# Adaptive Non-uniform Timestep Sampling for Accelerating Diffusion Model Training

Supplementary Material

## 9. Time Complexity of Our Algorithm

We compute the additional computational overhead introduced by the proposed timestep sampling mechanism. During each iteration of diffusion model training, particularly for sufficiently large models, the majority of the time is dedicated to the forward and backward passes of the objective $\mathcal{L}_t(\theta)$. The backward pass, in particular, is more computationally demanding due to the communication step required for gradient synchronization, typically taking about three times longer than the forward pass [21]. Thus, if we denote the time required for the forward pass on a batch of data $B$ as $t_{\text{fwd}(\theta)}$, a single training iteration will approximately take:

$$t_{\text{iter}(\theta)} = t_{\text{fwd}(\theta)} + t_{\text{bwd}(\theta)} \approx 4 \cdot t_{\text{fwd}(\theta)}.$$

If we do not approximate $\Delta_k^t$, its computation requires evaluating $\sum_{t=1}^{T} \mathcal{L}_t(\cdot)$ twice—once before and once after updating $\theta$—resulting in a time complexity of $t_\Delta = 2T \cdot t_{\text{fwd}(\theta)}$. To alleviate this computational burden, we have proposed an approximation using a subset of timesteps $S$, which reduces $T$ to $|S|$. However, this approach also incurs additional time for sampling objectives to identify an optimal subset $S$. The time complexity of this approximation scheme is given by:

$$t_{\tilde{\Delta}} = 2|S| \cdot t_{\text{fwd}(\theta)} + t_S, \quad \text{where} \quad t_S = 2\frac{T}{|B|} \cdot t_{\text{fwd}(\theta)}.$$

The reason behind the time $t_S$ spent on subset selection is that we use a single $x_0$ to compute objective samples for this selection, while $t_{\text{fwd}}$ involves a full batch of $|B|$ $x_0$s during the forward pass. The time spent for running the feature selection algorithm is negligible (takes about $1\%$ of $t_{\text{fwd}}$) and is therefore ignored. With typical hyperparameter choices in practice, such as $|S| = 3$, $|B| = 128$, $T = 1000$, this results in $t_{\tilde{\Delta}} \approx 21 \cdot t_{\text{fwd}(\theta)}$ for running Algorithm 2.

Since we run Algorithm 2 only once every $f_S = 40$ updates of $\theta$, the total time for the proposed algorithm is given by:

$$\begin{aligned} t_{\text{iter}(\theta,\phi)} &= t_{\text{iter}(\theta)} + t_{\text{fwd}(\phi)} + \frac{1}{f_S}\left(t_{\tilde{\Delta}} + t_{\text{iter}(\phi)}\right) \\ &\approx 5.63 \cdot t_{\text{fwd}(\theta)} \approx 1.41 \cdot t_{\text{iter}(\theta)}, \end{aligned}$$

where we have assumed $t_{\text{fwd}(\theta)} \approx t_{\text{fwd}(\phi)}$. In practice, there are additional overheads due to minor factors, and we observed that our algorithm takes approximately 1.5 times longer in terms of wall-clock time compared to the baseline.

## 10. Implementation Details

We mainly used a machine equipped with four NVIDIA RTX 3090 GPUs to train the models.

| Dataset | CIFAR-10 32×32 | CelebA-HQ 256×256 | ImageNet 256×256 |
|---|---|---|---|
| Diffusion architecture | DDPM [14] | LDM [24] | ADM [9] |

Table 5. Our implementation details based on CIFAR-10, CelebA-HQ, ImageNet datasets.

| Dataset | CIFAR-10 32×32 | CelebA-HQ 256×256 | ImageNet 256×256 |
|---|---|---|---|
| sampling steps | 1000 | 200 | 50 |
| sampling algorithm | DDPM sampler | DDIM sampler [28] | EDM sampler [17] |
| number of samples in evaluation | 50K | 50K | 50K |

Table 6. Our evaluation settings based on CIFAR-10, CelebA-HQ, ImageNet datasets.

**Baseline** For all baselines, we used the most popular settings. For Min-SNR [11], we used snr gamma=5. For P2 [7], we used gamma=0 and k=1. For log normal [17], we sampled weights from a normal distribution with a mean of 0 and a standard deviation of 1, followed by applying a sigmoid function. For SpeeD [31], we sampled timesteps according to the official code and used gamma=1 and k=1 for weighting.

**Feature Selection method** To overcome the computational burden of calculating $\delta_{k,i}^t$ across all timesteps, we employ a feature selection method. Specifically, we treat $\delta_{k,i}^t$ as features to predict the target $\Delta_k^t$ using a linear regression model. The F-statistic is then computed for each feature $\delta_{k,i}^t$ and by summing the top $M$ features with the highest F-statistics, we can efficiently approximate $\Delta_k^t$ while focusing on the most influential timesteps $i$. The F-statistic identifies the features that have the strongest linear relationship with the target variable, enabling us to focus on the most critical timesteps in the approximation process. Although the F-statistic is employed here, other feature selection methods could also be applied.

## 11. Hyperparameter settings

Table 7. Hyperparameter settings for CIFAR-10

| Category | Parameter | CIFAR-10 |
|---|---|---|
| **Diffusion** | Timesteps | 1000 |
| | Beta Start | 0.0001 |
| | Beta End | 0.02 |
| | Beta Schedule | Linear |
| | Model Mean Type | Eps |
| | Model Variance Type | Fixed-large |
| | Loss Type | MSE |
| | Backbone | UNet |
| | In Channels | 3 |
| | Hidden Channels | 128 |
| | Channel Multipliers | [1, 2, 2, 2] |
| | Number of Residual Blocks | 2 |
| | Drop Rate | 0.1 |
| | Learning Rate | 2e-4 |
| | Batch Size | 128 |
| | Gradient Norm | 1.0 |
| | Epochs | 2040 |
| | Warmup | 5000 |
| | Use EMA | True |
| | EMA Decay | 0.9999 |
| **Timestep Sampler** | Learning Rate | 1e-2(linear, quad), 1e-3(cosine) |
| | Entropy Coefficient | 1e-2 |
| | In Channels | 3 |
| | Hidden Channels | 128 |
| | Hidden Depth | 2 |
| | $f_s$ | 40 |
| | $\|Q\|$ | 20 |
| | $\|S\|$ | 3 |

Table 8. Hyperparameter settings for CelebA-HQ 256x256

| Category | Parameter | CelebA-HQ |
|---|---|---|
| **Diffusion** | Timesteps | 1000 |
| | Beta Schedule | Linear |
| | Model Mean Type | Eps |
| | Loss Type | MSE |
| | Backbone | UNet |
| | In Channels | 3 |
| | Hidden Channels | 224 |
| | Channel Multipliers | [1, 2, 3, 4] |
| | Number of Residual Blocks | 2 |
| | Drop Rate | 0.0 |
| | Optimizer | AdamW ($\beta_1 = 0.9, \beta_2 = 0.999$) |
| | Learning Rate | 1.92e-4 |
| | Batch Size | 24 |
| | Epochs | 300 |
| | Use EMA | True |
| | EMA Decay | 0.9999 |
| **Timestep Sampler** | Learning Rate | 1e-3 |
| | Entropy Coefficient | 1e-2 |
| | In Channels | 3 |
| | Hidden Channels | 256 |
| | Hidden Depth | 2 |
| | $f_s$ | 40 |
| | $\|Q\|$ | 20 |
| | $\|S\|$ | 3 |

Table 9. Hyperparameter settings for ImageNet 256x256

| Category | Parameter | ImageNet |
|---|---|---|
| **Diffusion** | Timesteps | 1000 |
| | Beta Schedule | Cosine |
| | Model Mean Type | Epsilon |
| | Loss Type | MSE |
| | Backbone | U-ViT |
| | Layers | 238 |
| | Hidden Size | 1152 |
| | Heads | 16 |
| | Depths | 12 |
| | Optimizer | AdamW ($\beta_1 = 0.99, \beta_2 = 0.99$) |
| | Learning Rate | 2e-4 |
| | Batch Size | 256 |
| | Training iterations | 2.1M |
| | Use EMA | True |
| | EMA Decay | 0.9999 |
| **Timestep Sampler** | Learning Rate | 1e-3 |
| | Entropy Coefficient | 1e-2 |
| | In Channels | 3 |
| | Hidden Channels | 256 |
| | Hidden Depth | 2 |
| | $f_s$ | 40 |
| | $\|Q\|$ | 20 |
| | $\|S\|$ | 3 |

## 12. Visualization

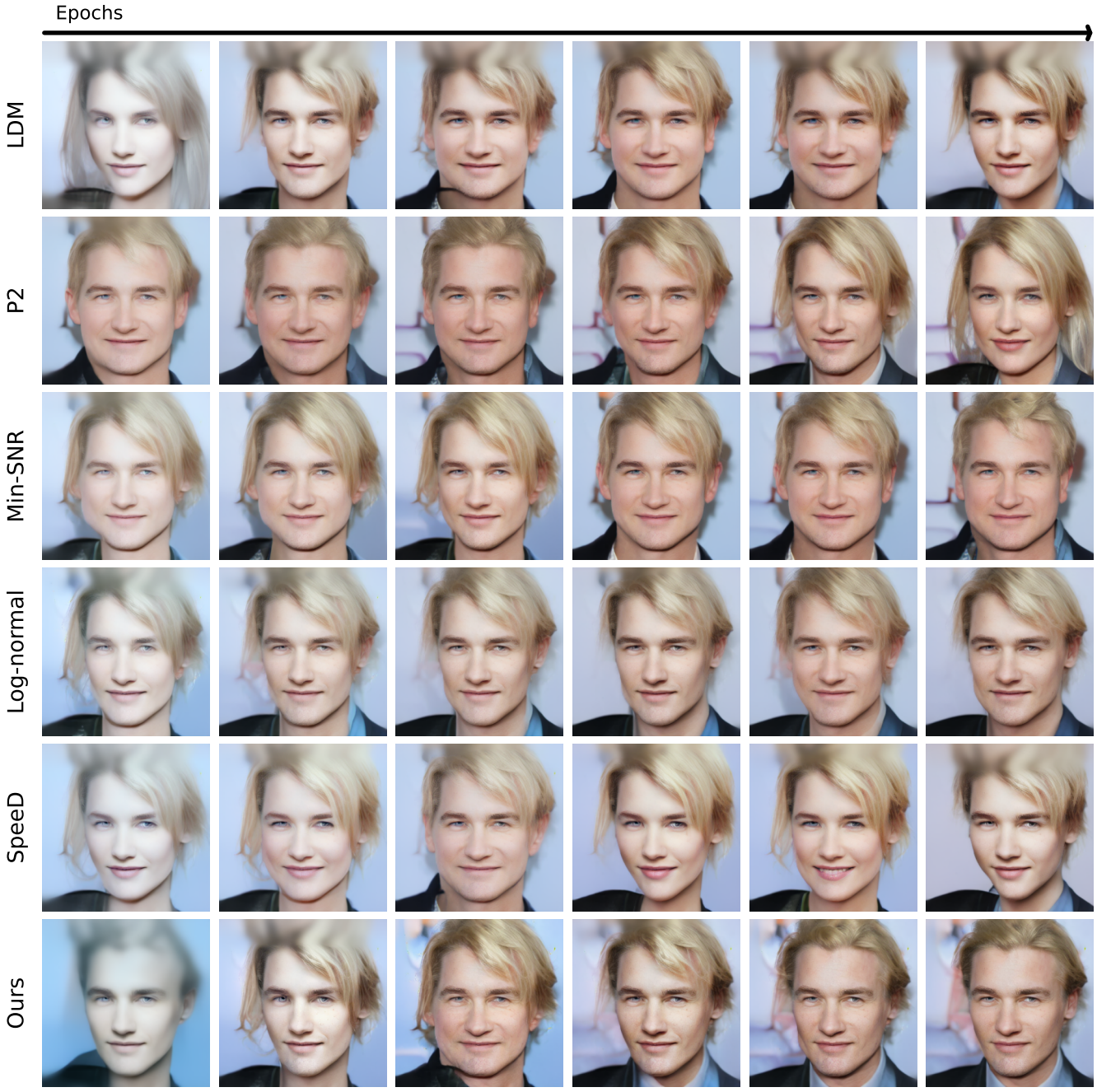


Figure 8. Visualization of CelebA-HQ images generated by our method and baseline methods.