# Interpolating between Images with Diffusion Models

**Clinton Wang and Polina Golland**
MIT CSAIL

## Abstract

One little-explored frontier of image generation and editing is the task of interpolating between two input images, a feature missing from all currently deployed image generation pipelines. We argue that such a feature can expand the creative applications of such models, and propose a method for zero-shot interpolation using latent diffusion models. We apply interpolation in the latent space at a sequence of decreasing noise levels, then perform denoising conditioned on interpolated text embeddings derived from textual inversion and (optionally) subject poses derived from OpenPose. For greater consistency, or to specify additional criteria, we can generate several candidates and use CLIP to select the highest quality image. We obtain convincing interpolations across diverse subject poses, image styles, and image content, and show that standard quantitative metrics such as FID are insufficient to measure the quality of an interpolation. Code and data are available at https://clintonjwang.github.io/interpolation.

## 1. Introduction

Image editing has long been a central topic in computer vision and generative modeling. Advances in generative models have enabled increasingly sophisticated techniques for controlled editing of real images (Kawar et al., 2022; Zhang & Agrawala, 2023; Mokady et al., 2022), with many of the latest developments emerging from denoising diffusion models (Ho et al., 2020; Song et al., 2022; Rombach et al., 2022; Ramesh et al., 2022; Saharia et al., 2022). But to our knowledge, no techniques have been demonstrated to date for generating high quality interpolations between real images that differ in style and/or content.

Current image interpolation techniques operate in limited contexts. Interpolation between generated images has been used to study the characteristics of the latent space in generative adversarial networks (Karras et al., 2019; 2020), but such interpolations are difficult to extend to arbitrary real images as such models only effectively represent a subset of the image manifold (e.g., photorealistic human faces) and poorly reconstruct most real images (Xia et al., 2022). Video interpolation techniques are not designed to smoothly interpolate between images that differ in style; style transfer techniques are not designed to simultaneously transfer style and content gradually over many frames. We argue that the task of interpolating images with large differences in appearance, though rarely observed in the real world and hence difficult to evaluate, will enable many creative applications in art, media and design.

We introduce a method for using pre-trained latent diffusion models to generate high-quality interpolations between images from a wide range of domains and layouts (Fig. 1), optionally guided by pose estimation and CLIP scoring. Our pipeline is readily deployable as it offers significant user control via text conditioning, noise scheduling, and the option to manually select among generated candidates, while requiring little to no hyperparameter tuning between different pairs of input images. We compare various interpolation schemes and present qualitative results for a diverse set of image pairs. We plan to deploy this tool as an add-on to the existing Stable Diffusion (Rombach et al., 2022) pipeline.

## 2. Related Work

**Image editing with latent diffusion models** Denoising diffusion models (Ho et al., 2020) and latent diffusion models (Rombach et al., 2022) are powerful models for text-conditioned image generation across a wide range of domains and styles. They have become popular for their highly photorealistic outputs, degree of control offered via detailed text prompts, and ability to generalize to out-of-distribution prompts (Ramesh et al., 2022; Saharia et al., 2022). Follow-up research continued to expand their capabilities, including numerous techniques for editing real images (Kawar et al., 2022; Brooks et al., 2023; Mokady et al., 2022) and pro-

This work was was supported in part by NIH NIBIB NAC P41EB015902, Wistron Corporation, and a Takeda Graduate Fellowship to Clinton Wang. Correspondence to clintonw@csail.mit.edu.

*Workshop on Challenges in Deployable Generative AI at International Conference on Machine Learning (ICML)*, Honolulu, Hawaii, USA. 2023. Copyright 2023 by the author(s).

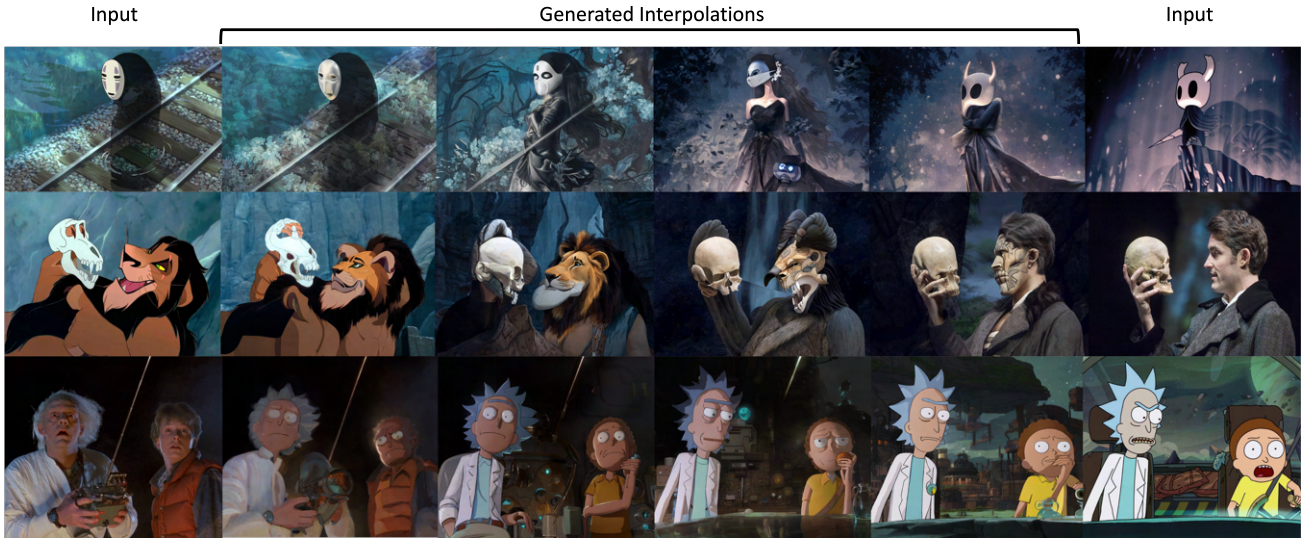


Figure 1: **Interpolations of real images.** By conditioning a pre-trained latent diffusion model on various attributes, we can interpolate pairs of images with diverse styles, layouts, and subjects.

viding new types of conditioning mechanisms (Zhang & Agrawala, 2023).

Perhaps the most sophisticated techniques for traversing latent space have been designed in the context of generative adversarial networks (GANs), where disentanglement between style and content (Karras et al., 2020), alias-free interpolations (Karras et al., 2021), and interpretable directions (Jahanian et al., 2020) have been developed. However, most such GANs with rich latent spaces exhibit poor reconstruction ability on real images, a problem referred to as GAN inversion (Xia et al., 2022). Moreover, compared to denoising diffusion models, GANs have fewer robust mechanisms for conditioning on other information such as text or pose. Latent diffusion models such as Stable Diffusion (Rombach et al., 2022) can readily produce interpolations of generated images (Lunarring, 2022), although to our knowledge this is the first work to interpolate real images in the latent space.

## 3. Preliminaries

Let $x$ be a real image. A latent diffusion model (LDM) consists of an encoder $\mathcal{E} : x \mapsto z_0$, decoder $\mathcal{D} : z_0 \mapsto \hat{x}$, and a denoising U-Net $\epsilon_\theta : (z_t; t, c_{\text{text}}, c_{\text{pose}}) \mapsto \hat{\epsilon}$. The timestep $t$ indexes a diffusion process, in which latent vectors $z_0$ derived from real images are mapped to a Gaussian distribution $z_T \sim \mathcal{N}(0, I)$ by composing small amounts of i.i.d. noise at each step. Each noisy latent vector $z_t$ can be related to the original input as $z_t = \alpha_t z_0 + \sigma_t \epsilon$, $\epsilon \sim \mathcal{N}(0, I)$, for parameters $\alpha_t$ and $\sigma_t$. The role of the denoising U-Net is to estimate $\epsilon$ (Ho et al., 2020). An LDM performs gradual denoising over several iterations, producing high quality outputs that faithfully incorporate conditioning information. $c_{\text{text}}$ is text that describes the desired image (optionally including a negative prompt), and $c_{\text{pose}}$ represents an optional conditioning pose for human or anthropomorphic subjects. The mechanics of text conditioning is described in (Rombach et al., 2022), and pose conditioning is described in (Zhang & Agrawala, 2023).

## 4. Real Image Interpolation

### 4.1. Latent interpolation

Our general strategy for generating sequences of interpolations is to iteratively interpolate pairs of images, starting with the two given input images. For each pair of parent images, we add shared noise to their latent vectors, interpolate them, then denoise the result to generate an intermediate image. The amount of noise to add to the parent latent vectors should be small if the parents are close to each other in the sequence, to encourage smooth interpolations. If the parents are far apart, the amount of noise should be larger to allow the LDM to explore nearby trajectories in latent space that have higher probability and better match other conditioning information.

Concretely, we specify a sequence of increasing timesteps $\mathcal{T} = (t_1, \ldots, t_K)$, and assign parent images using the following branching structure: images 0 and $N$ (the input images) are diffused to timestep $t_K$ and averaged to generate image $\frac{N}{2}$, images 0 and $\frac{N}{2}$ are diffused to timestep $t_{K-1}$ generate image $\frac{N}{4}$, images $\frac{N}{2}$ and $N$ are also diffused to timestep $t_{K-1}$ to generate image $\frac{3N}{4}$, and so on. By adding noise separately to each pair of parent images, this scheme encourages images to be close to their parents, but

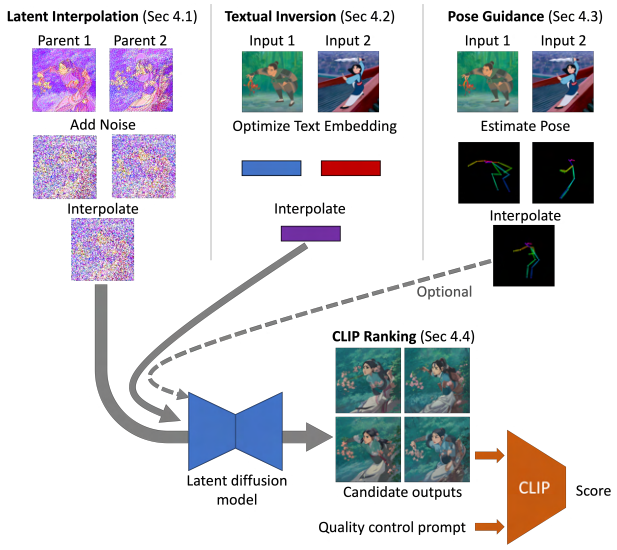


Figure 2: **Our pipeline.** To generate a new frame, we interpolate the noisy latent images of two existing frames (Section 4.1). Text prompts and (if applicable) poses are extracted from the original input images, and interpolated to provide to the denoiser as conditioning inputs (Section 4.2 and 4.3). This process can be repeated for different noise vectors to generate multiple candidates. The best candidate is selected by computing its CLIP similarity to a prompt describing desired characteristics (Section 4.4).

disentangles sibling images.

**Interpolation type** We use spherical linear interpolations (*slerp*) for latent space and text embedding interpolations, and linear interpolations for pose interpolations. Empirically, the difference between *slerp* and linear interpolation appears to be fairly mild.

**Noise schedule** We perform DDIM sampling (Song et al., 2022), and find that the LDM's quality is more consistent when the diffusion process is partitioned into at least 200 timesteps, and noticeably degrades at coarser schedules. Empirically, latent vectors denoised with less than 25% of the schedule often resemble an alpha composite of their parent images, while images generated with more than 65% of the schedule can deviate significantly from their parent images. For each interpolation we choose a linear noise schedule within this range, depending on the amount of variation desired in the output. Our approach is compatible with various stochastic samplers (Karras et al., 2022) which seem to yield comparable results.

### 4.2. Textual inversion

Pre-trained latent diffusion models are heavily dependent on text conditioning to yield high quality outputs of a particular style. Given an initial text prompt describing the overall content and/or style of each image, we can adapt its embedding more specifically to the image by applying textual inversion. In particular, we encode the text prompt as usual, then fine-tune the prompt embedding to minimize the error of the LDM on denoising the latent vector at random noise levels when conditioned on this embedding. Specifically, we perform 100-500 iterations of gradient descent with the loss $\mathcal{L}(c_{\text{text}}) = \|\hat{\epsilon}_\theta(\alpha_t z_0 + \sigma_t \epsilon; t, c_{\text{text}}) - \epsilon\|$ and a learning rate of $10^{-4}$. The number of iterations can be increased for images with complicated layouts or styles which are harder to represent with a text prompt.

In this paper we specify the same initial prompt for both input images, although one can also substitute a captioning model for a fully automated approach. Both positive and negative text prompts are used and optimized, and we share the negative prompt for each pair of images. Since our task does not require a custom token, we choose to optimize the entire text embedding.

### 4.3. Pose guidance

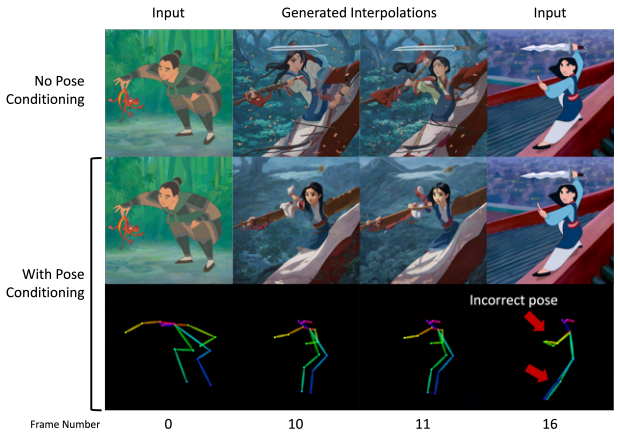


Figure 3: **Pose conditioning** mitigates the occurrence of abrupt pose changes between adjacent frames, even when the predicted pose is incorrect.

If the subject's pose differs significantly between the two images, image interpolation is challenging and often results in anatomical errors such as multiple limbs and faces. We obtain more plausible transitions between subjects in different poses by incorporating pose conditioning information in the LDM. We obtain poses of the input images using OpenPose (Cao et al., 2019), with the assistance of style transfer for cartoons or non-human subjects (see Fig. 4). We then linearly interpolate all shared keypoint positions from the two images to obtain intermediate poses for each image. The resulting pose is provided to the LDM using ControlNet (Zhang & Agrawala, 2023), a powerful method for conditioning on arbitrary image-like inputs. Interestingly, we observe that *even when the wrong pose is predicted* for input images, conditioning on pose still yields superior in-

terpolations as it prevents abrupt pose changes (see Fig. 3).

Figure 4: When the input image is stylized, OpenPose fails to produce a pose with high confidence. Thus we first perform image-to-image translation using our LDM, to convert the input image to the style of a photograph before applying OpenPose. It often still succeeds even when the translated image is of low quality.

### 4.4. CLIP ranking

LDMs can yield outputs of widely varying quality and characteristics with different random seeds. This problem is compounded in real image interpolation since a single bad generated image compromises the quality of all other images derived from it. Thus when quality is more important than speed, multiple candidates can be generated with different random seeds, then ranked with CLIP (Radford et al., 2021). We repeat each forward diffusion step with different noise vectors, denoise each of the interpolated latent vectors, then measure the CLIP similarity of the decoded image with specified positive and negative prompts (e.g., positive: "high quality, detailed, 2D", negative: "blurry, distorted, 3D render"). The image with the highest value of positive similarity minus negative similarity is kept. In applications requiring an even higher degree of control and quality, this pipeline can be changed into an interactive mode where users can manually select desired interpolations or even specify a new prompt or pose for a particular image.

## 5. Experiments

We analyze the effect of various design choices when applying Stable Diffusion v2.1 (Rombach et al., 2022) with pose-conditioned ControlNet on a curated set of 26 pairs of images spanning diverse domains (see Fig. A.1-A.3 for more examples). They include photographs, logos and user interfaces, artwork, ads and posters, cartoons, and video games.

### 5.1. Latent Interpolation

We compare our approach for latent vector interpolation against several baselines: interpolating without denoising (interpolate only), interpolating between noisy versions of the input vectors (interpolate-denoise), interpolating partially denoised versions of generated latents (denoise-interpolate-denoise), and denoise-interpolate-denoise with no shared noise added to the input latents.

**Interpolate only** The naive interpolation scheme simply interpolates the clean latent codes of the input images without performing any diffusion. We set $z_0^0 := \mathcal{E}(x^0)$, $z_0^N := \mathcal{E}(x^N)$, and all images are generated via $z_0^i = \texttt{slerp}(z_0^0, z_0^N, i/N)$, $x^i := \mathcal{D}(z_0^i)$. This approach completely fails to generate reasonable images as the denoised latent space in LDMs is not well-structured.

**Interpolate-denoise** We choose a sequence of increasing timesteps $\mathcal{T} = (0, \ldots, T)$ and create sequences of corresponding noisy latents $\{z_t^0\}_{t\in\mathcal{T}}, \{z_t^N\}_{t\in\mathcal{T}}$, such that:

$$z_t^0 = \alpha_t z_{t-1}^0 + \beta_t \epsilon_t, \tag{1}$$

$$z_t^N = \alpha_t z_{t-1}^N + \beta_t \epsilon_t, \tag{2}$$

where $\epsilon_t \sim \mathcal{N}(0, I)$ is shared for both images, and $z_0^0, z_0^N$ are obtained as before. Each intermediate image is assigned a particular timestep $t := \texttt{frame_schedule}(i)$ to generate its interpolated latent code: $z_t^i := \texttt{slerp}(z_t^0, z_t^N, i/N)$. $\texttt{frame_schedule}$ is a function that monotonically decreases as its input approaches 0 or $N$, to support smooth interpolation close to the input images. We then perform denoising with the LDM: $z_0^i := \mu_\theta(z_t^i, t)$ and use the decoder to produce the image.

**Denoise-interpolate-denoise** If we rely on $\{z_t^0\}$ and $\{z_t^N\}$ to generate all intermediate latents, adjacent images at high noise levels may diverge significantly during the denoising process. Instead, we can interpolate images in a branching pattern as follows: we first generate $z_{t_1}^{N/2}$ as an interpolation of $z_{t_1}^0$ and $z_{t_1}^N$, denoise it to time $t_2$, then generate $z_{t_2}^{N/4}$ as an interpolation of $z_{t_2}^0$ and $z_{t_2}^{N/2}$, and generate $z_{t_2}^{3N/4}$ similarly. These two new latents can be denoised to time $t_3$, and so on. The branching factor can be modified at any level so the total number of frames does not need to be a power of 2. This interpolation scheme is similar to latent blending (Lunarring, 2022).

Qualitatively we found that the most convincing and interesting interpolations were achieved by our method (Fig. 5). Other interpolation schemes either fully couple the noise between all frames, which results in less creative outputs that resemble alpha blending rather than a semantic transformation, or do not perform any noise coupling, which can result in abrupt changes between adjacent frames. Interestingly this phenomenon is not captured by distributional metrics such as Fréchet inception distance (FID) (Heusel et al., 2018) or smoothness metrics such as perceptual path length (PPL) (Karras et al., 2020) (see Table 1). We computed the FID between the distribution of input images and

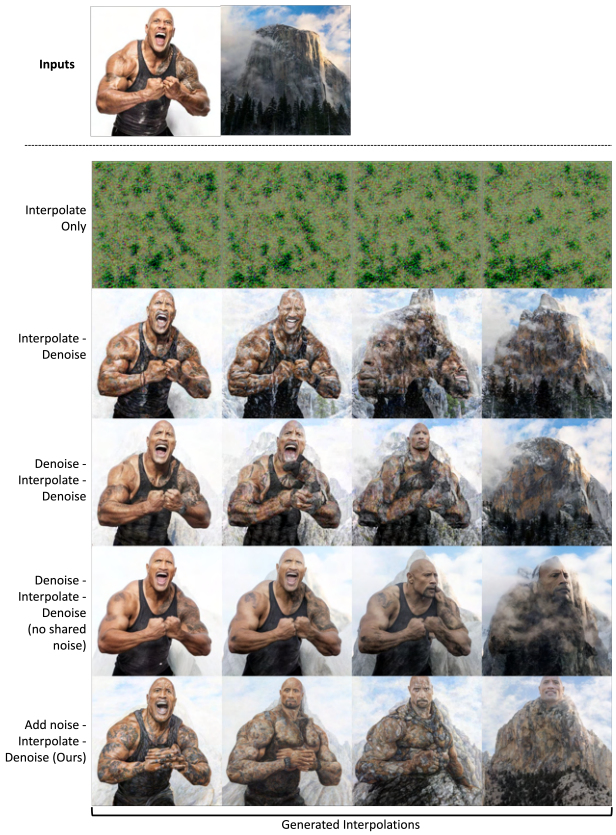


Figure 5: **Comparison of different interpolation schemes.** We add noise to the latents derived from our input images, and denoise the interpolated latents to generate output frames. This approach performs a more convincing semantic transformation from a human to a mountain compared to other approaches which instead resemble alpha blending.

distribution of output images (two random frames sampled from every interpolation) as a proxy for the degree to which output images lie on the image manifold. We compute PPL as the sum of Inception v3 distances between adjacent images in 17-frame sequences, to measure the smoothness of the interpolations and the degree to which the interpolation adheres to the appearance of the input images. We find that both these metrics favor interpolations that resemble simple alpha composites rather than more creative interpolations, as the latter deviate more in feature statistics from the original set of images, even if they would be preferred by users. Thus current metrics are insufficient to capture the effectiveness of an interpolation, an open question that we hope to tackle in future work.

### 5.2. Extensions

**Interpolation schedule** In all examples presented in this paper, we use a uniform interpolation schedule. But evenly spaced interpolations in the latent space do not necessarily translate to a constant rate of perceptual changes in the image space. While coloration and brightness seem to evolve at a constant rate between frames, we observe that stylistic changes can occur very rapidly close to the input images (for example, the transition from real to cartoon eyes in the third row of Fig. 1). Thus in applications where the user would like to control the rate of particular changes, it can be helpful to specify a non-uniform interpolation schedule.

Table 1: **Quantitative comparison.** Fréchet inception distance (FID) between input images and their interpolations, and perceptual path length (PPL, mean±std) of each interpolation in Inception v3 feature space.

| Interpolation Scheme | FID | PPL |
|---|---|---|
| Interpolate only | 436 | 56±8 |
| Interpolate-denoise | 179 | 172±32 |
| Denoise-interpolate-denoise (DID) | 169 | 144±26 |
| DID w/o shared noise | 199 | 133±22 |
| Add noise-interpolate-denoise (ours) | 214 | 193±27 |

**Adding motion** Interpolation can be combined with affine transforms of the image in order to create the illusion of 2D or 3D motion (Fig. 6). Before interpolating each pair of images, we can warp the latent of one of the images to achieve the desired transform.

Figure 6: Our pipeline can be combined with affine transforms such as zooming on a point.

## 6. Conclusion

We introduced a new method for real image interpolation that can generate imaginative, high-quality sequences connecting images with different styles, content and poses. This technique is quite general, and can be readily integrated with many other methods in video and image generation such as specifying intermediate prompts, and conditioning on other inputs such as segmentations or bounding boxes.

**Limitations** Our method can fail to interpolate pairs of images that have large differences in style and layouts. In Fig. A.4, we illustrate examples where the model cannot detect and interpolate the pose of the subject (top), fails to understand the semantic mapping between objects in the frames (middle), and struggles to produce convincing interpolations between very different styles (bottom). We also find that the model occasionally inserts spurious text, and can confuse body parts even given pose guidance.

## References


Brooks, T., Holynski, A., and Efros, A. A. Instructpix2pix: Learning to follow image editing instructions, 2023.

Cao, Z., Hidalgo Martinez, G., Simon, T., Wei, S., and Sheikh, Y. A. Openpose: Realtime multi-person 2d pose estimation using part affinity fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., and Hochreiter, S. Gans trained by a two time-scale update rule converge to a local nash equilibrium, 2018.

Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models, 2020.

Jahanian, A., Chai, L., and Isola, P. On the "steerability" of generative adversarial networks, 2020.

Karras, T., Laine, S., and Aila, T. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4401–4410, 2019.

Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., and Aila, T. Analyzing and improving the image quality of stylegan, 2020.

Karras, T., Aittala, M., Laine, S., Härkönen, E., Hellsten, J., Lehtinen, J., and Aila, T. Alias-free generative adversarial networks, 2021.

Karras, T., Aittala, M., Aila, T., and Laine, S. Elucidating the design space of diffusion-based generative models, 2022.

Kawar, B., Zada, S., Lang, O., Tov, O., Chang, H., Dekel, T., Mosseri, I., and Irani, M. Imagic: Text-based real image editing with diffusion models. *arXiv preprint arXiv:2210.09276*, 2022.

Lunarring. Latent blending. https://github.com/lunarring/latentblending, 2022.

Mokady, R., Hertz, A., Aberman, K., Pritch, Y., and Cohen-Or, D. Null-text inversion for editing real images using guided diffusion models, 2022.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., and Sutskever, I. Learning transferable visual models from natural language supervision, 2021.

Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., and Chen, M. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022.

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. High-resolution image synthesis with latent diffusion models, 2022.

Saharia, C., Chan, W., Saxena, S., Li, L., Whang, J., Denton, E., Ghasemipour, S. K. S., Ayan, B. K., Mahdavi, S. S., Lopes, R. G., Salimans, T., Ho, J., Fleet, D. J., and Norouzi, M. Photorealistic text-to-image diffusion models with deep language understanding, 2022.

Song, J., Meng, C., and Ermon, S. Denoising diffusion implicit models, 2022.

Xia, W., Zhang, Y., Yang, Y., Xue, J.-H., Zhou, B., and Yang, M.-H. Gan inversion: A survey, 2022.

Zhang, L. and Agrawala, M. Adding conditional control to text-to-image diffusion models, 2023.

## A. Additional Figures

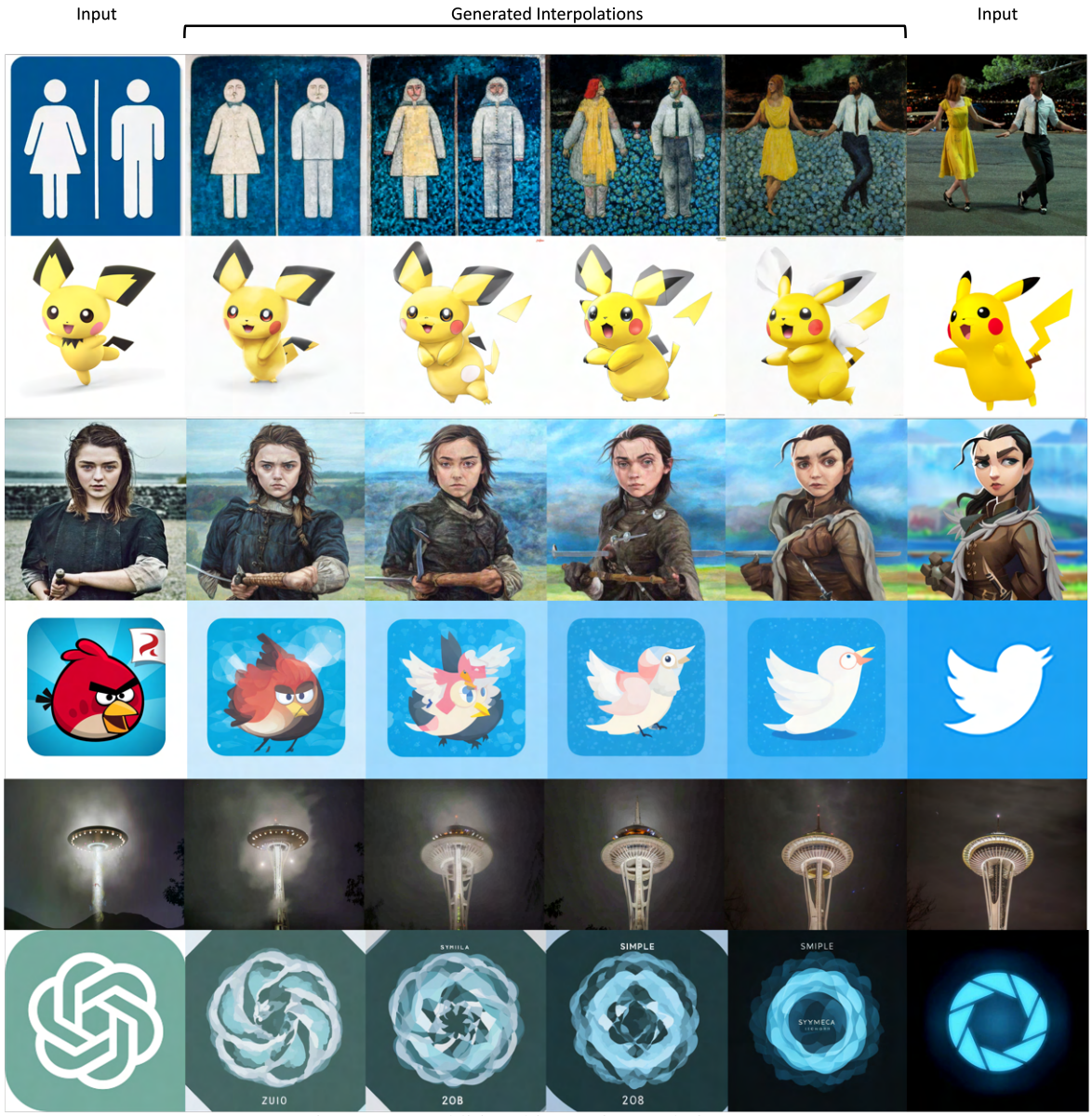


Figure A.1: Additional image interpolations (1/3).

Figure A.2: Additional image interpolations (2/3).

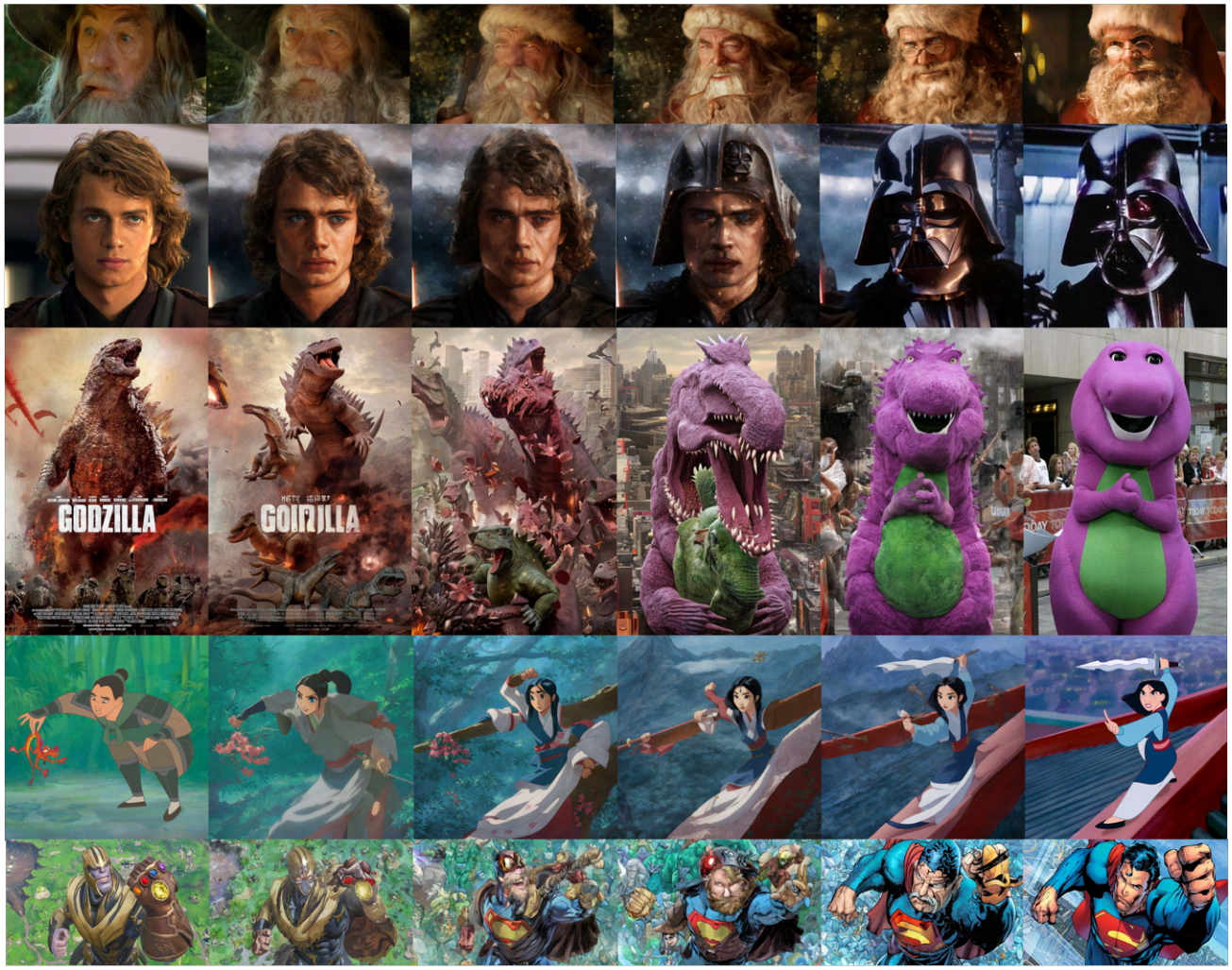


Figure A.3: Additional image interpolations (3/3).

Figure A.4: Failure cases. Our approach is still limited in its ability to bridge large gaps in style, semantics and/or layout.