Exploring Sparse MoE in GANs for Text-conditioned Image Synthesis

Supplementary Material

Appendix

A. Overview

This paper proposes Aurora that investigates how to expand the capacity of GANs to match the demand of open-vocabulary text-to-image generation. In this supplementary material, we first give the training details of our method including the base generator and upsampler in App. B. Second, more results are given in App. C, including more qualitative results and comparisons with other methods. Third, we give an additional ablation study on the upsampler in App. D.

B. Training Details

Base generator. During training, we employ the non-saturating loss function [23] combined with R_1 regularization using $\gamma = 0.2048$. The discriminator architecture follows that of GigaGAN [34] and incorporates both multi-scale inputs and multi-scale output losses. For the adversarial loss, we compute the loss at each input resolution (4, 8, 16, 32, and 64), assigning weights of 0.17, 0.17, 0.17, 0.17, and 0.33 respectively. Regarding the output loss, we calculate it at resolutions of 1, 4, 8, 16, and 32, then sum all the contributions. The matching weights are configured identically to the adversarial loss weights. Additionally, we set the clip weight to 16 for each image resolution. The number of experts is set to 4, 8, 12, and 12 for the resolutions of 8, 16, 32, and 64, respectively.

Upsampler. When training the upsampler, we still use the non-saturating loss [23] with R_1, setting $\gamma = 0.01$. The adversarial loss is calculated only at the highest resolution (*i.e.,* 512×512 for our model), while the output loss is computed at resolutions of $1, 4, 8, 16, 32$ of the discriminator, and calculating the loss on higher branches (such as compute loss on 64 resolution) offers negligible performance improvements according to our empirical findings. The perceptual loss weight is set to 4. The adversarial loss is involved once the upsampler has processed the entire training dataset (*i.e.,* 120M images in our case). For data augmentation, we apply blurring using a kernel size of 7 and a sigma value of 0.9. Additionally, we introduce random noise with a diffusion maximum step set to 50.

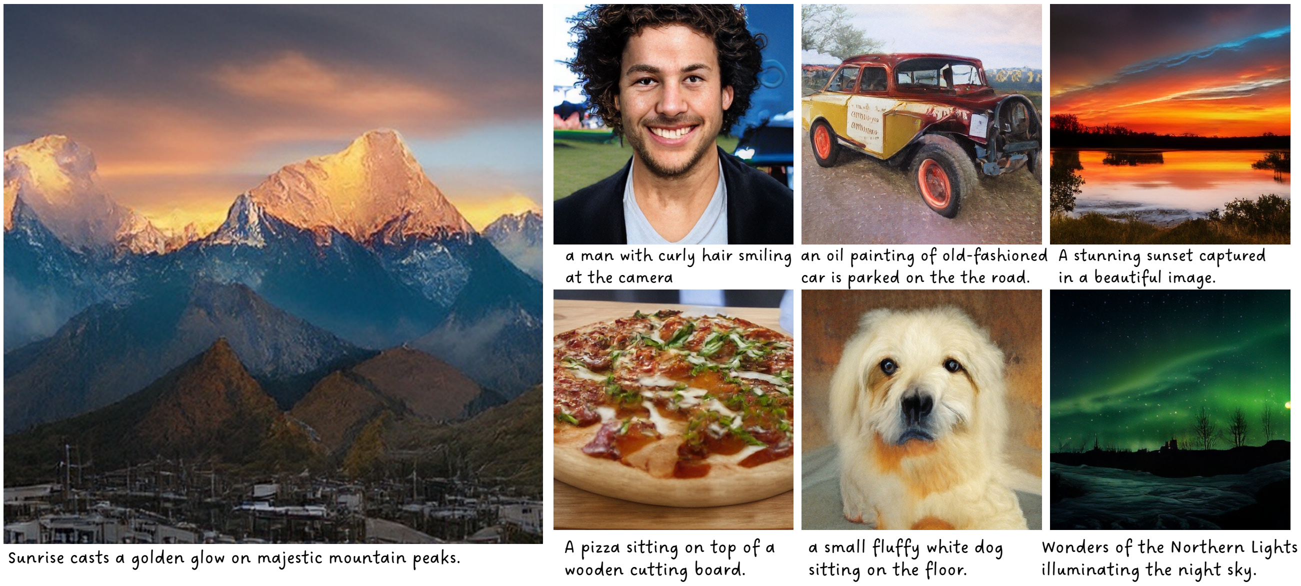
C. More Results

In this section, we provide more qualitative results of our method. Fig. S1 showcases a variety of diverse images generated by our approach. Fig. S2, Fig. S3, Fig. S4 present comparative results with Stable Diffusion [66]. For Stable Diffusion, we utilize V1.5 and 50 sampling steps as outlined in DDIM [88], with a guidance scale of 7.5 for comparison. Fig. S2 illustrates that in scene generation, our model is capable of producing more photorealistic images than Stable Diffusion. When it comes to generating buildings or simple objects, such as the pizza in the third row of Fig. S3, our method performs comparably to Stable Diffusion. In Fig. S4, we compare both methods in human generation, revealing that neither approach synthesizes humans effectively. Fig. S5 highlights several failure cases of our method in comparison to Stable Diffusion. These examples highlight our model's challenges in generating complex images, especially those with fine-grained details, such as the tofu and peas in the first row of Fig. S5, and scenarios involving multiple objects or intricate shapes, as seen in the second and third rows of Fig. S5.

Fig. S6 and Fig. S7 illustrate the results of interpolating text prompts. Specifically, Fig. S6 shows the interpolation results when the two prompts differ significantly, while Fig. S7 displays the results when the prompts have a minor difference. In both scenarios, a smooth semantic transition is evident, even with significant differences between prompts, as seen in the second row of Fig. S6 where the interpolation occurs between the prompts "A photo of a Victorian house" and "A photo of the moon," the interpolated images in the middle exhibit the semantics of both a building and the moon. Fig. S8 showcase of the interpolation results between the latent code, *i.e.,* in $\mathcal{W}$ space. Here, we can observe a smooth transition in the interpolated outcomes, regardless of the generated content, whether it be objects, buildings, or entire scenes.

D. Ablation Study

As outlined in Sec. 3.3 of the main paper, our upsampler diverges from the previous U-Net architecture [34, 66] in three key aspects: *removal of U-Net connections*, *input data augmentation*, and a *higher downsampling rate.* We provide the quantitative results of implementing these changes in Tab. S1. Each modification significantly impacts the FID [27] score. Furthermore, the final fine-tuning on the synthesized images also substantially influences performance.



Sunrise casts a golden glow on majestic mountain peaks.

a man with curly hair smiling at the camera

an oil painting of old-fashioned car is parked on the the road.

A stunning sunset captured in a beautiful image.

A pizza sitting on top of a wooden cutting board.

a small fluffy white dog sitting on the floor.

Wonders of the Northern Lights illuminating the night sky.

Figure S1. **Diversity of synthesized images using Aurora**, a large-scale GAN-based text-to-image generator.

Table S1. **Ablation study** of our upsampler, which targets generating images at 512×512 resolution. "zero-shot FID_{30K}" [27] on MS COCO [46] is employed as the evaluation metric, where a smaller number indicates better performance.

Model	Zero-shot FID_{30K}	# Params.
Base	19.04	0.301B
+ Remove U-Net	15.76	0.301B
+ Augmentation	12.80	0.301B
+ Higher Downsample Rate	10.46	0.336B
+ Finetuning	8.74	0.336B

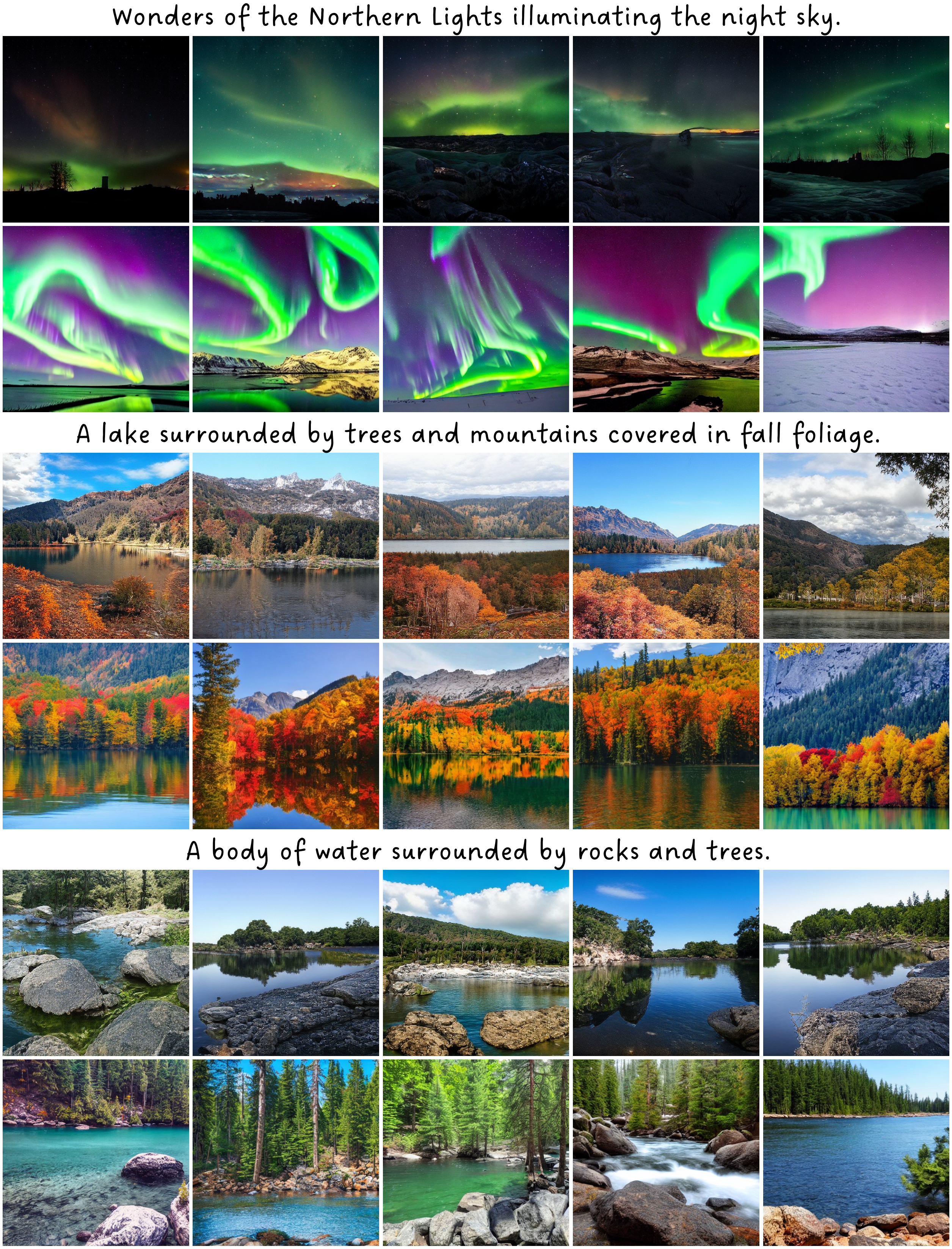

Wonders of the Northern Lights illuminating the night sky.
A lake surrounded by trees and mountains covered in fall foliage.
A body of water surrounded by rocks and trees.

Figure S2. Comparison with Stable Diffusion [66] on scene image generation using different prompts. In each group, the first row displays the results from our method, while the second row shows the outputs from Stable Diffusion.

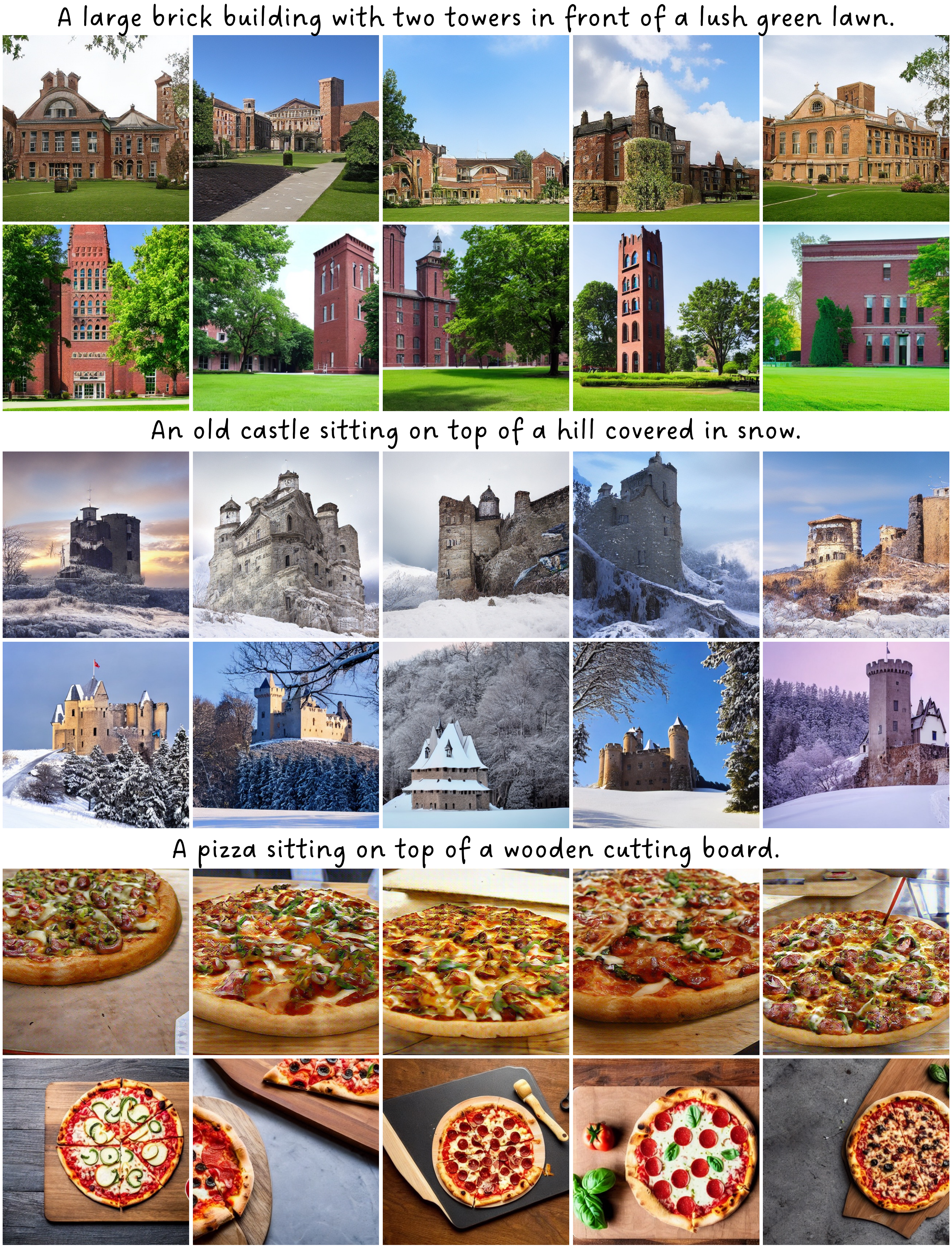
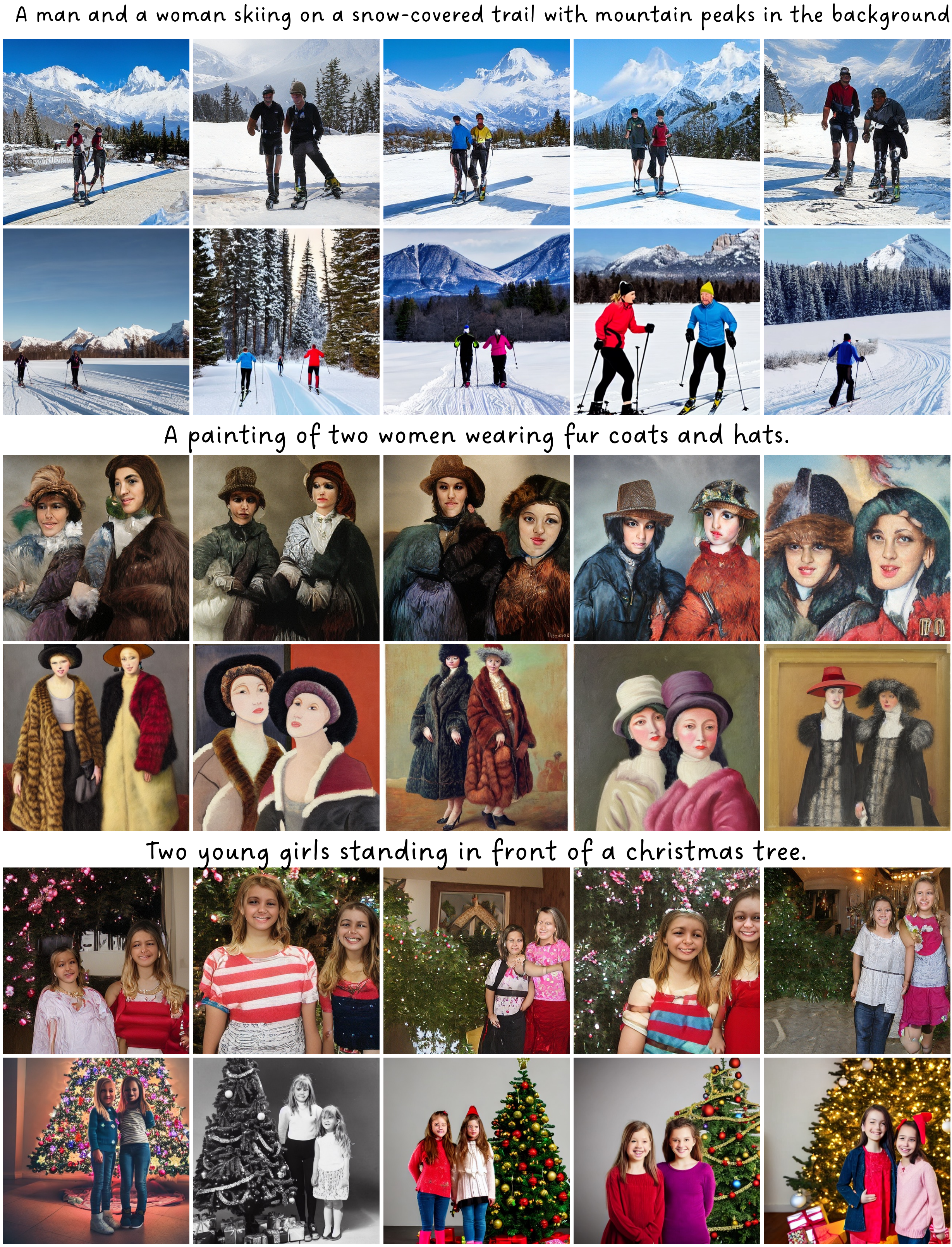

A large brick building with two towers in front of a lush green lawn.
An old castle sitting on top of a hill covered in snow.
A pizza sitting on top of a wooden cutting board.

Figure S3. Comparison with Stable Diffusion [66] using different prompts. In each group, the first row displays the results from our method, while the second row shows the outputs from Stable Diffusion.

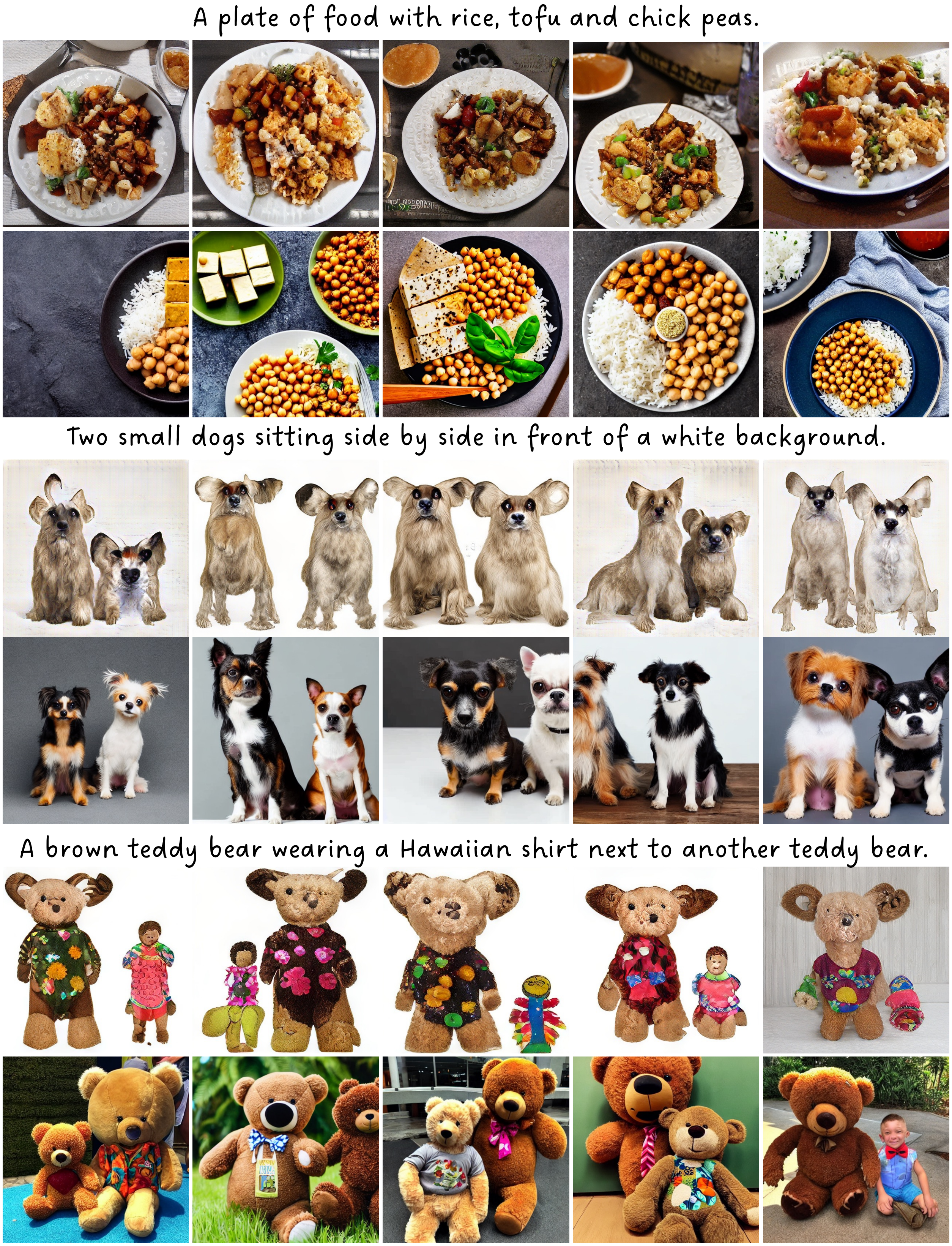


A man and a woman skiing on a snow-covered trail with mountain peaks in the background.

A painting of two women wearing fur coats and hats.

Two young girls standing in front of a christmas tree.

Figure S4. Comparison with Stable Diffusion [66] on human image generation using different prompts. In each group, the first row displays the results from our method, while the second row shows the outputs from Stable Diffusion.



A plate of food with rice, tofu and chick peas.

Two small dogs sitting side by side in front of a white background.

A brown teddy bear wearing a Hawaiian shirt next to another teddy bear.

Figure S5. **Failure cases** study under different prompt. In each group, the first row displays the results from our method, while the second row shows the outputs from Stable Diffusion.

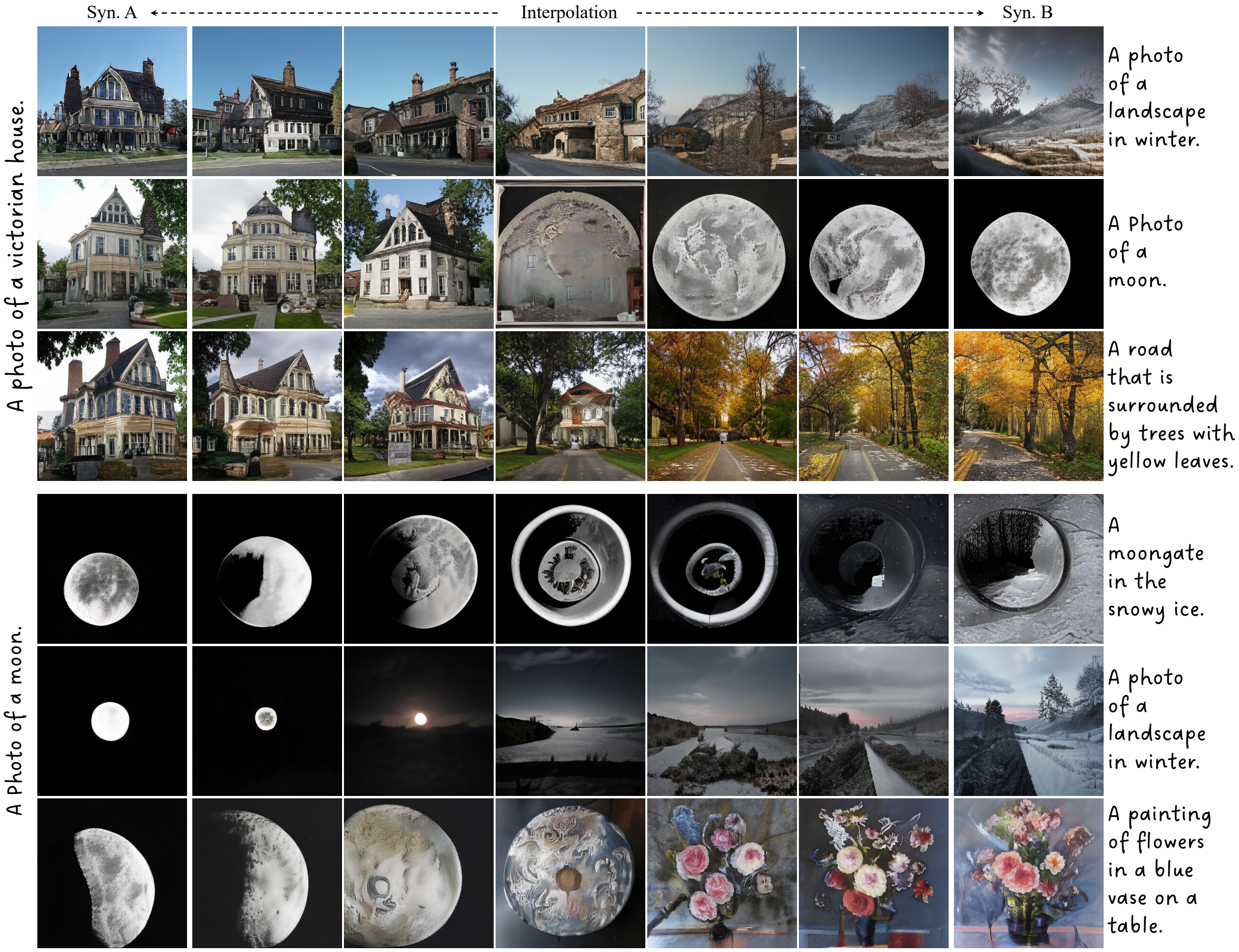

Syn. A
Interpolation
Syn. B
A photo of a victorian house.
A photo of a landscape in winter.
A Photo of a moon.
A road that is surrounded by trees with yellow leaves.
A Photo of a moon.
A moongate in the snowy ice.
A photo of a landscape in winter.
A painting of flowers in a blue vase on a table.

Figure S6. **Synthesized results through text prompt interpolation**, where we fix the global latent code, $\mathbf{z}$, and interpolate the text tokens, $\{\mathbf{t}_{seq}, \mathbf{t}_g\}$, extracted from two different text conditions, $\mathbf{c}$. `Aurora` enables smooth interpolation between prompts, even when the two input prompts are significantly different.

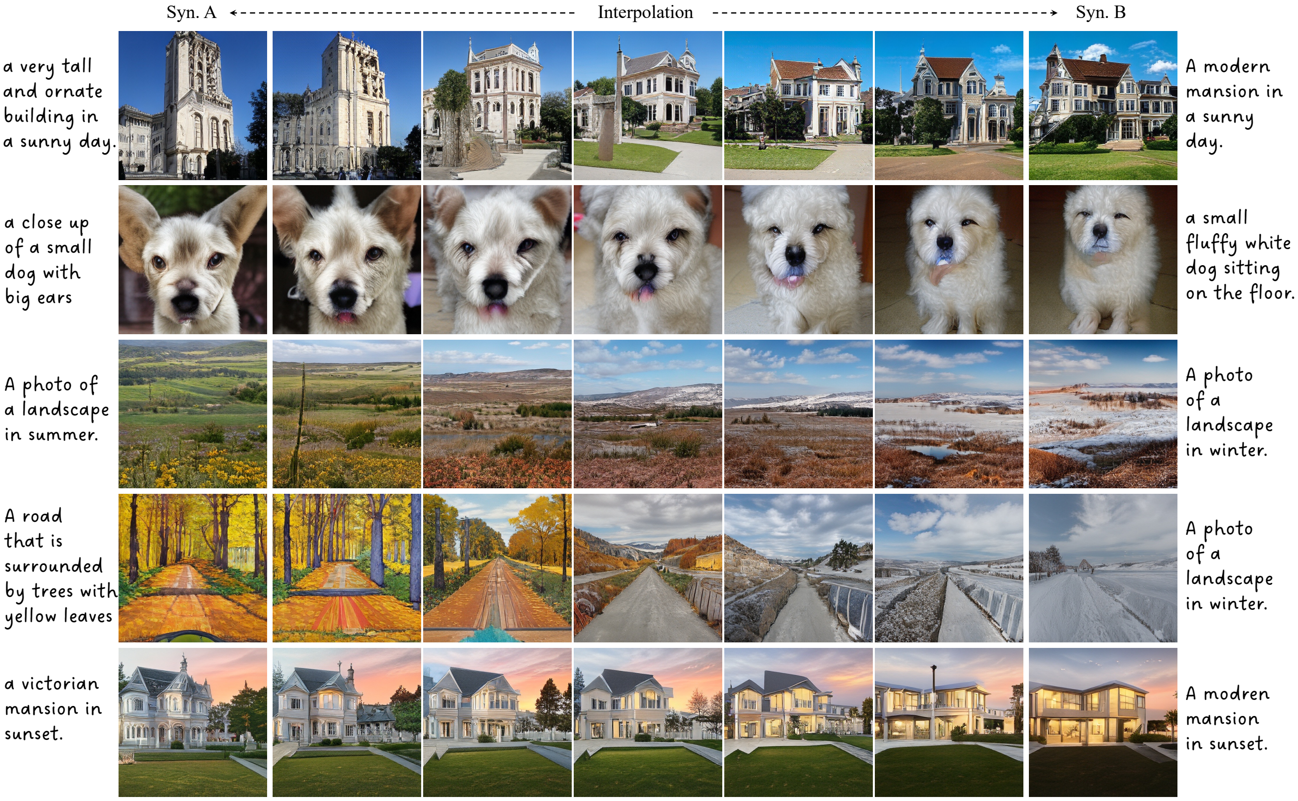

Syn. A
Interpolation
Syn. B
a very tall and ornate building in a sunny day.
A modern mansion in a sunny day.
a close up of a small dog with big ears
a small fluffy white dog sitting on the floor.
A photo of a landscape in summer.
A photo of a landscape in winter.
A road that is surrounded by trees with yellow leaves
A photo of a landscape in winter.
a victorian mansion in sunset.
A modren mansion in sunset.

Figure S7. **Synthesized results through text prompt interpolation**, where we fix the global latent code, $\mathbf{z}$, and interpolate the text tokens, $\{\mathbf{t}_{seq}, \mathbf{t}_g\}$, extracted from two different text conditions, $\mathbf{c}$. `Aurora` enables smooth interpolation between prompts, even when the two input prompts are significantly different.

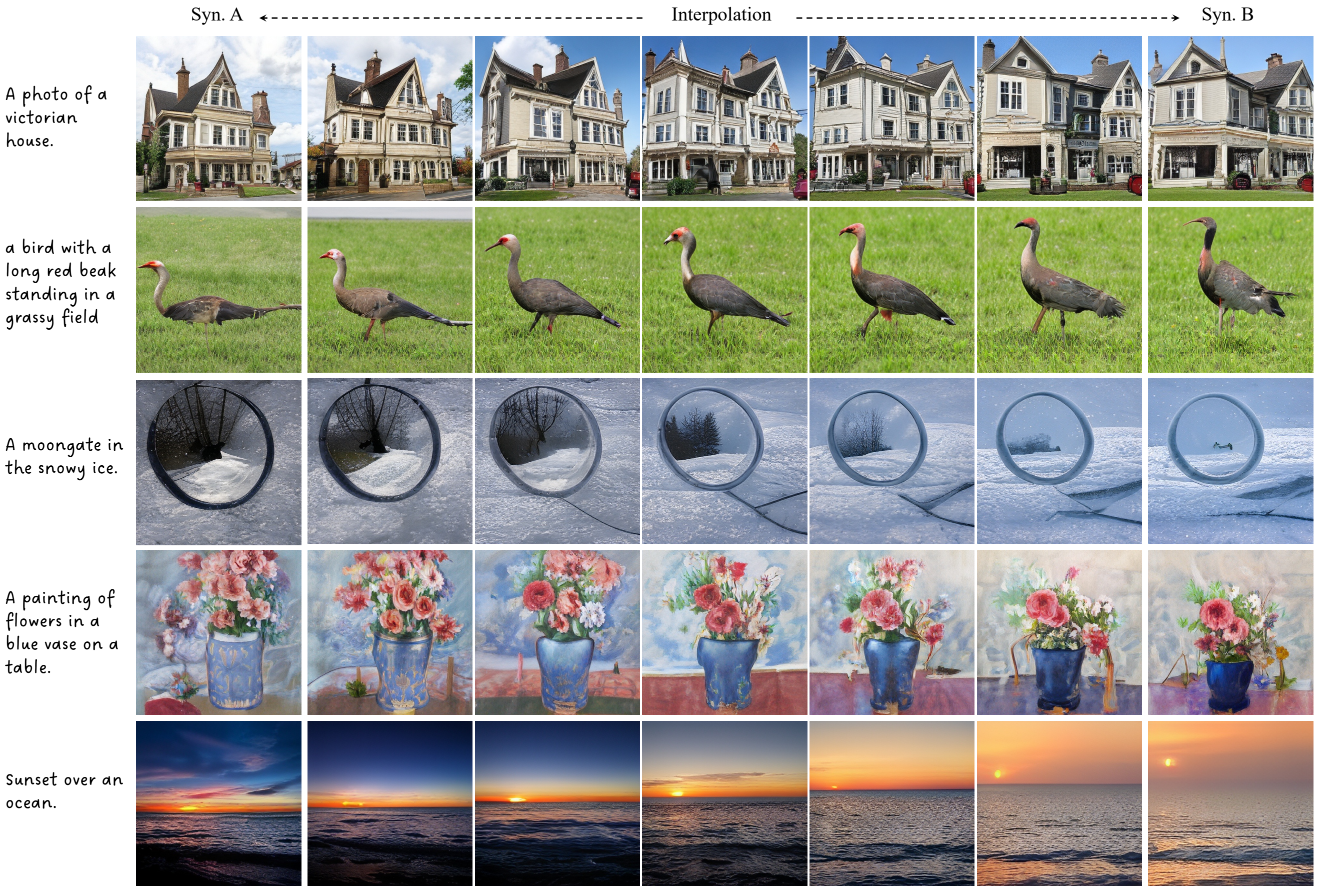

Syn. A
Interpolation
Syn. B
A photo of a victorian house.
a bird with a long red beak standing in a grassy field
A moongate in the snowy ice.
A painting of flowers in a blue vase on a table.
Sunset over an ocean.

Figure S8. **Synthesized results through latent code interpolation**, where we fix the text condition, **c**, and interpolate the latent codes within the disentangled latent space, $\mathcal{W}$. Aurora enables smooth interpolation between latent code regarding the generated content.