the form of $\mathcal{L}_{DM}$ is exactly the same as the original loss in DDPM or other common diffusion models, so it is called $\mathcal{L}_{DM}$. The second term acts similarly to the regularization and is therefore called $\mathcal{L}_r$. Now, we simplify $\mathcal{L}_r$ by first writing $p^\star_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)$ as the expectation of the conditional probability about y' realized through a simple Monte-Carlo sampling, and note the optimal distribution that we want to approximate as $q^\star_{\mathcal{Y}}$, we have:

$$\mathcal{L}_{\mathrm{r}}(\boldsymbol{x}_t,y,t) = \mathbb{E}_{y'\sim q^\star_{\mathcal{Y}}}[t||\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t,y) - \boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t,y')||^2], \tag{27}$$



(a) DDPM (b) CBDM

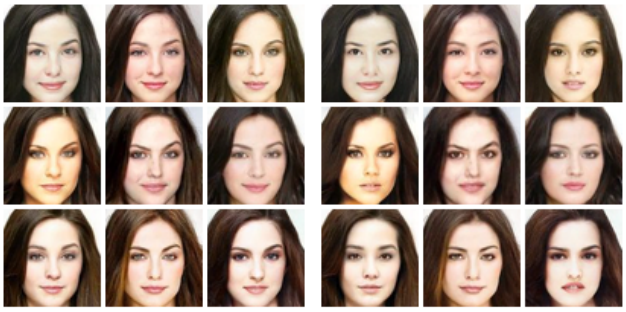
Figure 1. Image generation visualization for the CIFAR100LT dataset



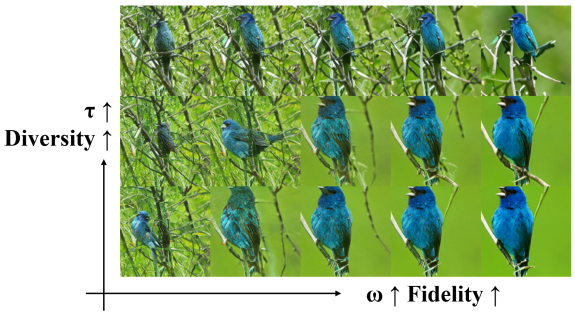
(a) DDPM (b) CBDM

Figure 2. Image generation visualization for the CIFAR10LT dataset

We note that the method can actually be combined with any diffusion model by simply adding the loss $\mathcal{L}_r$ to the DDPM algorithm. The method can adjust the distribution naturally in the iterations of the training process. Therefore, this method avoids the problem of time-consuming Monte Carlo sampling estimation of the conditional expectation in the post-hoc adjustment method, and it also avoids the problem of estimating the true prior distribution $q^{\star}(\boldsymbol{x}_t)$ by directly eliminating terms that are unrelated to the model parameters.



(a) DDPM(left)/CBDM(right) comparison



(b) Controllability on CUB dataset

Figure 3. (a) DDPM(left)/CBDM(right) comparison when denoising a same noised image in CelebA-5. (b) An exemplar about the fidelity and diversity control guided by strength ω and regularization weight τ for generators trained on the CUB dataset.

B. Additional Qualitative Results

Mode collapse caused by an inappropriate $\mathcal{Y}$ In our experiments, we found that using the theoretically optimal sampling set $\mathcal{Y}^{bal}$ often leads to mode collapse in tail class images, thus we used a more imbalanced label set $\mathcal{Y}$. We visualize the results under different label set $\mathcal{Y}$ in order to better explain this issue. In Figure 4, it can be observed that an imbalanced set $\mathcal{Y}^{train}$ ($\mathcal{Y}^{lt}$) demonstrates a better diversity while preserving the original class information of class 62. On the contrary, $\mathcal{Y}^{sqrt}$ and $\mathcal{Y}^{bal}$ demonstrate a more and more severe mode collapse issue when the set becomes more balanced.

(a) DDPM (b) CBDM($\mathcal{Y}^{train}$) (c) CBDM($\mathcal{Y}^{sqrt}$) (d) CBDM($\mathcal{Y}^{bal}$)

Figure 4. Comparison of image generation results on body class (62) between different label set. Image generated by DDPM is also shown for comparison.

C. Additional Experimental Results

Dataset	Model	ω	τ	FID↓	F_8 ↑	Recall ↑	IS↑
CUB [5]	DDPM	0.2	-	8.34	0.91	0.70	5.34
	CB-DDPM (ours)	0.4	0.1	**8.23**	**0.92**	**0.70**	**5.36**
CelebA-5 [2]	DDPM	0.6	-	10.68	0.92	0.51	2.22
	CBDM (ours)	1.0	0.05	**8.69**	**0.94**	**0.57**	**2.26**
ImageNet-LT [5]	DDPM	1.2	-	17.4	0.93	**0.33**	25.4
	CB-DDPM (ours)	1.6	0.01	**16.3**	**0.93**	0.26	**40.3**

Table 1. CBDM performance on high-resolution datasets. Here, CB-DDPM refers to the DDPM model fine-tuned by our method, ω refers to the guidance strength and τ refers to the weight of the regularization term. We note than the fine-tuning is applied on CUB and ImageNet-LT due to the limited calculation budget.

Performance on larger datasets We also investigated our methods' performance on commonly encountered datasets with higher resolution: CUB-200 [5] (of resolution 128), CelebA-5 [2] (of resolution 64) and ImageNet-LT [3] (of resolution 64).

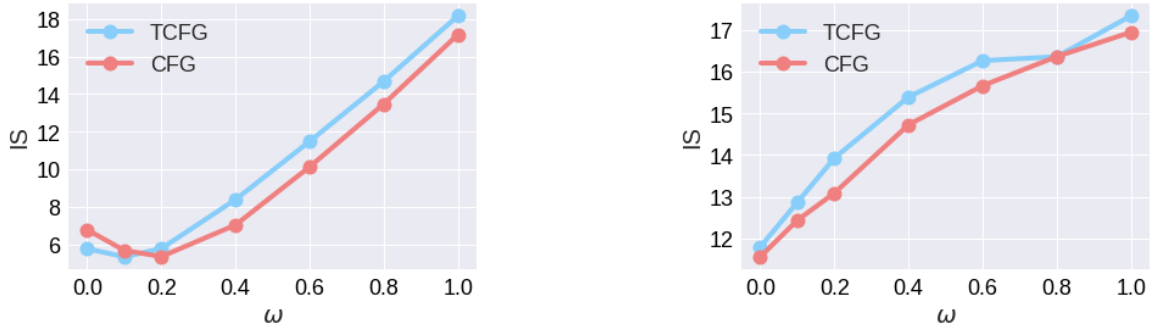
Note that, the sampling size for evaluation is based on its corresponding training set (except Imagenet-LT, which uses 50k samples for evaluation) size for correctly calculating some metrics. From the results in Table 1, we demonstrate CBDM and the fine-tuned model are consistently better than DDPM except on ImageNet-LT. ImageNet-LT has lower Recall when using CBDM, which may be attributed to the limited size of its evaluation dataset (50k images). In contrast, its FID and IS metrics are derived from the much larger training set statistics, making them more reliable. Also, similar to our observation before, the improvement on the imbalanced dataset (CelebA-5 & Imagenet-LT) is more obvious than on the balanced dataset (CUB). We noticed that the regularization weight should be chosen with more caution when training models with larger resolution in order to avoid potential mode collapse. As marked in Table 1, we use some small values such as 0.1 and 0.01 for experiments.

Trainable classifier-free guidance (TCFG) One drawback of classifier-free guidance (CFG) lies in its sampling speed. As CFG requires calling the model both in the unconditional and conditional case, CFG doubles the time complexity of common diffusion models during the sampling stage. Therefore, we tried to solve this issue by adding another conditional layer in the backbone to encode the guidance strength information. Precisely, we sample randomly the guidance strength ω and add another two loss terms to the model:

$$\mathcal{L}_g(\boldsymbol{x}_t, y) = ||[\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, y, \omega) - \omega(\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, y) - \boldsymbol{\epsilon}_\theta(\boldsymbol{x}_{t-1})).sg()] - \boldsymbol{\epsilon}||^2 \tag{28}$$

$$\mathcal{L}_{gc}(\boldsymbol{x}_t, y) = \frac{1}{4}||[\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, y, \omega).sg() - \omega(\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, y) - \boldsymbol{\epsilon}_\theta(\boldsymbol{x}_{t-1}))] - \boldsymbol{\epsilon}||^2 \tag{29}$$

Those two losses decrease significantly the sampling time at the price of a longer training time. Table 5 shows the comparison of using CFG and TCFG for sampling. We trained the DDPM as well as the TCFG model on the CIFAR100 dataset respectively with the same number of iterations for both. Given the large number of parameters to be tested for visualization, we generated 10k images for each setting and tested 6 guidance strength within 0 to 1.



(a) FID (b) IS

Figure 5. CFG/TCFG comparison with an embedded ω

The Figure 5 shows that the FID and IS scores of TCF and TCFG are quite comparable. Although TCFG shows a slightly higher FID compared to TCF, it demonstrates a slightly better IS value, which implies that TCFG prioritizes improving the image quality over diversity. However, if TCFG is shifted by 0.1 units, the curves display a high degree of similarity with CFG. This shift occurs because TCFG training utilizes all backbone parameters to train a model under difference guidance, which makes the model's parameters move closer to the guided generation results. Additionally, when the guidance strength of TCFG is set to 0.1, the FID score of TCFG (ω=0.1, FID=5.343) is slightly better than that of the CFG approach at the optimal guidance strength (ω=0.2, FID=5.357). In conclusion, the TCFG method demonstrates a high degree of similarity to the CFG method but maintained the sampling speed equivalent to the unguided generation. Moreover, the TCFG method is not difficult to implement and can be combined with any diffusion model using CFG guidance. However, the primary challenge would be the reduction in the training speed. For time limit, we only conduct TCFG experiments on CIFAR100 dataset, but we encourage testing this trick in practice when the sampling speed is an important issue.

Unconditional training probability Another hyperparameter in CBDM worth exploring is the probability of unconditional generation involved in CFG. According to the original paper, we performed unconditional generation with a probability of

10% and conditional generation with a probability of 90%. In our experiments, we increased the value of this probability and tried 20% and 50% as the new unconditional generation probabilities. As shown in the following table Table 2, we find that the performance of CBDM is optimal when ϕ equals to 0.1.

ϕ	FID	IS
10%	8.299	12.457
20%	9.161	12.261
50%	10.392	11.510

Table 2. Influence of ϕ to model performance

D. Additional Discussions

Relationship between Classifier-Free Guidance and Logit Adjustment It is not difficult to see that the structure of the adjustment form of CBDM and logit adjustment is similar. In classification tasks, logit adjustment can be implemented in the form of post-hoc adjustment or in the form of training. Precisely, by noting the classifier as f, the frequency of class y as $\pi(y)$ and the adjustment weight as τ, we have:

- Post-hoc: $f_y^\star(x) = f_y(x) + \tau\pi(y)$
- Training: $f_y^\star(x) = f_y(x) - \tau\pi(y)$

Similarly, for the diffusion model, we reverse the adjustment term as in Prop. (1) in order to realize a training adjustment, and we have:

- Post-hoc: $\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_{t-1}, y) = \boldsymbol{\epsilon}_\theta(\boldsymbol{x}_{t-1}, y) + \tau\boldsymbol{\epsilon}_{adj}$
- Training: $\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_{t-1}, y) = \boldsymbol{\epsilon}_\theta(\boldsymbol{x}_{t-1}, y) - \tau\boldsymbol{\epsilon}_{adj}$

Although the idea of this adjustment is very simple, it is convenient to transfer some cumbersome post-hoc adjustment from sampling to training.

References

[1] Prafulla Dhariwal and Alexander Quinn Nichol. Diffusion models beat GANs on image synthesis. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *NeurIPS*, 2021. 1

[2] Jaehyung Kim, Jongheon Jeong, and Jinwoo Shin. M2m: Imbalanced classification via major-to-minor translation. In *CVPR*, 2020. 5

[3] Ziwei Liu, Zhongqi Miao, Xiaohang Zhan, Jiayun Wang, Boqing Gong, and Stella X. Yu. Large-scale long-tailed recognition in an open world. In *CVPR*, 2019. 5

[4] Aditya Krishna Menon, Sadeep Jayasumana, Ankit Singh Rawat, Himanshu Jain, Andreas Veit, and Sanjiv Kumar. Long-tail learning via logit adjustment. In *ICLR*. OpenReview.net, 2021. 2

[5] Peter Welinder, Steve Branson, Takeshi Mita, Catherine Wah, Florian Schroff, Serge Belongie, and Pietro Perona. Caltech-ucsd birds 200. Technical Report CNS-TR-201, Caltech, 2010. 5