Wavelet Diffusion Models are fast and scalable Image Generators — Supplementary Material —

Hao Phung[†], Quan Dao[†], Anh Tran
VinAI Research
{v.haopt12, v.quandm7, v.anhtt152}@vinai.io

Abstract

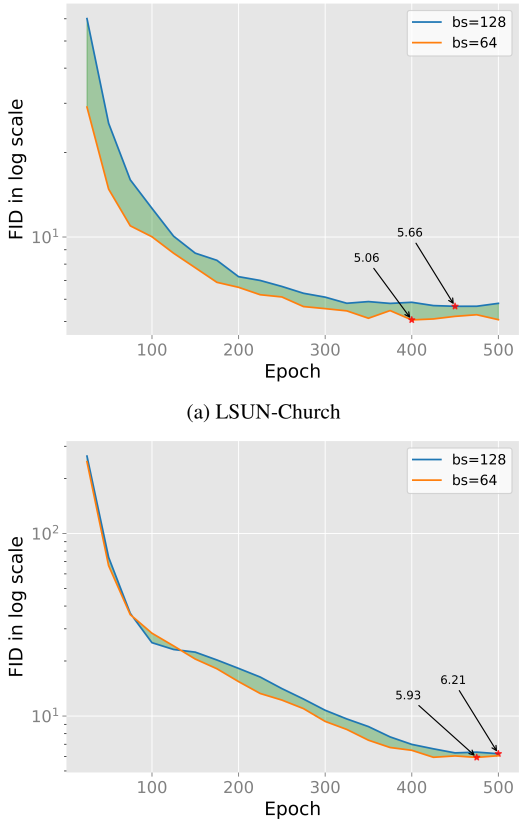
In this supplementary PDF, we first analyze the sensitivity of batch size to the model performance, then give details of training experiments, including network configurations and choices of hyper-parameters, and finally provide additional qualitative samples on different datasets, namely CIFAR-10, STL-10, CelebA-HQ (256, 512 & 1024), and LSUN-Church.

1. Sensitivity analysis of training batch size

We recognize that training batch size is a critical aspect affecting the final performance. Large batch size often results in worse performance, with the compensation being training time. Here, we carefully analyze the effect of training batch size on the model performance measured by Frechet inception distance (FID) [1] (depicted in Fig. 1).

As expected, the model trained with batch size 64 consistently performs better than the model trained with batch size 128, as illustrated via the training curves on LSUN-Church in Fig. 1a. The gap is initially large during the first 100 epochs and then shrunk in the following epochs. However, the performance gap is still significant. The best FID of the model with batch size 64 is 5.06, which is 0.6 points lower than the best FID of the model trained on batch size 128. More importantly, our model outperforms DDGAN (5.06 vs. 5.25) when using the same batch size of 64.

We further verify the effect of batch size to our trained models on CelebA-HQ (256) with batch sizes 128 and 64. As shown in Fig. 1b, the model trained with batch size 64 achieves a minimum FID of 5.93, which is 0.28 points lower than the FID 6.21 of the model trained with batch size 128. It again confirms that batch size is an important factor to be considered when evaluating and comparing model performance.



(a) LSUN-Church

(b) CelebA-HQ

Figure 1. Training curves on LSUN-Church and CelebA-HQ (256 × 256) with two different batch sizes, namely 64 and 128. The area under curves (green color) presents the difference in FID scores between the two trained models. Unlike CelebA-HQ (256), the gap between the two models on LSUN-Church is notably large across different epochs. By using an appropriate batch size of 64, the minimum FID of the trained model can reduce to 5.06. On CelebA-HQ (256), it again confirms that using sufficient batch size is highly necessary for the model performance as the model is trained with a batch size of 64 attaining a better FID score of 5.93.

Note that we used a larger batch size than DDGAN (32

[†] Equal contribution.

	CIFAR-10	STL-10	CelebA-HQ (256)	CelebA-HQ (512)	LSUN-Church
# of ResNet blocks per scale	2	2	2	2	2
Base channels	128	128	64	64	64
Channel multiplier per scale	(1,2,2)	(1,2,2,2)	(1,2,2,2,4)	(1,1,2,2,4,4)	(1,2,2,2,4)
Attention resolutions	None	16	16	16	16
Latent Dimension	100	100	100	100	100
# of latent mapping layers	4	4	4	4	4
Latent embedding dimension	256	256	256	256	256

Table 1. Network configurations.

vs. 16) on CelebA-HQ 512×512 with 8 GPUs[†]. Due to time and resource limits, we cannot retrain our model with batch size 16, but we expect our result will be further improved in this fair experiment configuration, further increasing the gap in performance between our algorithm and DDGAN.

2. Experimental details

2.1. Wavelet transformations

We utilize the implementation of wavelet transformations, including Discrete wavelet transform (DWT) and Discrete inverse wavelet transform (IWT), from [2]. We perform these transformations on both input images and feature maps for further processing in the proposed Wavelet-based Diffusion framework.

2.2. Network configurations

Generator. Our generator has a UNet alike architecture [3] which is mainly based on NCSN++ [4, 5]. As can be seen in Tab. 1, we show the detailed configurations of the generator for each corresponding dataset. We adjust the number of layers in the generator according to the input resolution of wavelet coefficients. The number of channels of time embedding is $4\times$ larger than the base channels.

Discriminator. The number of layers in the discriminator is the same as the one of the generator. For more details of the discriminator structure, please refer to [5].

2.3. Training hyper-parameters

For reproductivity, we further provide a full table of tuned hyper-parameters in Tab. 2. Basically, our hyper-parameters are the same as the baseline [5] except for the number of epochs and the allocated GPUs on specific datasets. Meanwhile, there are two new datasets, including STL-10 64×64 and CelebA-HQ 512×512, that share similar configurations of CIFAR-10 and CelebA-HQ 256×256, respectively. Besides, the setting of CelebA-HQ 1024×1024 is almost similar to CelebA-HQ 512×512.

For training time, CIFAR10 and STL10 models require 1.6 and 3.6 days on a single GPU, respectively. On CelebA-HQ 256 and LSUN-Church, they take 1.1 and 6.8 days on 2 and 4 GPUs, respectively. On high-resolution CelebA-HQ 512, it takes 4.3 days on 8 GPUs. Besides, the training time is mainly influenced by the number of denoising steps, the size of network architectures, and image resolutions (presented in Tab. 1) apart from the number of training epochs. This is also the same for the inference time.

3. More qualitative results

We further provide additional qualitative results on CIFAR-10 in Fig. 3, STL-10 in Fig. 4a, CelebA-HQ 256 in Fig. 5, CelebA-HQ 512 in Fig. 6, LSUN-Church in Fig. 7, and CelebA-HQ 1024 in Fig. 8.

A comparison of qualitative samples between ours and DDGAN [5] on STL-10 is also presented in Fig. 4. Our model clearly achieves better sample quality with a plausible appearance of generated objects, while the counterpart fails to represent object-specific shapes in output samples. We also add a qualitative comparison on the CelebA-HQ 512 dataset (Fig. 2), which further illustrates the advantages of our proposal in producing clearer details, such as eyebrows and wrinkles.

4. More discussion

Connection between wavelet transformation and speed. Let $X \in \mathbb{R}^{C\times H\times W}$ denote an input image. Wavelet transformation reduces its spatial dimension while increasing its channel dimension by four-folds: $Y \in \mathbb{R}^{4C\times H/2\times W/2}$. This input is then ***projected to the base channel D*** via the first linear layer, keeping the network width unchanged compared with DDGAN. Hence, most of the network benefits from $4\times$ reduction in spatial dimensions, significantly reducing its computation (measured in FLOPs). For further acceleration, we decrease the network depth since a smaller spatial dimension requires fewer downsampling steps.

Increment novelty. While wavelet transformation has been used in many previous works on different tasks, ours is the

[†] NVIDIA A100-40GB GPUs are used. Our model can fit a training batch size of 4 instead of 2 as DDGAN per GPU.

	CIFAR-10	STL-10	CelebA-HQ (256 & 512)	LSUN-Church	CelebA-HQ (1024)
lr_G	1.6e-4	1.6e-4	2.e-4	1.6e-4	2.e-4
lr_D	1.25e-4	1.25e-4	1.e-4	1.e-4	1.e-4
Adam optimizer (β_1 & β_2)	0.5, 0.9	0.5, 0.9	0.5, 0.9	0.5, 0.9	0.5, 0.9
EMA	0.9999	0.9999	0.999	0.999	0.999
Batch size	256	256	64 & 32	128	24
Lazy regularization	15	15	10	10	10
# of epochs	1800	900	500 & 400	500	400
# of timesteps	4	4	2	4	2
# of GPUs	1	1	2 & 8	4	8

Table 2. Choices of hyper-parameters

first work employing it in diffusion models and in a comprehensive manner. Wavelet transformation is carefully incorporated on both the pixel and feature levels. Particularly, for the feature level, we proposed three wavelet-based network components, and each component is designed to utilize low and high-frequency subbands for improved output quality. Thanks to these proposals, we achieve state-of-the-art running speed with a near real-time performance for a diffusion model, allowing this advanced technique to be applicable to real-time applications. Our method also provides a faster and more stable model training, as discussed in Sec 5.5. of the main paper. Hence, we believe our paper is essential and not just an incremental work.

Reasons why the high-frequency subbands are unprocessed and directly transmitted to the decoder in the frequency bottleneck block. When designing this block, we aimed to strengthen our model's focus on learning low-level features while preserving the details; hence we directly transmitted the high-frequency components to the IWT module. We have tested processing both low and high subbands on STL-10 and got almost the same FID (12.96 vs. 12.93), suggesting that this design is not critical.

Progressive upsampling. We actually tried this direction first, but it produced quite poor results. On CelebA-HQ 256, the FID score of the 2-level upsampling network is 13.11, much higher than ours (5.94). We suspect a discrepancy in conditional distribution between low and high-frequency $p(x_{hi}|x_{lo})$, causing mismatching between generated subbands, and deteriorating the output quality.

References

[1] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 2017. 1

[2] Qiufu Li, Linlin Shen, Sheng Guo, and Zhihui Lai. Wavelet integrated cnns for noise-robust image classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7245–7254, 2020. 2

[3] 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*. Springer, 2015. 2

[4] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021. 2

[5] Zhisheng Xiao, Karsten Kreis, and Arash Vahdat. Tackling the generative learning trilemma with denoising diffusion gans. In *International Conference on Learning Representations*, 2022. 2, 5

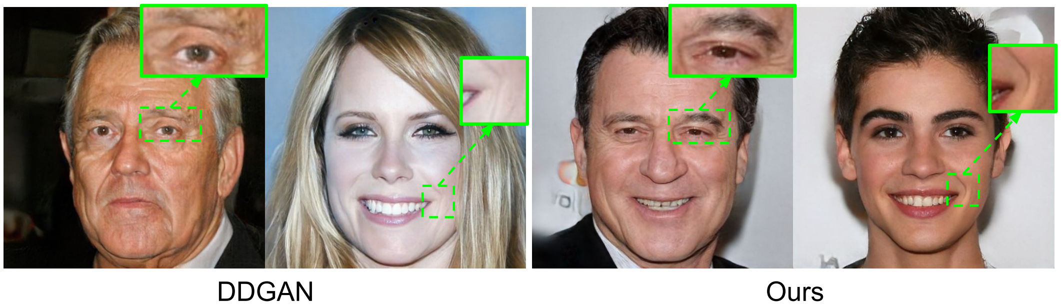


Figure 2. Qualitative comparision on CelebA-HQ 512×512

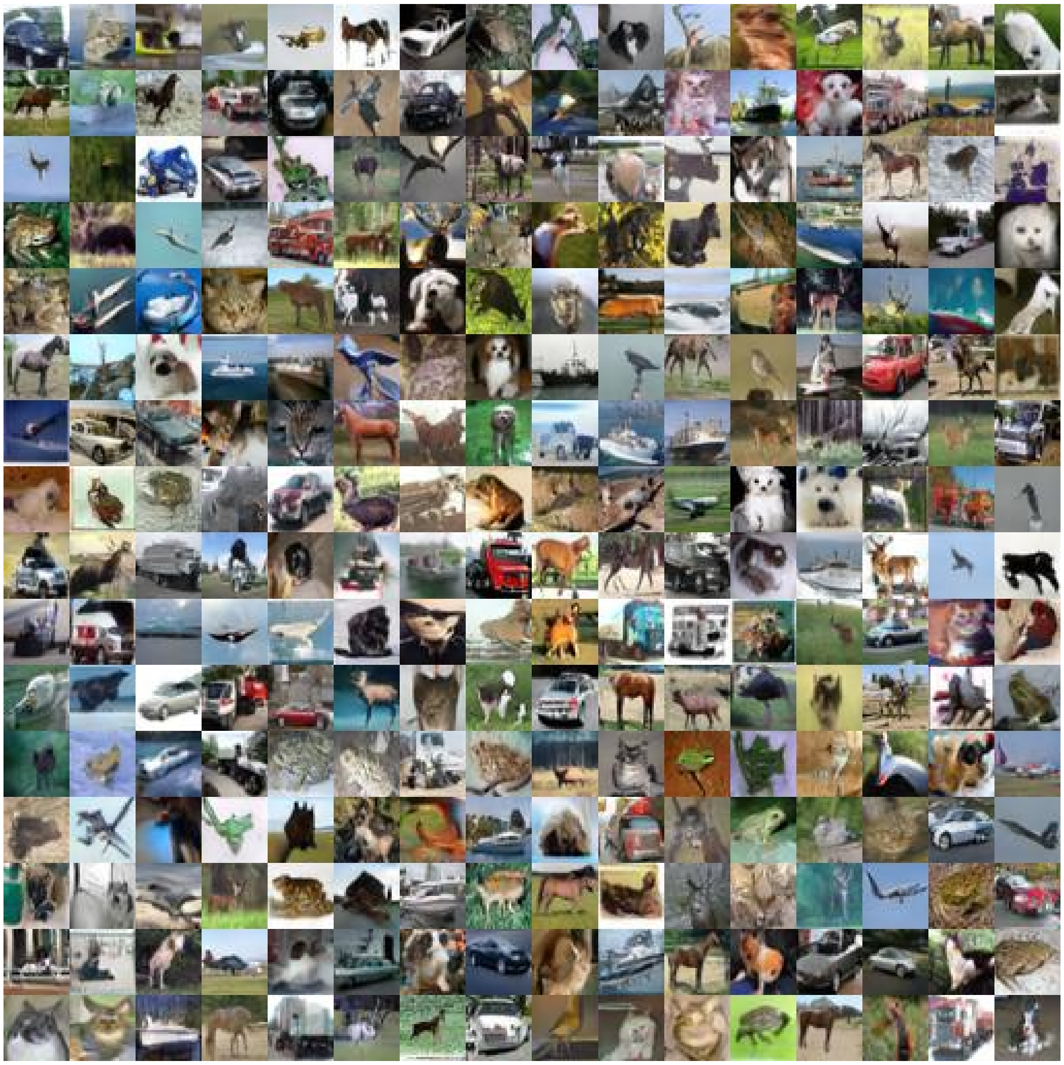
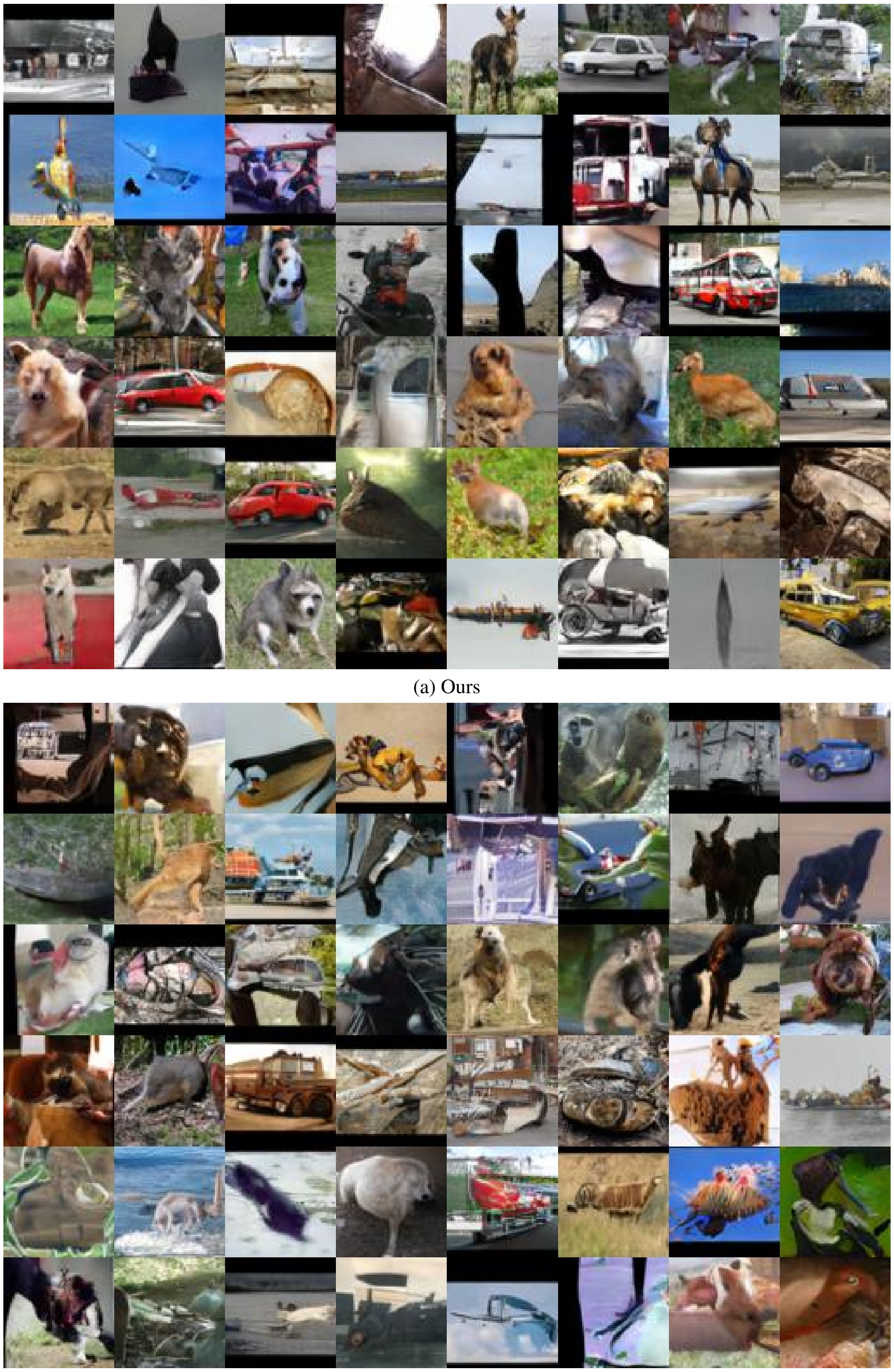


Figure 3. Non-curated generated samples on CIFAR-10



(a) Ours

(b) DDGAN

Figure 4. Non-curated generated samples between ours and DDGAN [5] on STL-10

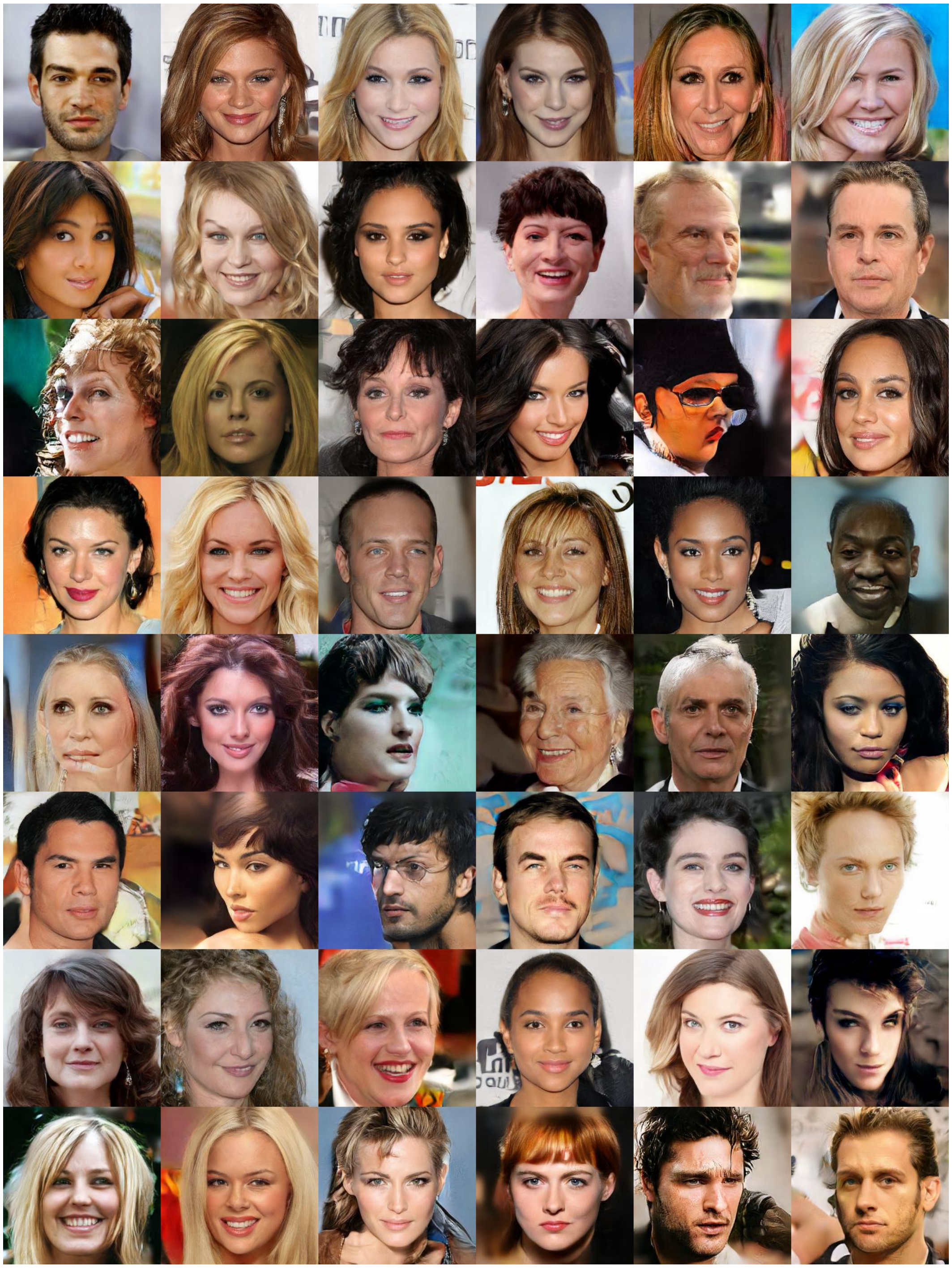


Figure 5. Non-curated generated samples on CelebA-HQ 256 × 256

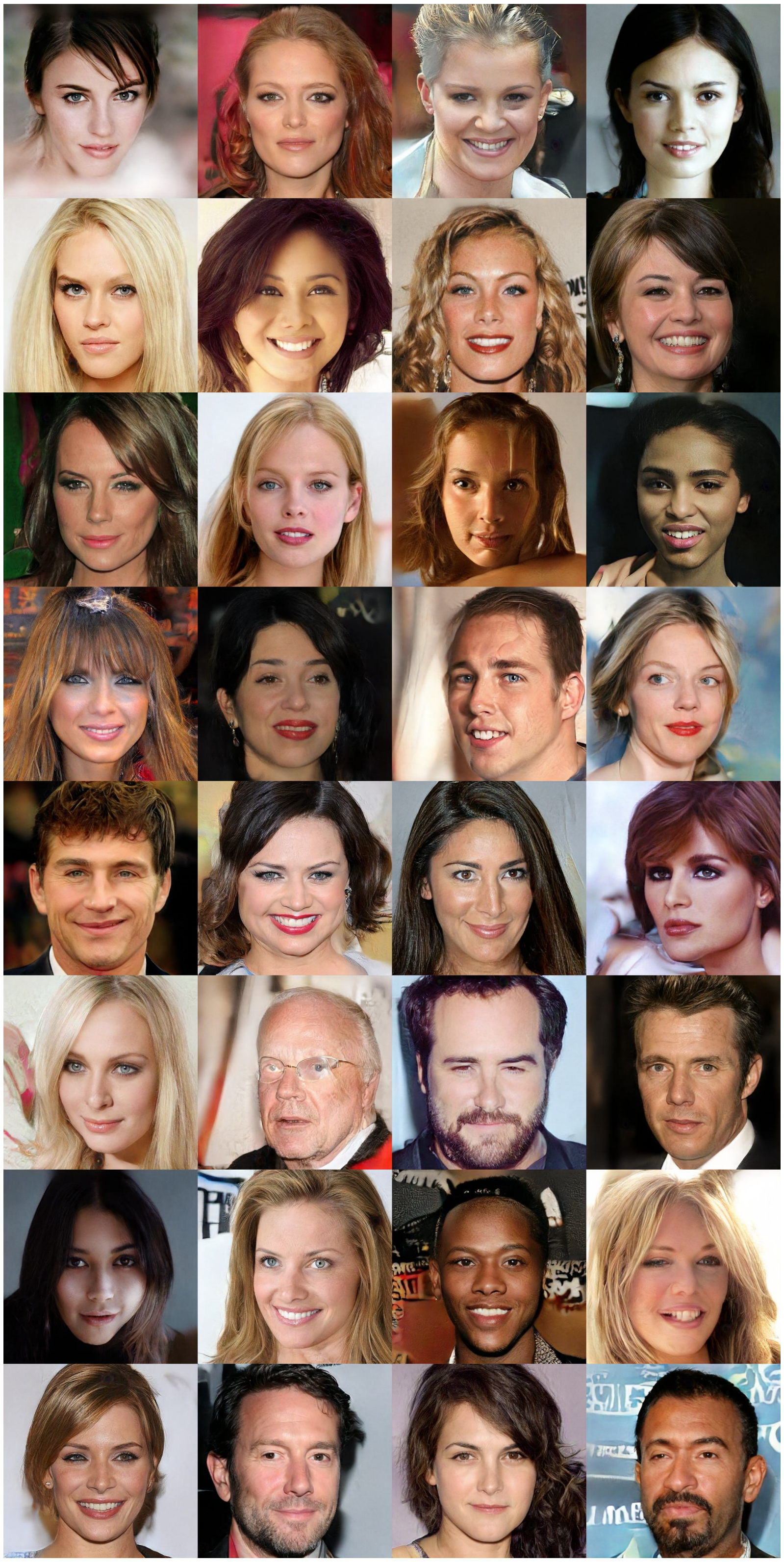


Figure 6. Non-curated generated samples on CelebA-HQ 512×512

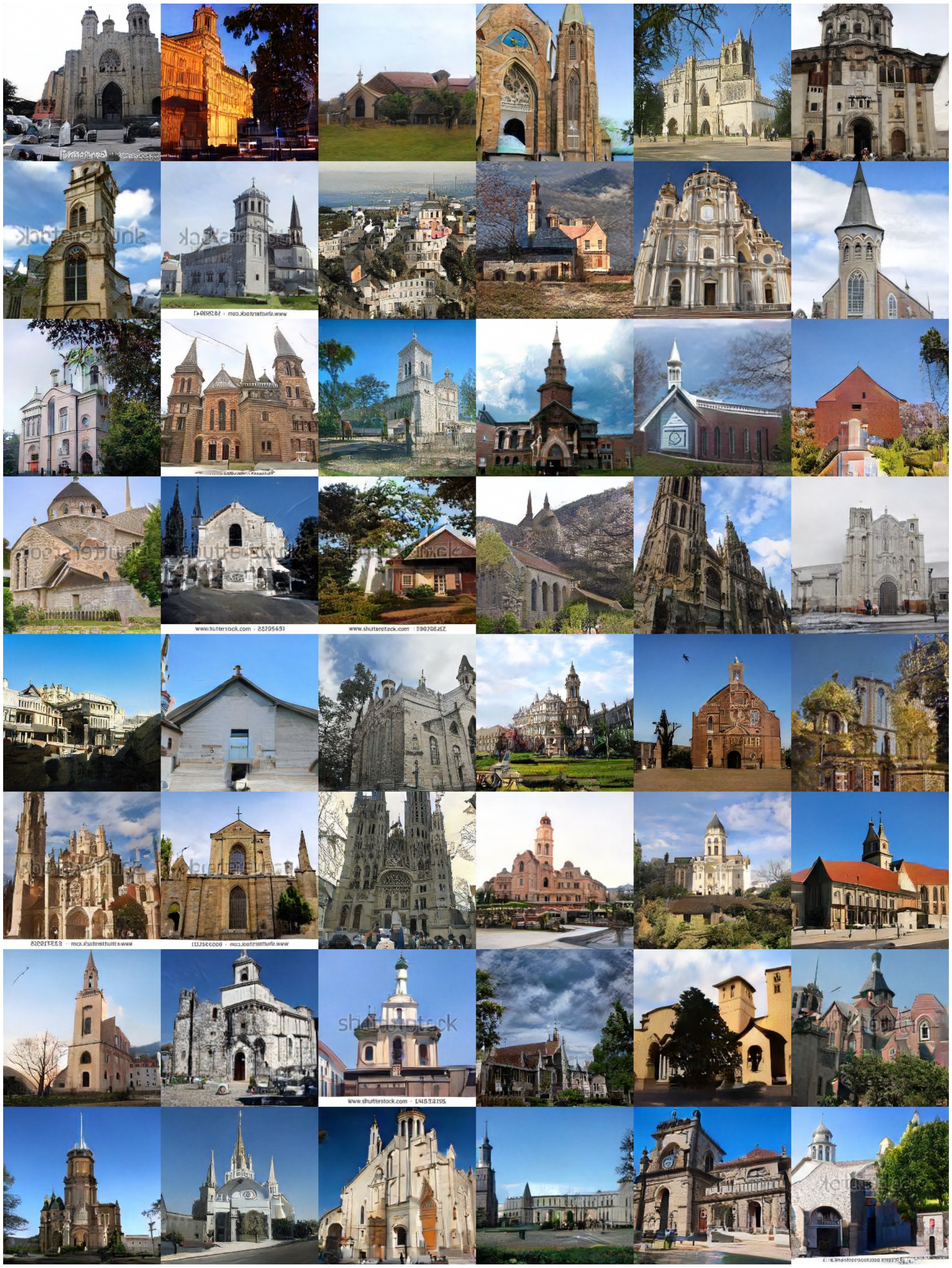


Figure 7. Non-curated generated samples on LSUN-Church

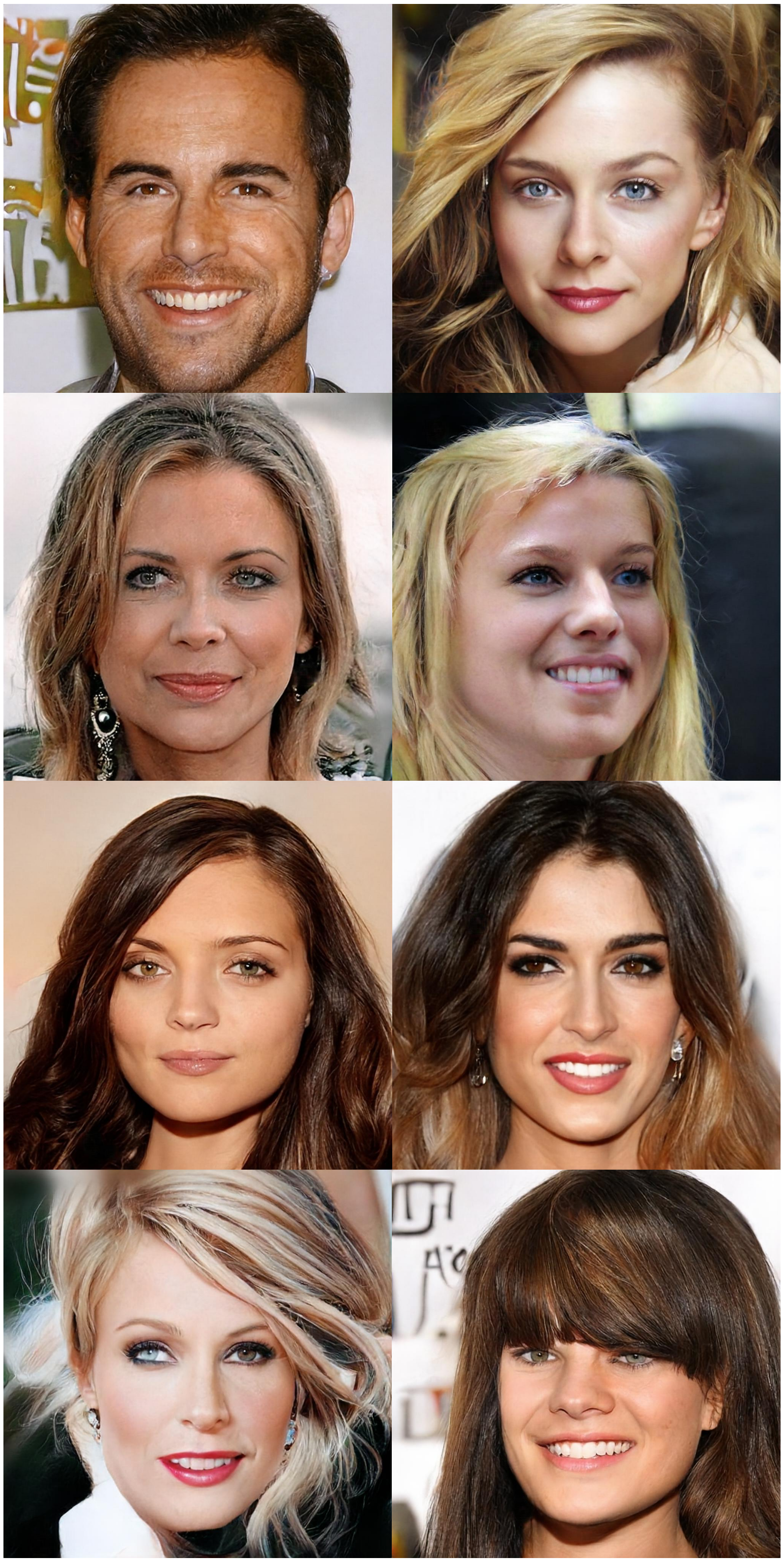


Figure 8. Non-curated generated samples on CelebA-HQ 1024×1024