# A. Weighting Schemes

## A.1. Additional Visualizations

In the main text, we showed weights of both our new weighting scheme and the baseline, as functions of signal-to-noise ratio (SNR). In Fig. A (left), we show weights as functions of time steps ($t$). To exhibit relative changes of weights, we show normalized weights as functions of both time steps (Fig. A (middle)) and SNR (Fig. A (right)). We normalized so that the sum of weights for all time steps become 1. Normalized weights suggest that larger $\gamma$ suppresses weights at steps near $t = 0$ and uplifts weights at larger steps. Note that weights of VLB objective are equal to a constant, as such objective does not impose any inductive bias for training. In contrast, as discussed in the main text, our method encourages the model to learn rich content rather than imperceptible details.

## A.2. Derivations

In the main text, we wrote the baseline weighting scheme $\lambda_t$ as a funtion of SNR, which characterizes the noise level at each step $t$. Below is the derivation:

$$
\begin{aligned}
\lambda_t &= (1-\beta_t)(1-\alpha_t)/\beta_t \\
&= (\alpha_t/\alpha_{t-1})(1-\alpha_t)/(1-\alpha_t/\alpha_{t-1}) \\
&= \alpha_t(1-\alpha_t)/(\alpha_{t-1}-\alpha_t) \\
&= \alpha_t(1-\alpha_t)/((1-\alpha_t)-(1-\alpha_{t-1})) \\
&= \frac{\text{SNR}(t)}{(1+\text{SNR}(t))^2}/(\frac{1}{1+\text{SNR}(t)} - \frac{1}{1+\text{SNR}(t-1)}) \\
&= \frac{\text{SNR}(t)}{(1+\text{SNR}(t))^2}/\frac{\text{SNR}(t-1)-\text{SNR}(t)}{(1+\text{SNR}(t))(1+\text{SNR}(t-1))} \\
&= \frac{\text{SNR}(t)(1+\text{SNR}(t-1))}{(1+\text{SNR}(t))(\text{SNR}(t-1)-\text{SNR}(t))} \\
&\approx \frac{-\text{SNR}(t)}{\text{SNR}'(t)}\ (T \to \infty), \qquad \text{(A)}
\end{aligned}
$$

which is a differential of log-SNR($t$) regarding time-step $t$.

# B. Discussions

## B.1. Limitations

Despite the promising performances achieved by our method, diffusion models still need multiple sampling steps. Diffusion models require at least 25 feed-forwards with DDIM sampler, which makes it difficult to use diffusion models in real-time applications. Yet, they are faster than autoregressive models which generate a pixel at each step. In addition, we have observed in section 4.3 that our method enables better FID with half the number of steps required by the baseline. Along with our method, optimizing sampling schedules with dynamic programming [9] or distilling DDIM sampling into a single step model [5] might be promising future directions for faster sampling.

## B.2. Broader Impacts

The proposed method in this work allows high-fidelity image generation with diffusion-based generative models. Improving the performance of generative models can enable multiple creative applications [2, 6]. However, such improvements have the potential to be exploited for deception. Works in deepfake detection [8] or watermarking [11] can alleviate the problems. Investigating invisible frequency artifacts [8] in samples of diffusion models might be promising approach to detect fake images.

# C. Implementation Details

For a given time-step $t$, the input noisy image $x_t$ and output noise prediction $\epsilon$ and variance $\sigma_t$ are images of the same resolution. Therefore, $\epsilon_\theta$ is parameterized with the U-Net [7]-style architecture of three input and six output channel dimensions. We inherit the architecture of ADM [3], which is a U-Net with large channel dimension, BigGAN [1] residual blocks, multi-resolution attention, and multi-head attention with fixed channels per head. Time-step $t$ is provided to the model by adaptive group normalization (AdaGN), which transforms $t$ embeddings to scales and biases of group normalizations [10]. However, for efficiency, we use fewer base channels, fewer residual blocks, and a self-attention at a single resolution (16×16).

Hyperparameters for training models are in Tab. A. We use $\gamma = 0.5$ for FFHQ and CelebA-HQ as it achieve slightly better FIDs than $\gamma = 1.0$ on those datasets. Models consist of one or two residual blocks per resolution and self-attention blocks at 16×16 resolution or at bottleneck layers of 8×8 resolution. Our default model has only 94M parameters, while recent works rely on large models (larger than 500M) [3]. While recent works use 2 or 4 blocks per resolution, we use only one block, which leads to speed-up of training and inference. We use dropout when training on limited data. We trained models using EMA rate of 0.9999, 32-bit precision, and AdamW optimizer [4].

# D. Additional Results

**Qualitative.** Additional samples for all datasets mentioned in the paper are in Fig. D.

**Quantitative.** In Fig. 1 of the main text, we measured perceptual distances to investigate how the diffusion process corrupts perceptual contents. In Fig. 2, we qualitatively explored what a trained model learned at each step (Fig. 2). Here, we reproduce Fig. 1 at various datasets and resolutions in Fig. B and show the quantitative result of Fig. 2 in Fig. C. These results indicate that our investigation in Sec. 3.1 holds for various datasets and resolutions.

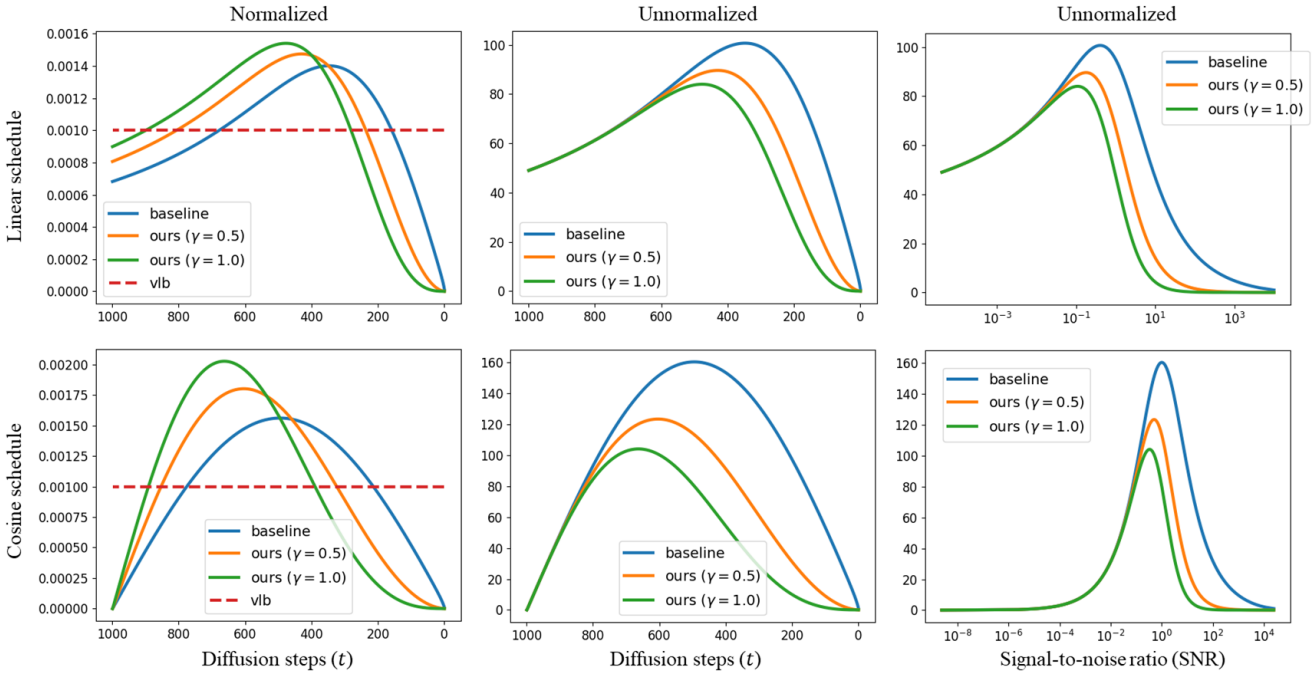


Figure A. Unnormalized or normalized weights as functions of diffusion steps or signal-to-noise ratio (SNR). Large $t$ and small SNR indicates noisy image $x_t$ near random noise $x_T$, whereas small $t$ and large SNR indicates $x_t$ near a clean image $x_0$.

| | FFHQ, CelebA-HQ | AFHQ-D | CUB, Flowers, MetFaces | Tab. 3 (b) | Tab. 3 (c) | Tab. 3 (d) |
|---|---|---|---|---|---|---|
| $T$ | 1000 | 1000 | 1000 | 1000 | 1000 | 1000 |
| $\beta_t$ | linear | linear | linear | linear | linear | linear |
| Model Size | 94 | 94 | 94 | 81 | 90 | 132 |
| Channels | 128 | 128 | 128 | 128 | 128 | 128 |
| Blocks | 1 | 1 | 1 | 1 | 1 | 2 |
| Self-attn | 16, bottle | 16, bottle | 16, bottle | 16, bottle | bottle | 16, bottle |
| Heads Channels | 64 | 64 | 64 | 64 | 64 | 64 |
| BigGAN Block | yes | yes | yes | no | yes | yes |
| $\gamma$ | 0.5 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 |
| Dropout | 0.0 | 0.1 | 0.1 | 0.1 | 0.1 | 0.1 |
| Learning Rate | $2e^{-5}$ | $2e^{-5}$ | $2e^{-5}$ | $2e^{-5}$ | $2e^{-5}$ | $2e^{-5}$ |
| Images (M) | 18, 4.4 | 2.4 | 4.8, 4.4, 1.6 | 0.8 | 0.8 | 0.8 |

Table A. Hyperparameters.

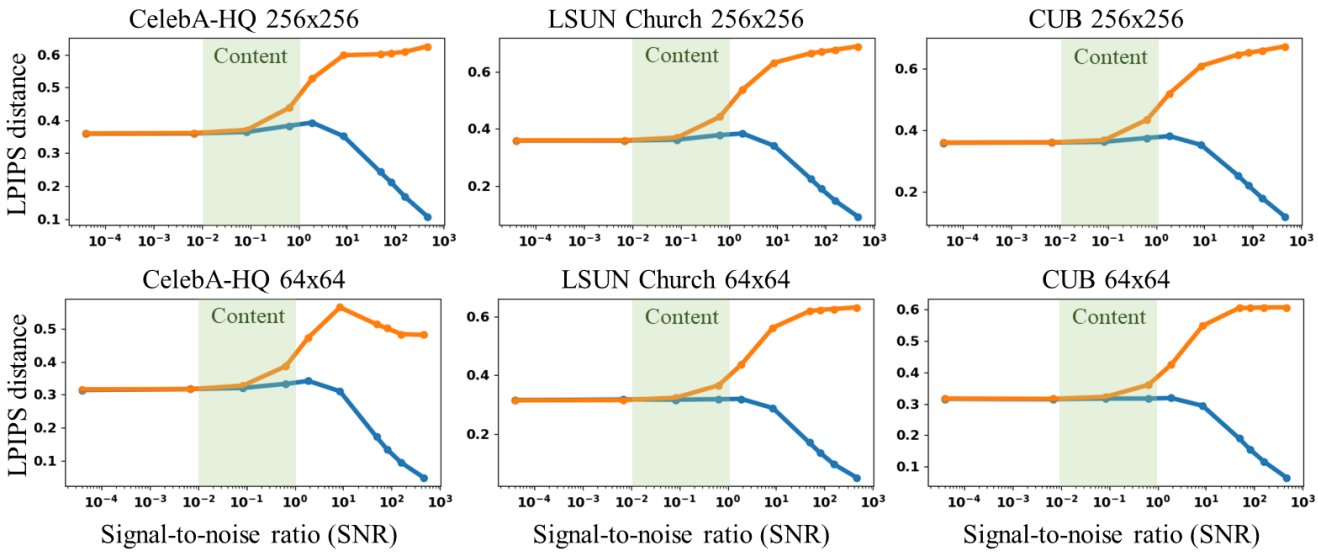


Figure B. **Generalization of sec. 3.** Results with CelebA-HQ, LSUN-Church, and CUB at $256^2$ and $64^2$ resolutions.

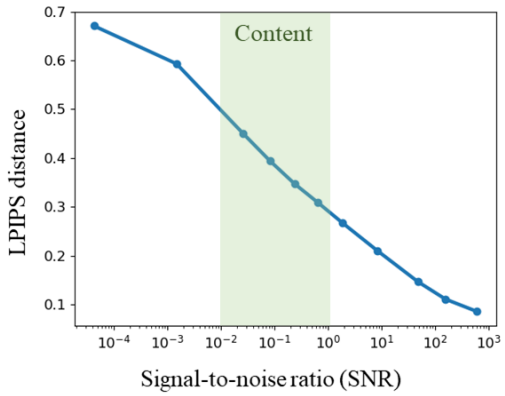


Figure C. **Stochastic reconstruction.** Perceptual distance between input and reconstructed image as a function of signal-to-noise ratio, measured with random 200 images from FFHQ.

## References


[1] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018. 1

[2] Jooyoung Choi, Sungwon Kim, Yonghyun Jeong, Youngjune Gwon, and Sungroh Yoon. Ilvr: Conditioning method for denoising diffusion probabilistic models. *arXiv preprint arXiv:2108.02938*, 2021. 1

[3] Prafulla Dhariwal and Alex Nichol. Diffusion models beat gans on image synthesis. *arXiv preprint arXiv:2105.05233*, 2021. 1

[4] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 1

[5] Eric Luhman and Troy Luhman. Knowledge distillation in iterative generative models for improved sampling speed. *arXiv preprint arXiv:2101.02388*, 2021. 1

[6] Chenlin Meng, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. Sdedit: Image synthesis and editing with stochastic differential equations. *arXiv preprint arXiv:2108.01073*, 2021. 1

[7] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, 2015. 1

[8] Sheng-Yu Wang, Oliver Wang, Richard Zhang, Andrew Owens, and Alexei A Efros. Cnn-generated images are surprisingly easy to spot... for now. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8695–8704, 2020. 1

[9] Daniel Watson, Jonathan Ho, Mohammad Norouzi, and William Chan. Learning to efficiently sample from diffusion probabilistic models. *arXiv preprint arXiv:2106.03802*, 2021. 1

[10] Yuxin Wu and Kaiming He. Group normalization. In *Proceedings of the European conference on computer vision (ECCV)*, pages 3–19, 2018. 1

[11] Ning Yu, Vladislav Skripniuk, Sahar Abdelnabi, and Mario Fritz. Artificial fingerprinting for generative models: Rooting deepfake attribution in training data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14448–14457, 2021. 1

Figure D. Additional samples generated with our models traind on various datasets.