A. Noise Scheduler

We show in Figure A that with different noise schedulers, how 1 (blue curves) and 0 (red curves) in the binary latent representations progressively reach 0.5 in the diffusion processes. We use the *Linear* noise scheduler in all the experiments.

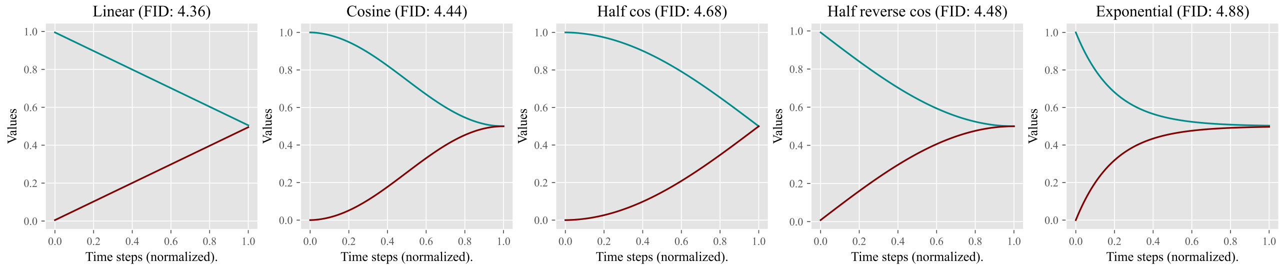


Figure A. Noise schedulers. FID numbers are obtained with the 256×256 LSUN Churches experiments.

B. Implementation Details

We use Adam as the default optimizer for all experiments. For the training of the binary auto-encoder in Section 3, we use a consistent learning rate of 5×10^{-4} with a linear learning rate warm-up for 10k iterations, and a batch size of 8. For $\mathcal{C}$ in (2), we use mean squared error, perception loss, and adversarial loss, with an equal $\omega_i = 1.0$ for each term.

For the training of the binary latent diffusion models in Section 4, we use a consistent learning rate of 1×10^{-4} with a linear learning rate warm-up for 10k iterations. All models are trained for 20K iterations. We do not use any additional regularizations such as drop-out or weight decay. We summarize the general training and sampling of the proposed binary diffusion models in Algorithm 1 and Algorithm 2, respectively.

All 256×256 unconditional binary latent diffusion models are trained with a batch size of 32. All 1024×1024 unconditional binary latent diffusion models are trained with a batch size of 16. All 256×256 conditional binary latent diffusion models are trained with a batch size of 256.

All training is conducted on cloud servers with V100 GPUs. All the speeding testing is conducted on a single RTX3090 GPU.

B.1. Network Architectures

For the binary auto-encoder in Section 3, we adopt a standard architecture using convolutional layers enhanced with self-attention layers, which is nearly identical to the one used in [16]. The only major difference is that the vector quantization layer is replaced by our binary latent layer implementing (1). This architecture gives a downsampling ratio of $16\times$. For the 1024×1024 high-resolution image generation experiments, of which the downsampling ratio raises to $32\times$, we simply insert one more downsampling block and one more upsampling block into the architecture.

For the conditional image generation experiments with ImageNet-1K, we train transformer networks with 24 layers and 768 feature channels with 12 heads in the self-attention layers. For the unconditional image generation experiments, we train transformer networks with 24 layers and 512 feature channels with 8 heads in the self-attention layers. In both conditional and unconditional image generation experiments, we use an extra time embedding token to specify the time step t. And the class labels in conditional image generation with ImageNet are also specified by an additional token. Both class tokens and time-step tokens are trainable. We initialize the trainable position embedding in the transformer networks with 2D sine embedding.

B.2. Evaluation Metrics

FID. Fréchet Inception Distance (FID) is a commonly used metric for evaluating the quality of the generated images. It estimates the distance between two image distributions by comparing the mean and standard deviation of the deep image features extracted by a trained Inception network [59]. We conduct comparisons by generating 50,000 samples and comparing them with the corresponding training dataset in every experiment.

PR. To comprehensively evaluate both the quality and mode-coverage of the generated samples compared to the training datasets, we further include precision-recall [36] as additional performance measurements.

IS. Inception score [52] is a commonly used metric for evaluating the performance of class-conditional image generation. It favors generated images with low entropy of label predictions and diverse labels given a pretrained Inception-V3 network.

CAS. Classification accuracy score [45] works by first training a ResNet-50 [22] using the generated image across classes, and measuring the results of applying the trained classifier to the ImageNet validation set. CAS offers a comprehensive measurement of the generative quality as a robust classifier demands the generated images used for network training to be both diverse and of high quality.

Algorithm 1 Training procedure. We assume unconditional image generation with a batch size of *one* for the sake of discussion. The described training process can be easily extended to practical cases with arbitrary batch sizes by batching multiple samples.

```
Given: Trained encoder $\boldsymbol{\Psi}$; Binary diffusion model $f_\theta$ parametrized by $\mathcal{T}_\theta$; An image dataset $\mathbf{X}$.
Given: Diffusion steps $T$; Noise scheduler defined by $\{k^t\}_{t=1}^{T}$ and $\{b_t\}_{t=1}^{T}$; Training steps $I$; and $\lambda$ in (13).
Initializing $\mathcal{T}_\theta$.
for Step $i = 1 : I$ do
    Sampling image $\mathbf{x} \sim \mathbf{X}$, and time step $t \sim \{1, \dots, T\}$.
    Obtaining binary code $\mathbf{z}^0 = \text{Bernoulli}(\sigma(\boldsymbol{\Psi}(\mathbf{x})))$.
    Obtaining $\mathbf{z}^t$ using $\mathbf{z}^0$, $t$, and noise scheduler with (5).
    Predicting flipping probability $f_\theta(\mathbf{z}^t, t)$.
    Obtaining predicted $\mathbf{z}^0$ as $p_\theta(\mathbf{z}^0) = (1 - \mathbf{z}^t) \odot f_\theta(\mathbf{z}^t, t) + \mathbf{z}^t \odot (1 - f_\theta(\mathbf{z}^t, t))$.
    Obtaining predicted $p_\theta(\mathbf{z}^{t-1})$ using $p_\theta(\mathbf{z}^0)$ and $\mathbf{z}^t$ with (8).
    Calculating loss $\mathcal{L}$ using (13).
    Backpropagating $\mathcal{L}$ and updating $\theta$.
end for
Return Binary diffusion model $f_\theta$.
```

Algorithm 2 Sampling procedure. We assume unconditional image generation with a batch size of *one* for the sake of discussion.

```
Given: Trained decoder $\boldsymbol{\Phi}$; Trained binary diffusion model $f_\theta$.
Given: Diffusion steps $T$; Noise scheduler defined by $\{k^t\}_{t=1}^{T}$ and $\{b_t\}_{t=1}^{T}$; Temperature $\tau$; Latent dimension specified by $h', w', c$,
    e.g., $h' = w' = 16, c = 32$ for the 256 × 256 image generation experiments.
Sampling $\mathbf{z}^T = \text{Bernoulli}(\mathbf{z}^{\text{init}})$, where $\mathbf{z}^{\text{init}} \in \mathbb{R}^{h' \times w' \times c}$ and contains $0.5$ only.
for Step $t = T : 2$ do
    Predicting $p_\theta(\mathbf{z}^{t-1})$ with $f_\theta(\mathbf{z}^t, t) = \sigma(\mathcal{T}_\theta(\mathbf{z}^t, t)/\tau)$ and (8).
    Sampling $\mathbf{z}^{t-1} = \text{Bernoulli}(p_\theta(\mathbf{z}^{t-1}))$
end for
Return the sampled image as $\boldsymbol{\Phi}(\mathbf{z}^{t-1})$.
```

C. Classifier-free Guidance

Classifier-free guidance [25] is introduced to improve the generation fidelity and promote high correspondence to the conditions for conditional diffusion models. While it is previously believed that classifier-free guidance can hardly be extended to diffusion models, we show that the reparameterization to the prediction target as the binary residuals allows classifier-free guidance to apply seamlessly to the proposed binary latent diffusion for improved image fidelity in conditional image generation. In conditional image generation, we introduce an extra condition token c that carries the conditions such as class embedding in ImageNet class-conditional image generation or text embedding in the text-to-image generation. The model performs class-conditional prediction as $f_\theta^{\text{c}}(\mathbf{z}^t, t, c) = \sigma(\mathcal{T}_\theta(\mathbf{z}^t, t, c))$. An unconditional prediction can be drawn by simply dropping the condition token c as $f_\theta^{\text{u}}(\mathbf{z}^t, t) = \sigma(\mathcal{T}_\theta(\mathbf{z}^t, t))$. The final prediction at each step with classifier-free guidance can then be implemented as $f_\theta(\mathbf{z}^t, t, c) = \sigma((1 + \omega)\mathcal{T}_\theta(\mathbf{z}^t, t, c) - \omega\mathcal{T}_\theta(\mathbf{z}^t, t))$, where ω is a non-negative scalar controlling the strength of the guidance. Note that same the temperature scale τ, the guidance strength ω is only effective in the sampling stage and is not involved in training, which allows us to arbitrarily adjust the sampling quality without retraining the models. While it is inevitable that the classifier-free guidance doubles that computation cost at each sampling step due to the extra unconditional predictions, we consistently observe performing classifier-free guidance allows us to skip sampling steps. For example, performing only a quarter of the sampling steps enhanced with classifier-free guidance reduce the overall computation cost by half, and does not noticeably reduce the sample quality. We present in Appendix Figure M results generated with different scales of classifier-free guidance.

D. Additional Qualitative Results

D.1. Unconditional Image Generation

We present additional qualitative results of 256×256 LSUN Bedrooms and LSUN Churches unconditional image generation experiments in Figure B and Figure C, and qualitative comparisons of FFHQ unconditional image generation against Absorbing Diffusion models [6] and Latent Diffusion Models [49] in Figure D.

D.2. High-resolution Image Generation

We present additional qualitative results of 1024×1024 unconditional image generation experiments in Figure E and Figure F.

D.3. Conditional Image Generation

We present additional qualitative results of class-conditional image generation experiments and comparisons with BigGAN-deep [7], VQ-VAE-2 [46], VQGAN [16] and MaskGIT [8] in Figure I and Figure L.

D.4. Image Inpainting

We present additional image inpainting results with different mask patterns in Figure N.

D.5. Nearest Neighbors

To further show that our model is generating novel samples instead of overfitting to the training datasets, we compare the generated images with the corresponding training datasets using LPIPS, and visualize the top-10 nearest neighbors in Figure O.

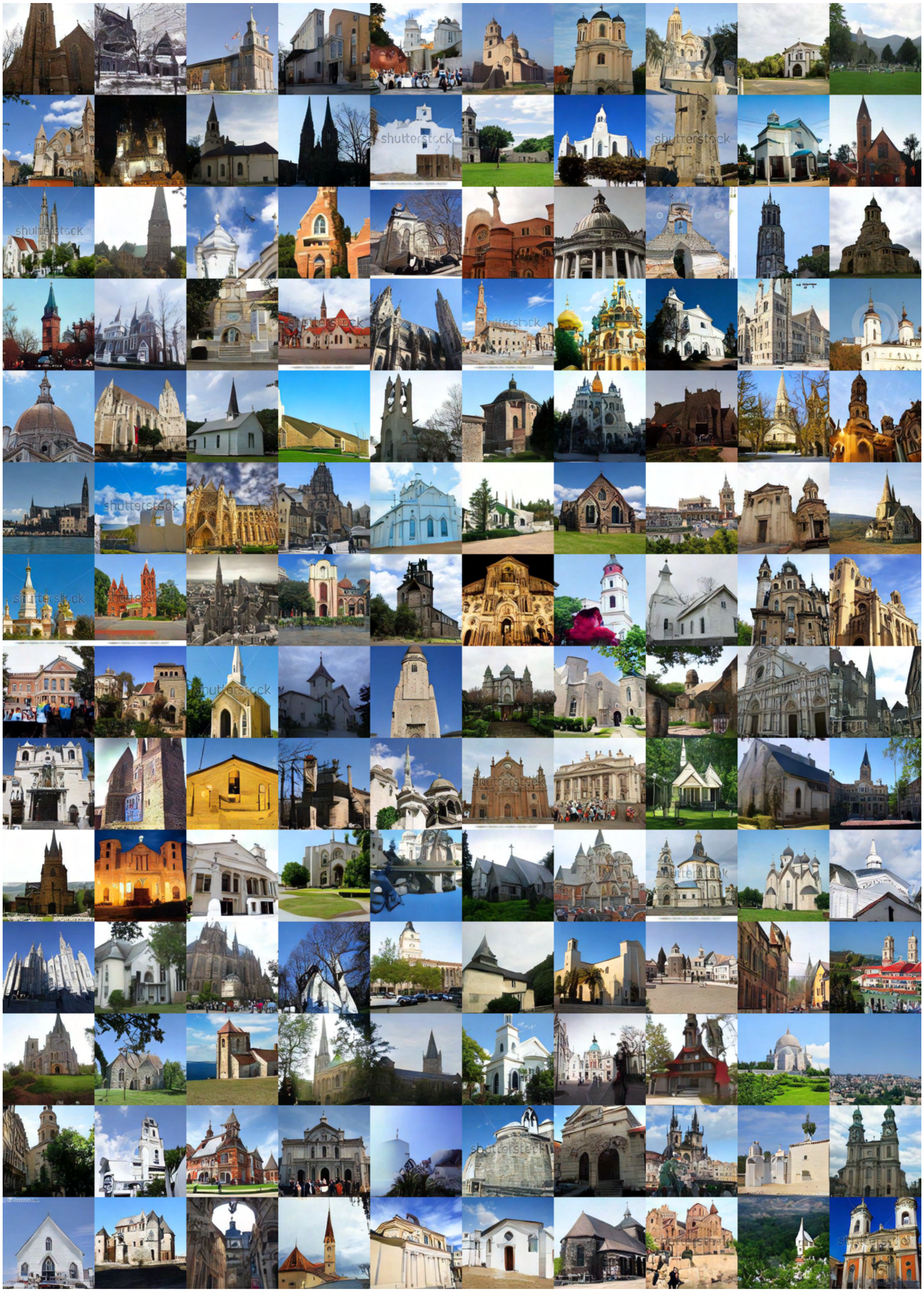


Figure B. Additional unconditional image generation results and comparisons at 256×256 with the LSUN Churches dataset.

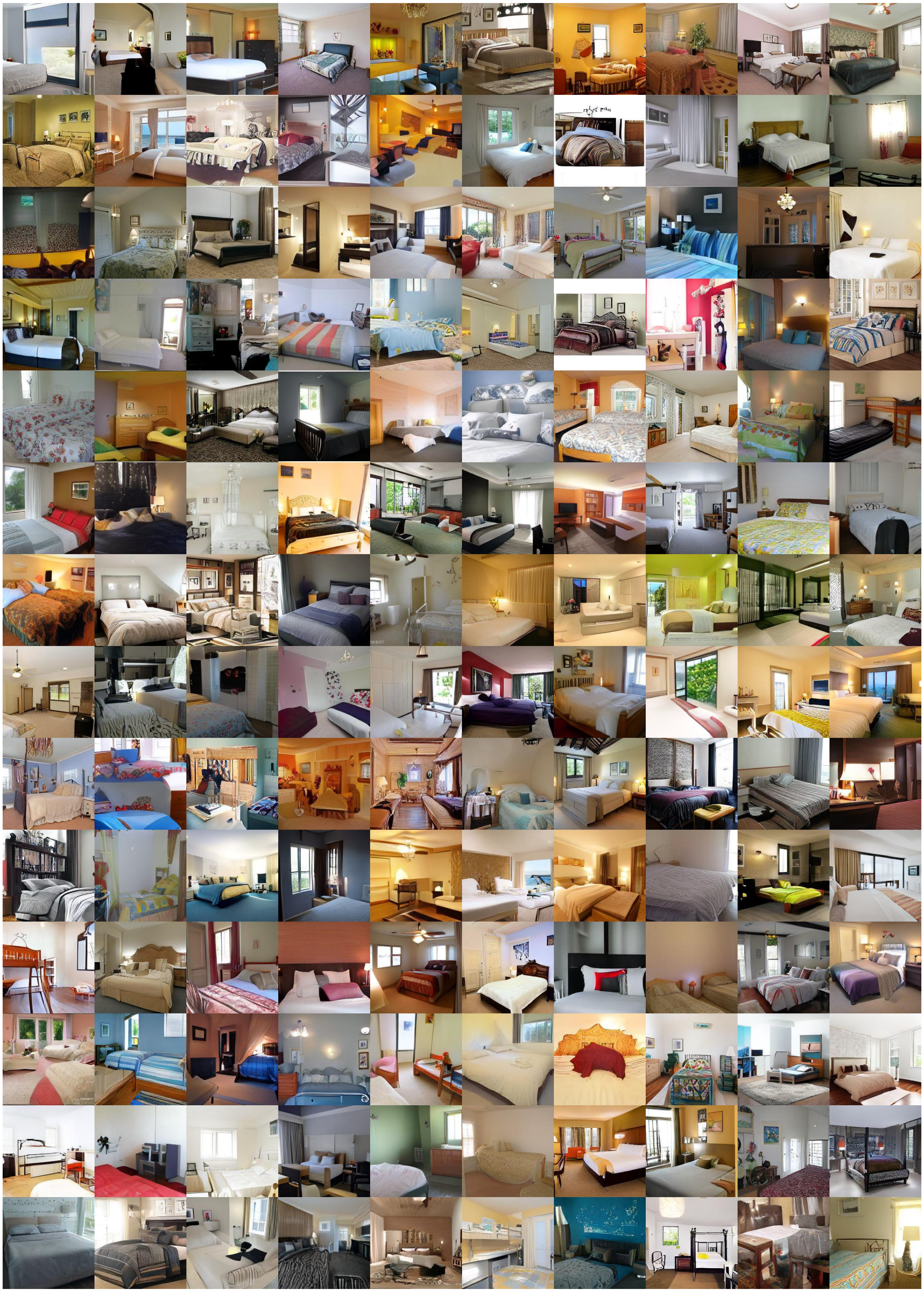
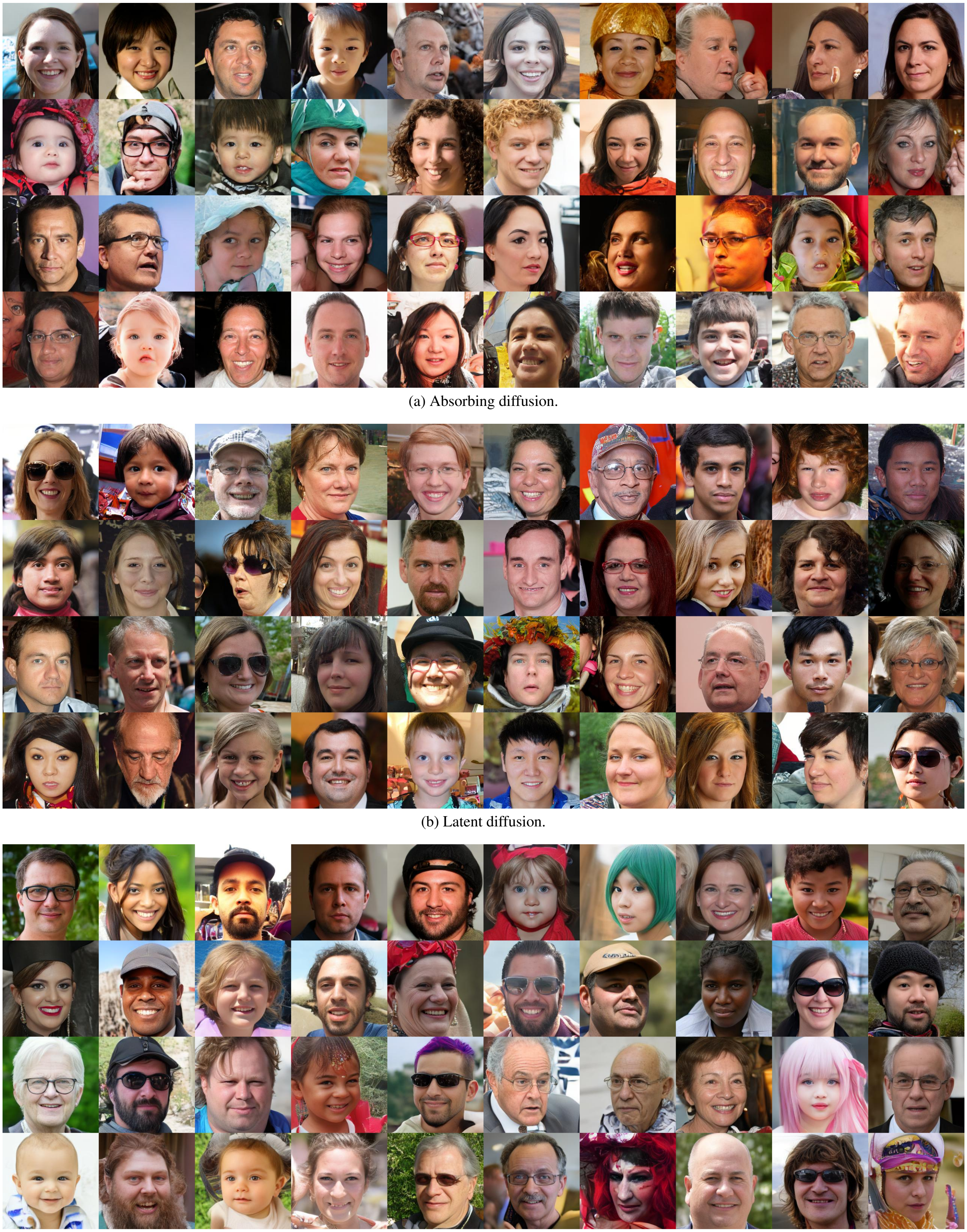


Figure C. Additional unconditional image generation results and comparisons at 256 × 256 with the LSUN Bedrooms dataset.



(a) Absorbing diffusion.

(b) Latent diffusion.

(c) Ours.

Figure D. Additional unconditional image generation results and comparisons at 256 × 256 with the FFHQ dataset.

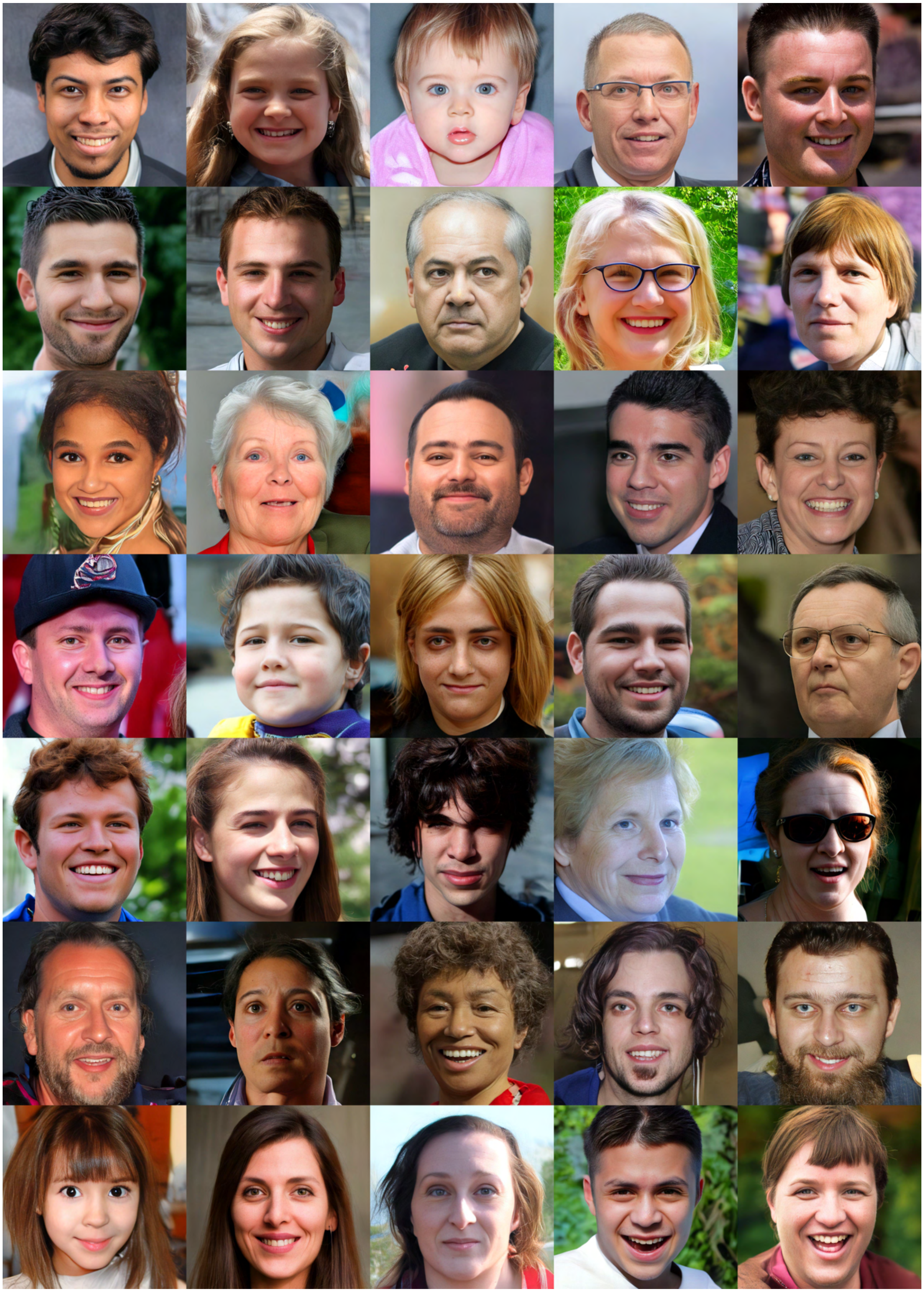


Figure E. Additional high-resolution image generation results at 1024×1024 with the FFHQ dataset ($\tau = 0.8$).

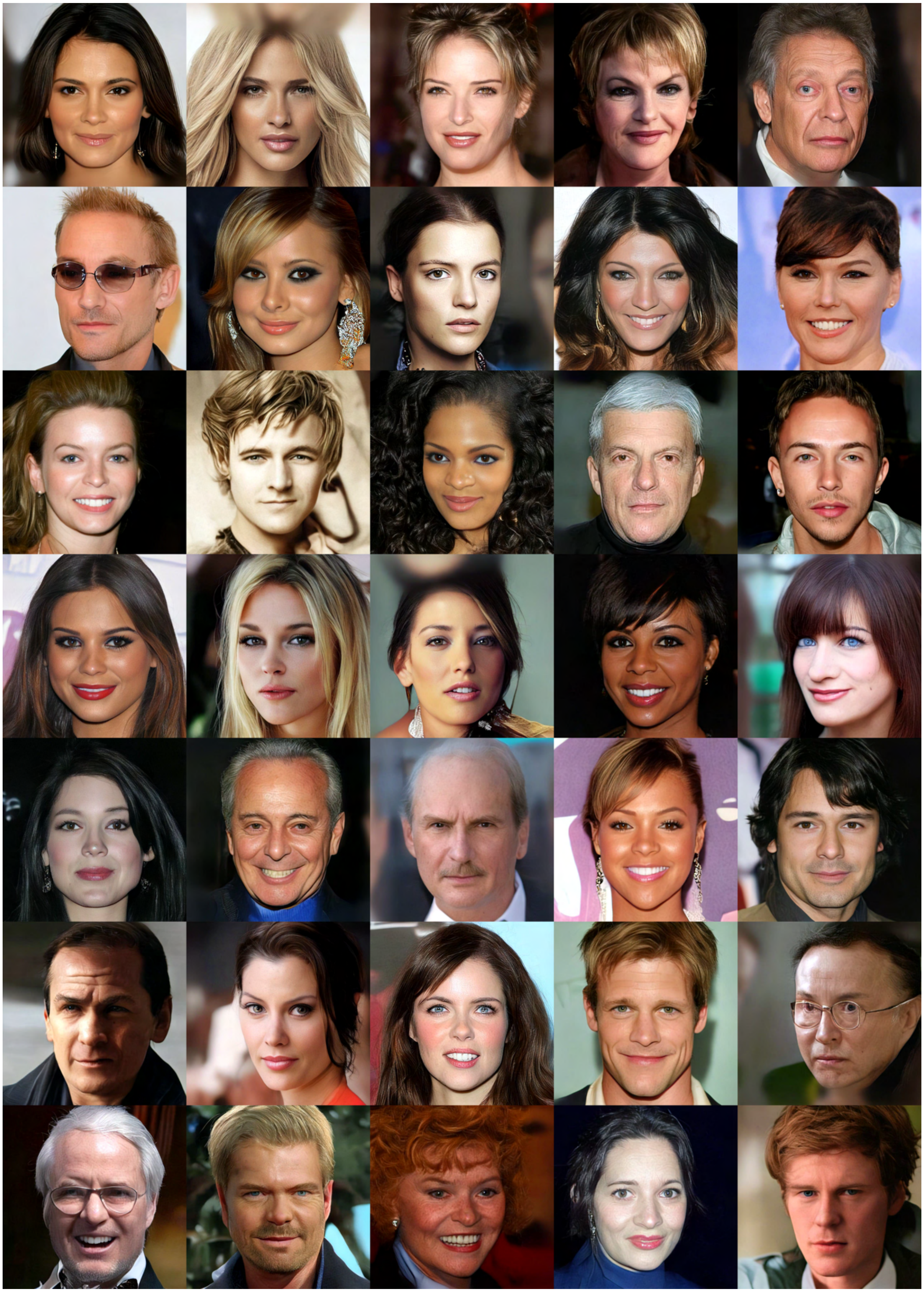
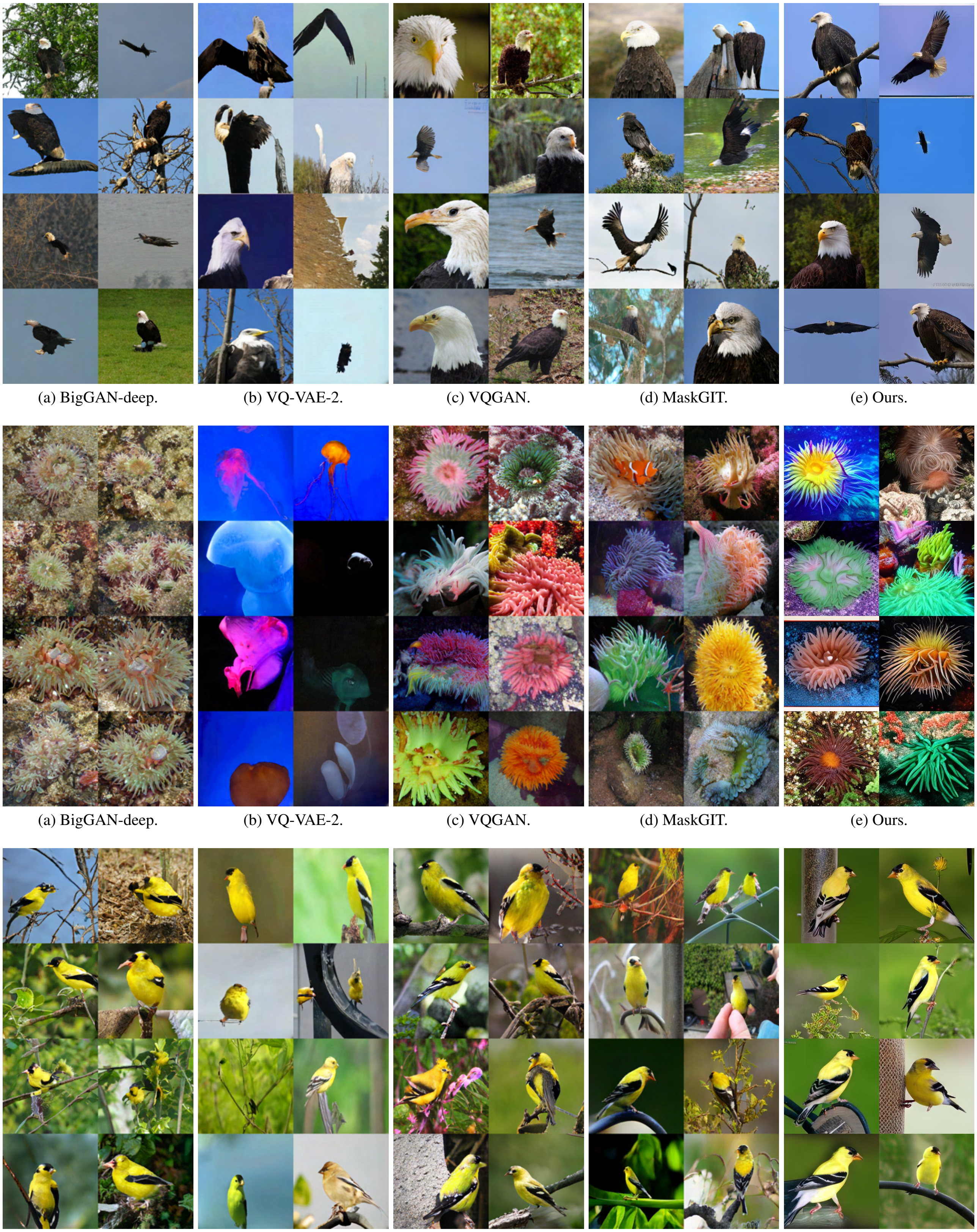


Figure F. Additional high-resolution image generation results at 1024×1024 with the CelebA-HQ dataset ($\tau = 0.8$).

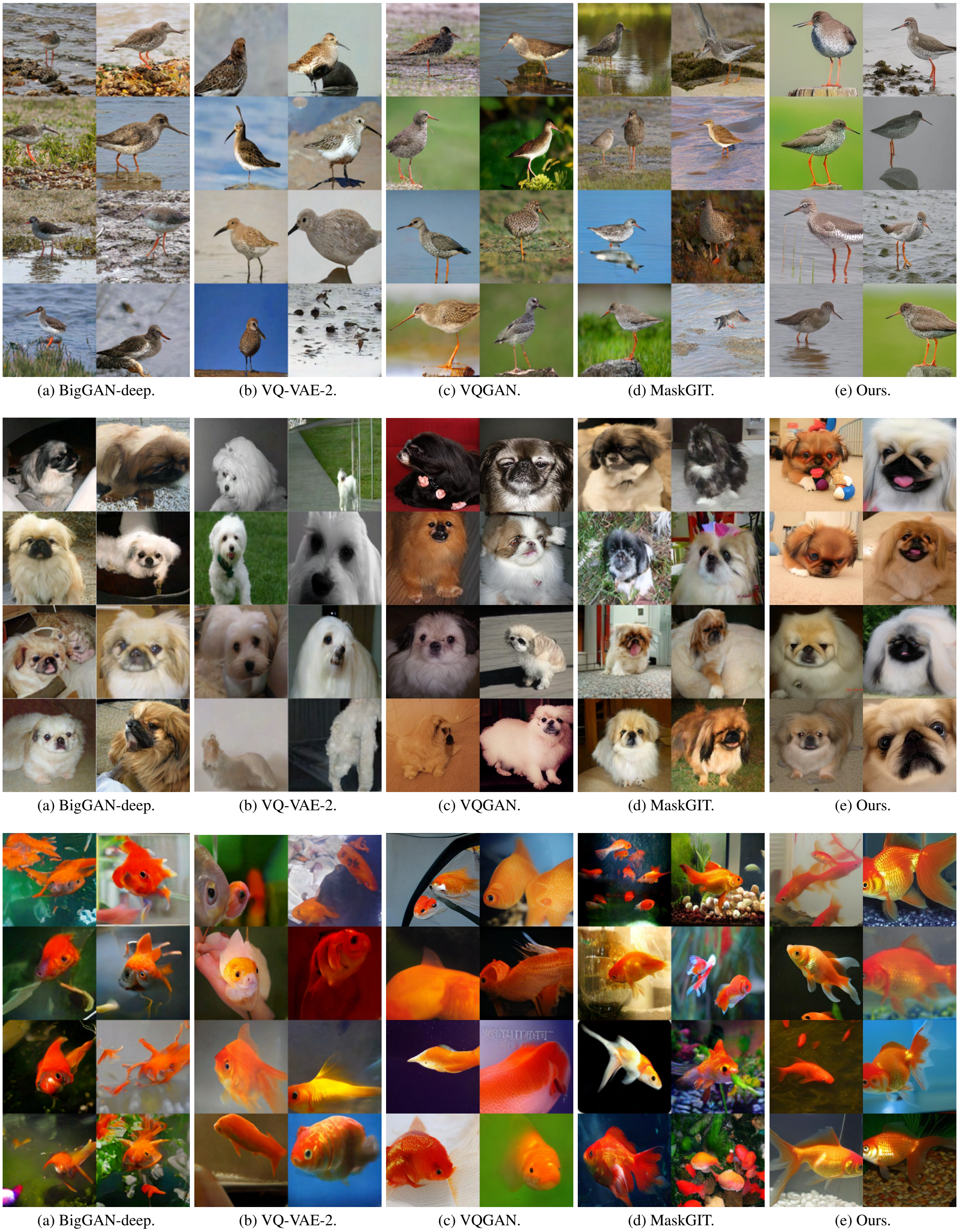


(a) BigGAN-deep. (b) VQ-VAE-2. (c) VQGAN. (d) MaskGIT. (e) Ours.

(a) BigGAN-deep. (b) VQ-VAE-2. (c) VQGAN. (d) MaskGIT. (e) Ours.

(a) BigGAN-deep. (b) VQ-VAE-2. (c) VQGAN. (d) MaskGIT. (e) Ours.

Figure I. Qualitative comparisons on class-conditional image generation with ImageNet class IDs: 22, 108, and 11.



(a) BigGAN-deep. (b) VQ-VAE-2. (c) VQGAN. (d) MaskGIT. (e) Ours.

(a) BigGAN-deep. (b) VQ-VAE-2. (c) VQGAN. (d) MaskGIT. (e) Ours.

(a) BigGAN-deep. (b) VQ-VAE-2. (c) VQGAN. (d) MaskGIT. (e) Ours.

Figure L. Qualitative comparisons on class-conditional image generation with ImageNet class IDs: 141, 154, and 1.

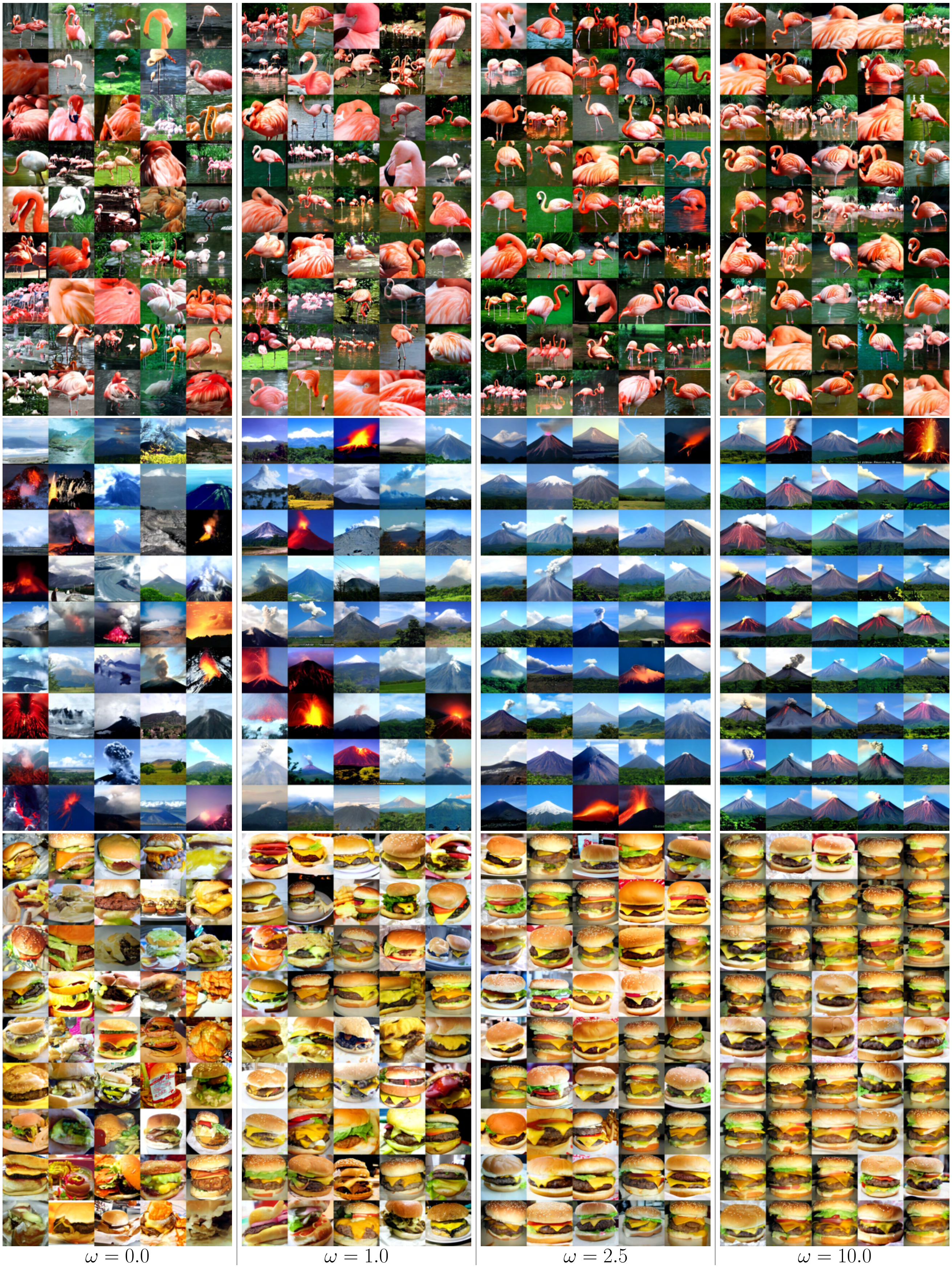


Figure M. Label-conditioned image generation with different scales of classifier-free guidance (ω). Larger ω improves the sample quality at the cost of lower diversity.

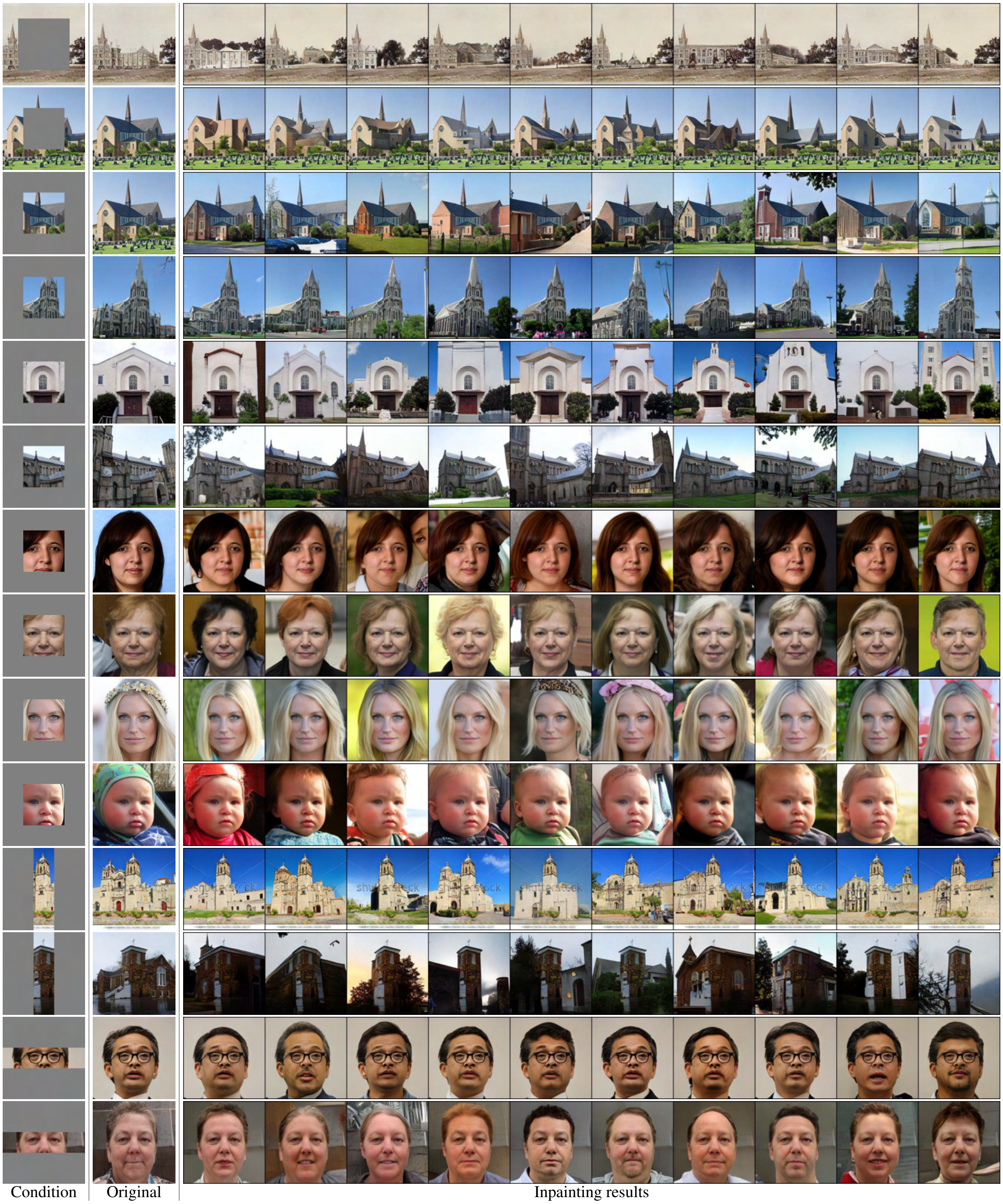


Figure N. Conditional image inpainting with different masking patterns.

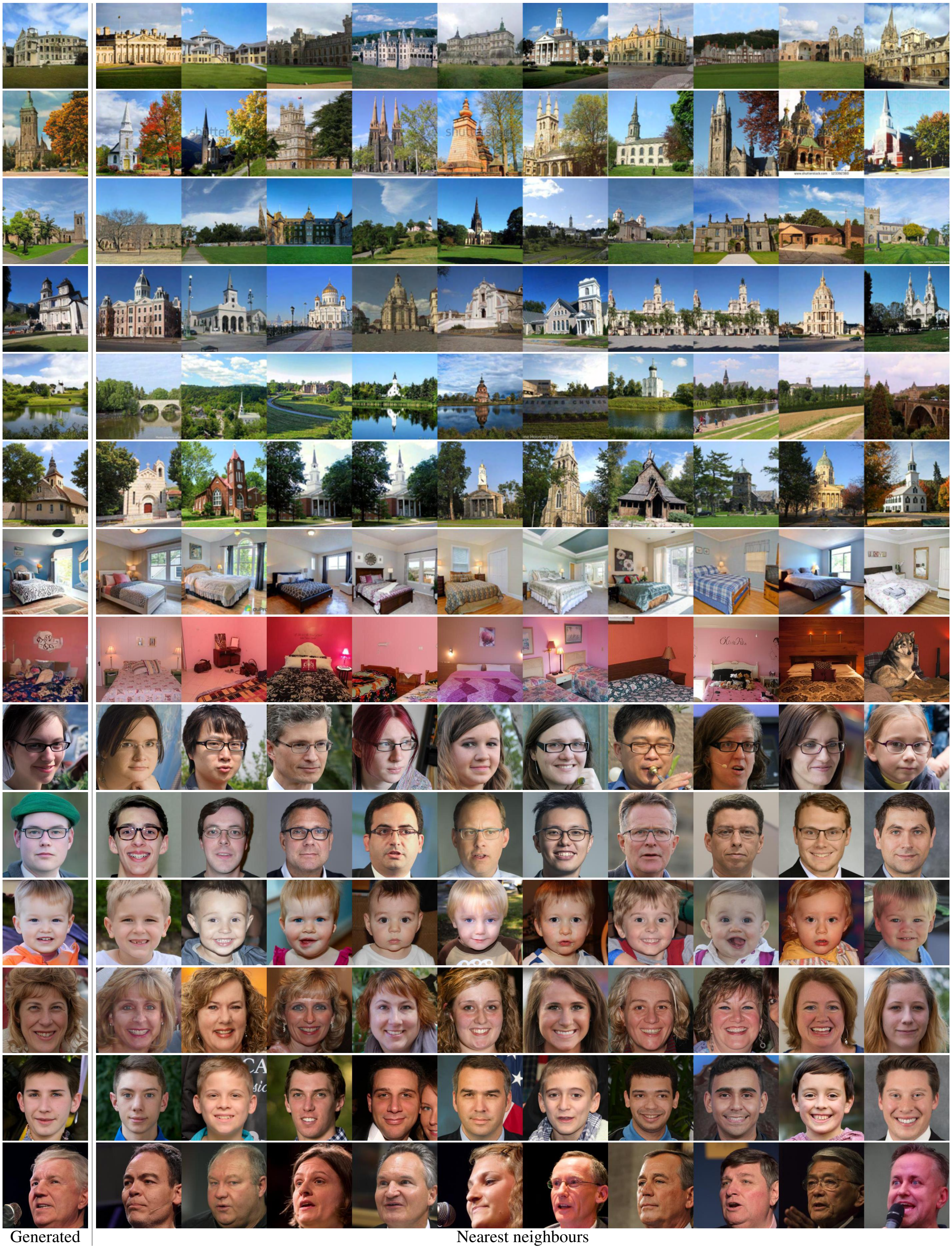


Figure O. Top-10 nearest neighbours in the training datasets of our generated samples. Results show that our model is not overfitting to the training datasets.