

Probability Density Geodesics in Image Diffusion Latent Space

Qingtao Yu¹, Jaskirat Singh¹, Zhaoyuan Yang², Peter Henry Tu²,
 Jing Zhang¹, Hongdong Li¹, Richard Hartley¹, Dylan Campbell¹
¹Australian National University ²GE Research

{terry.yu, jaskirat.singh, jing.zhang, hongdong.li, richard.hartley, dylan.campbell}@anu.edu.au
 {peter.tu, zhaoyuan.yang}@ge.com

Abstract

Diffusion models indirectly estimate the probability density over a data space, which can be used to study its structure. In this work, we show that geodesics can be computed in diffusion latent space, where the norm induced by the spatially-varying inner product is inversely proportional to the probability density. In this formulation, a path that traverses a high density (that is, probable) region of image latent space is shorter than the equivalent path through a low density region. We present algorithms for solving the associated initial and boundary value problems and show how to compute the probability density along the path and the geodesic distance between two points. Using these techniques, we analyze how closely video clips approximate geodesics in a pre-trained image diffusion space. Finally, we demonstrate how these techniques can be applied to training-free image sequence interpolation and extrapolation, given a pre-trained image diffusion model.

1. Introduction

When trained over a data space, a diffusion model [43–46] can tell us the direction in which a data point of that space should move in order to increase its likelihood. That is, it learns a vector field corresponding to the gradient of the log-probability of the data (the Stein score function [47]). This is sufficient information to define a Riemannian manifold in this space, where the norm induced by a spatially-varying inner product is chosen to be inversely proportional to the probability density. This has the effect that geodesics—loosely, shortest paths—on this manifold ‘prefer’ to traverse high-density regions, which can be thought of as shortcuts in the space. In the context of image latent diffusion models [37], these regions correspond to the latent vectors of probable images, while low-density regions correspond to unrealistic image latents. By computing geodesics in this space, we can find shortest paths between images and study the structure of the learned space.

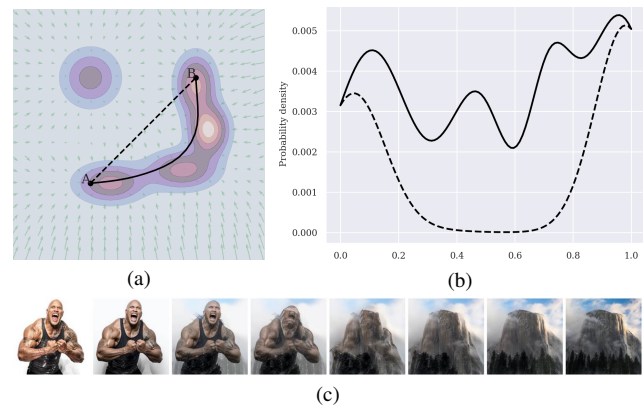


Figure 1. Given a probability density and initial or boundary conditions (here, the position of two points A and B), geodesics can be computed in this space. If the norm is chosen to be inversely proportional to the probability density, these geodesics preferentially traverse high density regions of the space. For an image data space, these correspond to plausible, realistic images according to the probability density, such as that learned by an image diffusion model. Here, we show the outputs of our boundary value problem (BVP) solver that computes a geodesic between endpoints A and B on a toy 2D example. (a) The geodesic and straight-line trajectories between A and B, given the underlying visualized probability density field (contours) and its gradient (arrows). (b) Probability density curves for both trajectories, showing that the straight-line path drops to zero probability very rapidly whereas the geodesic remains in higher probability regions. (c) Images corresponding to points along a geodesic in Stable Diffusion [37] latent space, given the left and right endpoints, computed using our BVP solver.

In this work, we outline the relevant theory needed for computing probability density geodesics in diffusion latent space and present algorithms for solving the associated initial and boundary value problems. A 2D toy example of the boundary value problem (BVP) is presented in Fig. 1, alongside image outputs from the BVP solved in the 16384-D latent space of Stable Diffusion [37]. We address several challenges associated with computing

geodesics in this space: (i) accurately estimating the score function, especially in low-density regions; (ii) handling (potentially spatially-varying) conditional probability densities; and (iii) solving initial and boundary value problems efficiently in an extremely high-dimensional space.

We also describe how to compute several useful quantities for analysis: the relative probability density along the path, the geodesic distance between two points, and the geodesic gradient norm. Using these techniques, we analyze how closely video clips approximate geodesics in a pre-trained image diffusion space. This work is part exploratory (e.g., what image sequences correspond to geodesics?) and part applied (e.g., how can these techniques be used to solve image sequence-based tasks?). For the latter, we evaluate the performance of these techniques on training-free image sequence interpolation and extrapolation tasks, given a pre-trained image diffusion model. Our contributions are:

1. characterizing probability density geodesics in diffusion latent space;
2. algorithms for solving initial and boundary value problems that address challenges specific to conditional diffusion models; and
3. an application to image sequence interpolation and extrapolation tasks.

We evaluate our training-free approach on five datasets with respect to other interpolation methods, including those that fine-tune the underlying diffusion model, achieving state-of-the-art results.

2. Related Work

Text-to-image diffusion models. Recently, diffusion models [8, 19] have gained significant attention in deep generative modeling. Given specific input conditions, these models generate realistic data by progressively denoising Gaussian noise through a noise prediction network, resulting in outputs that closely resemble real data distributions based on the provided conditions. With the availability of large-scale image-text pairs for training, diffusion models have achieved remarkable performance in the text-to-image generation task [34, 37, 40]. Due to the flexibility of textual prompting, text-to-image diffusion models have been widely applied to image editing [2, 14, 15, 26], personalization [9, 12, 29, 38], stylization [16, 48, 49], etc. In this work, we use a pretrained Stable Diffusion [37] model with which to explore probability density geodesics for video analysis, image interpolation and extrapolation.

Image interpolation. Image interpolation, also known as image morphing, refers to the process of generating a smooth transition between two images by creating intermediate images [51, 56]. Traditional methods rely on pixel-based transformations, such as mesh morphing or defining

a blending function [30, 41, 55]. However, these methods typically require pre-labeled corresponding points and human involvement. Some attempts have been made to automate this process [21, 32], but they generally work only on image pairs generated in controlled environments. With advances in generative models, Generative Adversarial Networks (GANs) [10] have been applied to the image interpolation task [28, 33, 42] and achieve moderate success. However, GAN-based image interpolation frameworks do not perform well in open-world settings. Specifically, input image pairs are required to be in-distribution with the training data of GAN models, which limits the application scenarios for image interpolation. With recent progress in text-to-image diffusion models, several works [11, 13, 50, 52, 53] have shown that these models achieve impressive results in image interpolation. Given any image pair, through textual embedding inversion and embedding interpolation, relatively smooth intermediate images can be obtained. In this work, we also employ a diffusion model; however, compared with prior works [11, 52, 53], our method is training-free and can be applied to image extrapolation. Furthermore, our approach offers a theoretical framework with which to understand these tasks.

3. Probability Density Geodesics

In this section, we will outline the theory of probability density geodesics, followed by the algorithmic development. The following section will apply the theory to a diffusion latent space and address some design decisions particular to that context. We aim to show that the structure of a data space can be studied by computing geodesics with respect to the probability distribution of the data, which can be obtained by training a diffusion model in that space.

A geodesic is the (locally) shortest path between two points on a Riemannian manifold with a constant speed parameterization and can be obtained by minimizing the length of a curve between them using the calculus of variations. In the following, time derivatives are denoted \dot{x} and second derivatives as \ddot{x} .

Equation for path length. Let $\gamma : [0, 1] \rightarrow \mathbb{R}^n$ be a path such that $\gamma(0) = x_0$ and $\gamma(1) = x_1$, and $S : \{\gamma_i\} \rightarrow \mathbb{R}$ be the action functional on the set of such paths, defined by

$$S[\gamma] = \int_a^b L(t, \gamma(t), \dot{\gamma}(t)) dt, \quad (1)$$

where L is the Lagrangian given by

$$L(t, \gamma, \dot{\gamma}) = \sqrt{\langle \dot{\gamma}, \dot{\gamma} \rangle_{K(\gamma)}} \text{ with } K(\gamma) = p(\gamma)^{-2} I. \quad (2)$$

Then $S[\gamma]$ is the path length for path γ . The norm $\|x\| = \sqrt{\langle x, x \rangle_M}$ induced by the inner product $\langle x, y \rangle_M = x^T M y$ is inversely proportional to the probability density with

function $p : \mathbb{R}^n \rightarrow \mathbb{R}_+$. The interpretation here is that paths that pass through high density regions are ‘shorter’ than paths through low density regions. Note that this reduces to the standard formulation for the path length of a curve when the probability distribution is uniform.

Euler–Lagrange equations. Given this definition, a path γ is a stationary point of S if and only if it satisfies the Euler–Lagrange equations, viz.,

$$\frac{\partial}{\partial \gamma} L(t, \gamma(t), \dot{\gamma}(t)) - \frac{d}{dt} \frac{\partial}{\partial \dot{\gamma}} L(t, \gamma(t), \dot{\gamma}(t)) = 0. \quad (3)$$

Using $\nabla \log p(\gamma) = \frac{1}{p} \frac{dp}{d\gamma}^\top$ and a constant speed parameterization of the path, for our Lagrangian we obtain

$$\ddot{\gamma} + \|\dot{\gamma}\|^2 \left(I - \hat{\gamma} \hat{\gamma}^\top \right) \nabla \log p(\gamma) = 0, \quad (4)$$

where the unit velocity is given by $\hat{\gamma} = \dot{\gamma} / \|\dot{\gamma}\|$. That is, we obtain a nonlinear second-order ordinary differential equation (ODE) that expresses the relationship between the scaled acceleration and the gradient of the log probability. It is important to observe here that $\nabla \log p(\gamma)$ is the Stein score function [20, 47], the exact quantity that is estimated by a score-based diffusion model. The full derivation is presented in the appendix.

Functional derivative. This second-order ODE expresses the relationship at optimality, i.e., given an initial position and velocity we can obtain the associated optimal path. However, we can also derive the functional derivative $\frac{\delta S}{\delta \gamma}$ of the path length functional S by approximating the curve by a polygonal line with n segments, as n grows arbitrarily large. We obtain, for any γ with a constant speed parameterization,

$$\frac{\delta S}{\delta \gamma} = \frac{-1}{p(\gamma) \|\dot{\gamma}\|} \left(\left(I - \hat{\gamma} \hat{\gamma}^\top \right) \nabla \log p(\gamma) + \frac{\ddot{\gamma}}{\|\dot{\gamma}\|^2} \right). \quad (5)$$

Probability density along the path. While estimating the absolute probability of any data point is challenging, the probability relative to the starting point is convenient to compute. For a conservative vector field $\nabla \log p$ (the gradient of a scalar field), line integrals are path independent.¹ Therefore, the relative log-probability between two points $\gamma(a)$ and $\gamma(t)$, along *any* path γ connecting them, is

$$\log \tilde{p}_0(\gamma(t)) := \log p(\gamma(t)) - \log p(\gamma(0)) \quad (6)$$

$$= \int_0^t \dot{\gamma}(\tau)^\top \nabla \log p(\gamma(\tau)) d\tau. \quad (7)$$

Let $f(x) = \dot{\gamma}(x)^\top \nabla \log p(\gamma(x))$ and let $t_i = i/n$ for $0 \leq i \leq n$ be $n + 1$ equally spaced samples along the path.

¹This assumption is not strictly true for standard diffusion models, but is a good approximation [3].

Then by the trapezoidal rule, the relative probability at these points along the path can be approximated as

$$\log \tilde{p}_0(\gamma(t_i)) \approx \frac{1}{n} \left(\frac{f(t_0)}{2} + \sum_{k=1}^{i-1} f(t_k) + \frac{f(t_i)}{2} \right). \quad (8)$$

Higher-order approaches, such as Simpson’s rule, may also be used for this approximation. From this, we may compute the relative probability $\tilde{p}_0(\gamma(t)) = p(\gamma(t_i))/p(\gamma(0))$ at every point $\gamma(t_i)$ along the curve.

Geodesic distance. Given the probabilities computed in the previous section, it is trivial to estimate the geodesic distance. Let $t_i = i/n$ for $0 \leq i \leq n$ be $n + 1$ equally spaced samples along the geodesic γ , then the relative geodesic distance is given by

$$\begin{aligned} \tilde{d}_a(b) &= \int_a^b \frac{\|\dot{\gamma}(t)\|}{\tilde{p}_a(\gamma(t))} dt \\ &\approx \frac{1}{n} \left(\frac{\|\dot{\gamma}(a)\|}{2} + \sum_{i=1}^{n-1} \frac{\|\dot{\gamma}(t_i)\|}{\tilde{p}_a(\gamma(t_i))} + \frac{\|\dot{\gamma}(b)\|}{2\tilde{p}_a(\gamma(t_n))} \right) \end{aligned} \quad (9)$$

and the absolute geodesic distance is given by $d(a, b) = \tilde{d}_a(b)/p(\gamma(a))$.

4. Geodesics in Diffusion Latent Space

In this section, we show how to compute geodesics in the latent space of a pre-trained Stable Diffusion [37] model. An element $x \in \mathbb{R}^{64 \times 64 \times 4}$ of this latent space can be obtained from an image \mathcal{I} via an encoder $\mathcal{E}(\mathcal{I})$; a decoder $\mathcal{D}(x)$ performs the inverse mapping. A forward process incrementally corrupts these elements with Gaussian noise, up to timestep $\tau = T$, resulting in a noise element $x_\tau \sim \mathcal{N}(0, I)$. To distinguish between the time dependence of the diffusion process and that of the path formulation in Sec. 3, we use τ instead of t to denote the diffusion time-step throughout the rest of the paper. The diffusion model is trained on a very large dataset of images to denoise elements of this latent space at all timesteps. It is parameterized by a neural network $\epsilon_\theta(x_\tau; z)$ that predicts the noise ϵ that was used to produce noisy sample x_τ from clean sample x_0 [19]. This is proportional to the score function $\nabla_{x_\tau} \log p_\tau(x_\tau)$ of the smoothed (noised) density p_τ [19]. Finally, our method makes use of the deterministic DDIM forward and backward processes (DDIM-F, DDIM-B) with noise schedule parameter $\bar{\alpha}_\tau$, known as inversion [44], defined by

$$\frac{d}{d\tau} \left(\frac{x_\tau}{\sqrt{\bar{\alpha}_\tau}} \right) = \frac{d}{d\tau} \left(\sqrt{\frac{1 - \bar{\alpha}_\tau}{\bar{\alpha}_\tau}} \right) \epsilon_\theta(x_\tau; z). \quad (10)$$

Working with diffusion models introduces several challenges, which we address in this section. In particular, (i) how to handle (potentially varying) conditional probability

densities; (ii) how to accurately estimate the score function $\nabla \log p(x)$, especially in low density regions; and (iii) how to solve the initial and boundary value problems efficiently in an extremely high-dimensional space (e.g., \mathbb{R}^{16384}).

4.1. Conditional probability density geodesics

Most image diffusion models are conditional models, trained to denoise an image conditioned on a signal z such as a CLIP-encoded text prompt [36, 37, 40]. Hence, we extend the formulation from Sec. 3 to consider the conditional probability density function $p(\gamma(t) | \zeta(t, \gamma(t)))$. Here, $\zeta : \mathbb{R} \times \mathbb{R}^n \rightarrow \mathbb{R}^d$ is a function over the path parameter and latent space that obtains the values of the conditioning vectors z_a and z_b at the endpoints, such that $\zeta(0, \gamma(0)) = z_0$ and $\zeta(1, \gamma(1)) = z_1$ for $(a, b) = (0, 1)$. In our experiments, we primarily consider the case where ζ is linear in time and constant in the latent space: $\zeta(t) = (1 - t)z_0 + tz_1$.

4.2. Score estimation

The score function is likely to be poorly estimated in low-density regions [45]. This occurs since the diffusion model does not have enough evidence during training to estimate the score accurately in regions of the latent space where data is scarce. However, estimating accurate directions in these regions is critical for solving boundary value problems, since the initial path to be optimized is likely to traverse low-density regions. Diffusion models apply multiple levels of noise to data elements in order to alleviate this problem. By doing so, low-density regions of the space can be populated by noised data, improving the quality of the estimated gradients at those locations. As a result, we choose to compute geodesics (and query the diffusion model) at a particular non-zero noise level, with the associated diffusion timestep τ . A side benefit of this is that the probability density becomes increasingly Gaussian as the timestep increases, which provides a useful inductive bias, as explained in the next section.

However, there is an additional complexity. For a noised training sample x_τ , the unconditional model $\epsilon_\theta(x_\tau; z = \emptyset, \tau)$ is expected to predict the noise ϵ that was used to produce this sample from clean sample x_0 . Unfortunately, in our case, we obtain vectors x_τ from an optimization procedure, initialized along a great circle between endpoints $x_{a,\tau}$ and $x_{b,\tau}$. These points are (at least initially) out-of-distribution (OOD) for the model, and we should not expect it to predict the noise well. Instead, Katzir et al. [25] observe that the prediction consists of a domain correction term δ_D and a denoising term δ_N . The residual $\delta_N - \epsilon$, as used in score distillation sampling [35], is generally non-zero and noisy, leading to an averaging (over-smoothing) effect in the optimized latents. In our optimization procedure, we are not aiming to denoise the samples, so neglect the denoising direction δ_N and instead use noise-free score distillation [25]

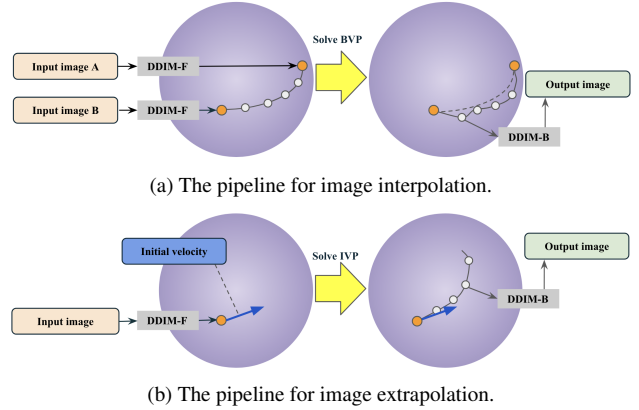


Figure 2. Pipelines for the image interpolation and extrapolation tasks, addressed by solving boundary and initial value problems in diffusion latent space.

direction $\phi(x | z, \tau)$ to approximate the gradient direction,

$$\nabla \log p(x | z, \tau) \approx \beta \phi(x | z, \tau) \quad (11)$$

$$\phi(x | z, \tau) = \mathbb{E}_{\tau' \in \mathcal{R}_\tau} w(\tau') (\sigma d(x_{\tau'} | z) - d(x_{\tau'} | z_{\text{neg}})), \quad (12)$$

where β is a scalar hyperparameter, $w(\tau)$ is a weighting function, σ is the classifier-free guidance parameter (0 if unconditional) [18], z_{neg} is a negative prompt embedding that says something about the specific OOD latent vector (see Appendix B), $d(x_\tau | z) = \epsilon_\theta(x_\tau | \emptyset) - \epsilon_\theta(x_\tau | z)$ is the direction function, and the expectation is taken over a range of timesteps $\mathcal{R}_\tau = [\tau - \Delta\tau, \tau + \Delta\tau]$.

4.3. Efficient IVP and BVP solvers

Solving the initial and boundary value problems efficiently with respect to time and memory becomes challenging in the extremely high-dimensional latent space $\mathbb{R}^{4 \times 64 \times 64}$ of Stable Diffusion [37]. For this dimensionality, standard techniques, such as collocation algorithms for solving BVPs [27], become prohibitive in time and memory. We also aim to minimize calls to the diffusion model, since this is expensive to compute. In view of these aims, we use a simple low-memory parametrization and optimization strategy.

Before outlining the algorithms, we first observe that the probability distribution becomes more Gaussian as the diffusion timestep increases [5] and that the Gaussian Annulus Theorem [1] states that a point chosen at random from a unit variance d -dimensional Gaussian distribution will be located in a small annulus around a sphere of radius \sqrt{d} with high probability. However, for finite-sized gradient descent steps, the optimizer may take the curve away from the sphere, moving into low-density regions where probability density gradients are poorly estimated. To mitigate this, we reproject the curve back onto the sphere, ensuring that optimization remains within the high-density region where gra-

Algorithm 1: BVP solver for image interpolation.

Input: start/end image $\mathcal{I}_0/\mathcal{I}_1$; text prompts p_0/p_1 ;
diffusion timestep τ ; learning rate η ;
optimizer steps n ; hyperparameter β ; VAE
encoder \mathcal{E} ; CLIP text encoder \mathcal{C}

Output: geodesic γ

```

1  $\{z_0, z_1\} \leftarrow \{\mathcal{C}(p_0), \mathcal{C}(p_1)\}$ 
2  $\{x_0, x_1\} \leftarrow \{\text{DDIM-F}(\mathcal{E}(\mathcal{I}_i), \tau, z_i)\}_{i=\{0,1\}}$ 
3  $\gamma \leftarrow \text{Interpolate}(\{(0, x_0), (1, x_1)\})$ 
4 for  $i = 1$  to  $n$  do
5    $\mathcal{T} \leftarrow \text{TimeSampler}(i)$ 
6    $\mathcal{S} \leftarrow \{(0, x_0), (1, x_1)\}$ 
7   forall  $t \in \mathcal{T}$  do
8      $x \leftarrow \gamma(t)$ ;
9      $z \leftarrow (1-t)z_0 + tz_1$ 
10     $s \leftarrow \text{Score}(x, z, \tau, \beta)$ ; % Eq. (11)
11     $g \leftarrow \text{FuncDeriv}(s, x, \dot{x}, \ddot{x})$ ; % Eq. (13)
12     $x \leftarrow \|x\|(x - \eta g) / \|x - \eta g\|$ 
13     $\mathcal{S} \leftarrow \mathcal{S} \cup \{(t, x)\}$ 
14  $\gamma \leftarrow \text{Interpolate}(\mathcal{S})$ 

```

dient estimates are more reliable, and we project the functional derivative to the tangent space of the sphere,

$$g = (I - \hat{\gamma}\hat{\gamma}^\top) \frac{\delta S}{\delta \gamma}. \quad (13)$$

BVP. Given a start position $x^{(0)}$ and an end position $x^{(k+1)}$, we parameterize the path γ with a set of control points $\{x_\tau^{(i)}\}_{i=1}^k$ connected by a spherical piecewise linear function (great circle arcs). For clarity, we notate points on this curve as x_t with the curve parameter $t \in [0, 1]$, dropping τ and the ordinal superscript. Algorithm 1 initializes the path as a great circle. It then computes the projected gradient descent update for the control points using Eq. (13), where the associated velocities $\dot{\gamma}$ and accelerations $\ddot{\gamma}$ are obtained from a natural cubic spline fit to the control points and end points, and projects the updated control points back onto the sphere. To save computation, the algorithm uses a coarse-to-fine discretization, where the number of control points $k \in [1, 3, 7, 15]$ varies as optimization progresses, using a bisection strategy.

IVP. Algorithm 2 takes an image \mathcal{I}_0 and its corresponding text description p_0 , and extrapolates how the geodesic evolves given an initial velocity formulated from the target prompt p_1 . As with Eq. (13), the updates are projected to the tangent space, resulting in the ODE given by

$$\ddot{\gamma} = -\|\dot{\gamma}\|^2 (I - \hat{\gamma}\hat{\gamma}^\top) \left(I - \hat{\gamma}\hat{\gamma}^\top \right) \nabla \log p(\gamma), \quad (14)$$

which can be solved by applying the Runge–Kutta (RK4) method to the first-order system of equations.

Algorithm 2: IVP solver for image extrapolation.

Input: image \mathcal{I}_0 ; source text prompt p_0 ; target text
prompt p_1 ; diffusion timestep τ ; optimizer
steps n ; VAE encoder \mathcal{E} ; CLIP text encoder
 \mathcal{C} ; hyperparameter β

Output: image sequence \mathcal{I} ; geodesic γ

```

1  $\{z_0, z_1\} \leftarrow \{\mathcal{C}(p_0), \mathcal{C}(p_1)\}$ 
2  $x \leftarrow \text{DDIM-F}(\mathcal{E}(\mathcal{I}_0), \tau, z_0)$ 
3  $\dot{x} \leftarrow \text{GetInitVelocity}(x, z_0, z_1)$ 
4  $\mathcal{S} \leftarrow \{(0, x)\}$ 
5  $\mathcal{I} \leftarrow \{\mathcal{I}_0\}$ 
6 for  $i = 1$  to  $n$  do
7    $z \leftarrow (1 - i/n)z_0 + (i/n)z_1$ 
8    $s \leftarrow \text{Score}(x, z, \tau, \beta)$ ; % Eq. (11)
9    $\ddot{x} \leftarrow \text{ODE}(s, x, \dot{x})$ ; % Eq. (14)
10   $\dot{x} \leftarrow \text{RK4}(\dot{x}, \ddot{x}, 1/n)$ 
11   $x \leftarrow \|x\|\text{RK4}(x, \dot{x}, 1/n) / \|\text{RK4}(x, \dot{x}, 1/n)\|$ 
12   $\mathcal{S} \leftarrow \mathcal{S} \cup \{(i/n, x)\}$ 
13   $\mathcal{I} \leftarrow \mathcal{I} \cup \{\mathcal{D}(\text{DDIM-B}(x, \tau, z))\}$ 
14  $\gamma \leftarrow \text{Interpolate}(\mathcal{S})$ 

```

5. Experiments

In this section, we first analyze whether short video sequences are geodesics in diffusion latent space. Second, we evaluate applications of the theory, focusing on the image interpolation task framed as solving a boundary value problem. For all experiments, we use a pre-trained Stable Diffusion v2.1-base model [37].

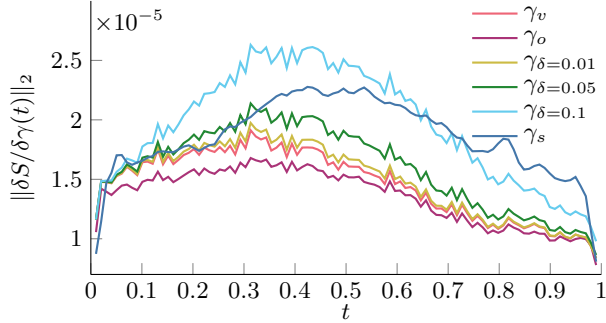
5.1. Geodesic analysis of videos

The objective of this experiment is to assess how close video clips are to geodesics.

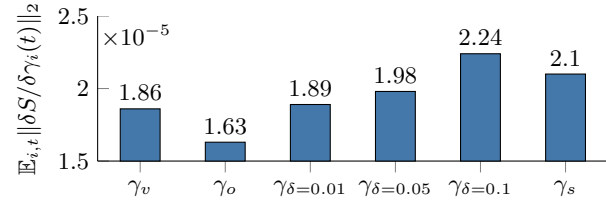
Dataset. We use the CLEVR framework [22] to render 40 synthetic videos with Blender. Each 100-frame sequence contains one of the three CLEVR objects (cube, sphere, cylinder) and has a single varying attribute, including camera rotation and translation, light source location, and the size, color and motion of the object.

Baselines. The path γ_v corresponding to the original video clip is compared to the path after geodesic optimization γ_o , the path after a sinusoidal perturbation $\gamma_\delta = \gamma_v + \delta \sin(\pi\gamma)$, and the path after smoothing γ_s , by fitting a smoothed cubic spline [7] with smoothing factor 0.9999.

Results. In Fig. 3a, we report the l_2 norm of the functional derivative in Eq. (5) as it varies along the path, for a single video clip (translating red cube). For a geodesic, this is zero everywhere along the path. In Fig. 3b, we report the average of this norm across the path and the dataset. If a video path was a geodesic in the diffusion latent space, we would expect (i) the norm along the path γ_v to be near zero, (ii)



(a) Geodesic gradient norm along the path, for the translating red cube.



(b) Geodesic gradient norm, averaged across the path and dataset.

Figure 3. Analysis of simple videos generated using CLEVR [22]. If a video path was a geodesic in the diffusion latent space, we would expect (i) the norm of the geodesic gradient along the path γ_v to be near zero, (ii) the norm to be close to that of the optimized path γ_o , and (iii) the norm of the perturbed γ_δ or smoothed γ_s paths to be larger. From the evidence, we conclude that many of the videos are approximately geodesic.

the norm to be close to that of the optimized path γ_o , and (iii) the norm of the perturbed γ_δ or smoothed γ_s paths to be larger. We observe these trends in the results, and conclude that many of the videos are approximately geodesic.

5.2. Applications

Here, we assess the performance of the geodesic solvers for the image interpolation and extrapolation tasks.

Datasets. We compile a union of datasets from prior works [50, 52, 53]. This includes MorphBench [53], which contains 90 image pairs of object animations and object metamorphoses; Animals and Humans [52], which contains 50 animal image pairs from AFHQ [6] with an LPIPS below 0.7 and 50 human face image pairs from CelebA-HQ [23] with an LPIPS below 0.6; and Web, which contains 20 image pairs sourced from publicly accessible websites, some of which have been used in other related studies [50].

Metrics. Quantitatively evaluating image interpolation and extrapolation is extremely challenging and subjective. Following previous work [52, 53], we report (1) the Fréchet inception distance (FID) [17] between the set of input images and the set of generated images to measure how close the distributions are; (2) the perceptual path length (PPL) [24] as the sum of LPIPS between adjacent images to assess the

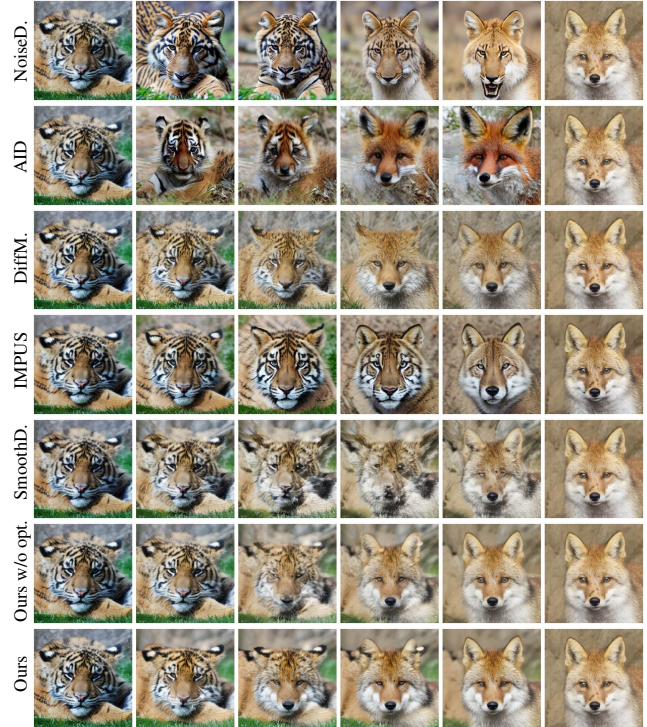


Figure 4. Qualitative comparison of image interpolation results.

directness of the generated image sequence; (3) perceptual distance variance (PDV) as the standard deviation of LPIPS between consecutive images to assess the consistency of transition rates across the sequence; and (4) the TOPIQ [4] score, an image quality metric that evaluates the perceptual quality of each generated image in alignment with human perception. To focus on the quality of interior frames, we compute a weighted TOPIQ score, detailed in Appendix C. We report all metrics on 17-frame sequences by sampling 15 intermediate images between the input pairs.

Implementation details. We use BLIP [31] to generate short text prompts p for each input image. For text inversion, we finetune the CLIP [36] encoded text embedding z , using 500 steps with a learning rate of 0.005 and the AdamW optimizer. For BVP optimization, we perform 400 steps of gradient descent with an initial learning rate of 0.1, following a linear learning rate schedule. The hyperparameters are set as $\tau = 600$, $\Delta\tau = 100$, $\beta = 0.002$, and $\sigma = 1$. As detailed in Sec. 4.3, we adopt a bisection strategy to add additional sample points every 100 steps. For IVP optimization, we set the iteration number to 200. To obtain a diverse set of initial velocities, we apply text inversion to the target text embedding using several generated images from random initial noise $\mathcal{N}(0, I)$, conditioned on the target prompt [39]. All methods use the same pre-trained diffusion model (Stable Diffusion v2.1-base) and text prompts, and default settings otherwise. Further details are in Appendix B.

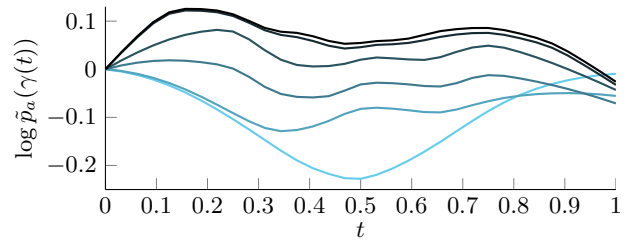
Table 1. Image interpolation results. We report the performance on three datasets: MorphBench (MB), Animals and Humans (AH), and Web data (Web). TF denotes “training-free”, which also prohibits fine-tuning; best results are in bold; second best ones are underlined. “Ours” represents the optimized geodesic path and “Ours w/o opt.” represents the unoptimized path as the great circle initialization.

Method	TF	FID ↓			PPL ↓			PDV ↓			TOPIQ ↑		
		AH	MB	Web	AH	MB	Web	AH	MB	Web	AH	MB	Web
NoiseDiffusion [54]	✓	71.85	100.87	201.08	2.507	2.718	3.738	0.117	0.111	0.099	0.700	<u>0.666</u>	<u>0.650</u>
AID [13]	✓	86.30	124.11	242.34	2.709	2.648	3.813	0.181	0.188	0.208	<u>0.699</u>	0.675	0.665
IMPUS [52]	✗	<u>25.75</u>	36.33	89.73	1.861	1.718	2.554	0.065	0.066	0.115	0.622	0.587	0.539
DiffMorpher [53]	✗	32.89	42.39	160.69	1.195	1.011	1.934	<u>0.018</u>	0.016	0.024	0.686	0.651	0.592
SmoothDiffusion [11]	✗	30.80	52.19	135.78	<u>0.903</u>	<u>0.879</u>	1.371	0.027	0.033	0.045	0.571	0.515	0.389
Ours w/o opt.	✓	24.71	<u>37.85</u>	<u>112.81</u>	0.874	0.841	<u>1.473</u>	0.032	0.035	0.053	0.584	0.546	0.466
Ours	✓	33.87	46.51	134.68	0.960	0.921	1.565	0.016	<u>0.022</u>	<u>0.026</u>	0.607	0.559	0.479

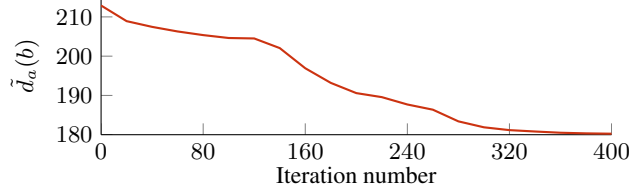
Table 2. We ablate the impact of different text conditioning strategies for BVP solver, including text inversion, positive (\oplus) prompts, and negative (\ominus) prompts, on the validation dataset.

Text inv.	\oplus prompt	\ominus prompt	FID ↓	PPL ↓	PDV ↓	TOPIQ ↑
✓			70.56	1.06	0.017	0.518
		✓	69.06	1.046	0.017	0.529
✓	✓		65.05	1.045	0.018	0.562
	✓	✓	64.72	1.037	0.019	0.564
✓	✓	✓	63.28	1.000	0.017	0.553

Results. We compare our method for the image interpolation task with several state-of-the-art diffusion-based methods, including NoiseDiffusion [54], DiffMorpher [53], IMPUS [52], AID [13], SmoothDiffusion [11] and report the result of a baseline (‘Ours w/o opt.’), which returns the initial path before geodesic optimization. This corresponds to the great circle trajectory, the most direct path between the endpoints on the sphere. The quantitative and qualitative results are presented in Tab. 1 and Figs. 4 and 6. We observe that methods involving finetuning the diffusion model on input images, like DiffMorpher and IMPUS, tend to score better with respect to most of the metrics. While AID and NoiseDiffusion generate high-quality images, they have a weaker connection to the input images. SmoothDiffusion [11], trained on the large LAION dataset, performs well with respect to perceptual path length, but has high variance and weaker image quality scores, especially on the partially OOD Web dataset. In contrast, our method has high directness (low PPL), high fidelity to the input distribution (low FID), and very high perceptual smoothness (low PDV), but has slightly lower image quality (low TOPIQ). Overall, our approach is on-par with the best methods without requiring any training. Finally, we show qualitative results for the image extrapolation task in Fig. 7, where we visualize two trajectories for each prompt.



(a) Relative log-probability along the path as optimization progresses, where iteration 0 is the lightest curve and iteration 400 is the darkest.



(b) Path length with respect to optimization step.

Figure 5. Example of the evolution of the probability density along the path and the path length during BVP optimization.

Analysis. We constructed a validation dataset for ablation and analysis by randomly selecting 25 image pairs from the three datasets (Sec. 5.2). In Fig. 5, we show that optimization correctly increases the log probability of samples along the path and decreases the path length. This indicates that the curve smoothly approaches a geodesic during optimization. We also present an ablation study in Tab. 2, where we show how the text conditioning formulation (text inversion, positive prompt, and negative prompt) contributes to the performance of the method. In Appendix D, we compare the time-linear conditioning signal with the constant one. In Appendix E, we analyze the trade-off between the different choices of hyperparameter settings.

5.3. Limitations

This work has several limitations. First, the metrics used for evaluating image interpolation struggle to capture inter-

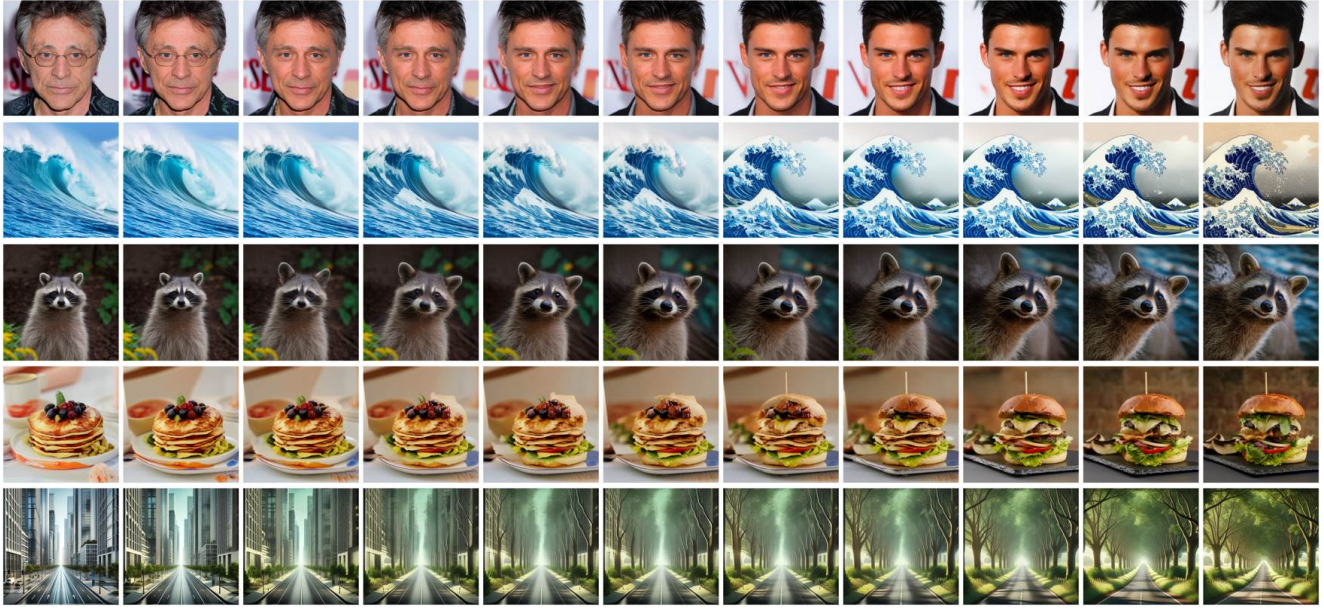


Figure 6. Qualitative image interpolation results using our geodesic BVP solver.

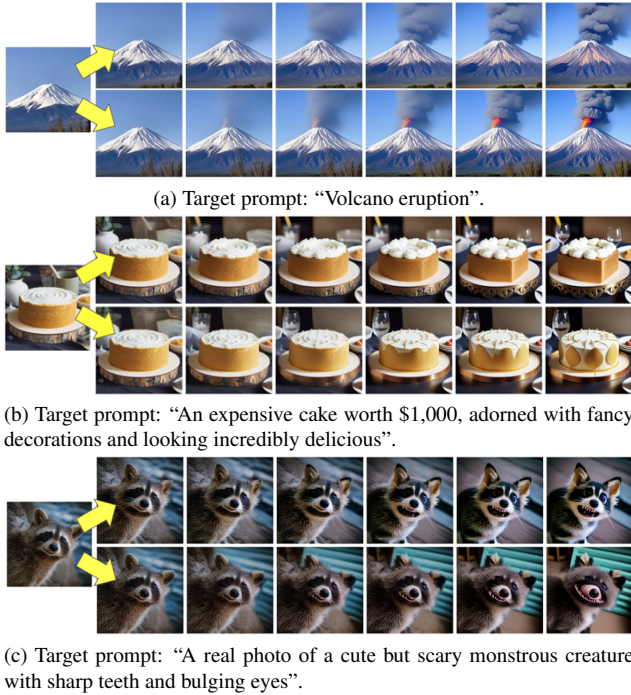


Figure 7. Qualitative image extrapolation results using our geodesic IVP solver. For each image, we plot two extrapolated paths, each with different initial velocities but the same prompt.

polation quality (smoothness, directness, and realism). In particular, FID, which measures fidelity to the input distribution, is unreliable when applied to small image sets like these, and the other metrics only capture individual aspects

of interpolation quality. Second, the method performs well for image morphing (local changes) but struggles with large camera motions or domain gaps, as illustrated in Figs. 15 and 16. We hypothesize that by initializing with a great circle, which also performs poorly in such cases, our optimizer gets stuck in nearby local optima rather than finding more optimal solutions. A global search strategy is indicated. Third, we approximate the log-probability gradient with a score distillation gradient, which is not necessarily well-aligned. Fourth, the path representation is inelegant: a spherical piecewise linear function (great circle arcs) for which the velocities and accelerations are approximated by fitting a cubic spline. Finally, the image extrapolation performance is unreliable, as determining a good initial velocity is challenging; a more robust approach is called for.

6. Conclusion

In this paper, we have presented the theory required for computing probability density geodesics in diffusion latent space and algorithms for solving the associated initial and boundary value problems. We also described how to compute several useful quantities for analysis: the relative probability density along the path, the geodesic distance between two points, and the geodesic gradient norm. Finally, we presented applications to image interpolation and extrapolation and evaluated the performance of these training-free approaches. We show that they perform comparably or better than existing state-of-the-art. We expect these techniques to be useful for tasks involving generative modeling, as well as for studying the distribution of image space.

Acknowledgments

We thank all reviewers and ACs for their constructive comments. This research was, in part, funded by the U.S. Government—DARPA TIAMAT HR00112490421. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Government.

References

- [1] Avrim Blum, John Hopcroft, and Ravindran Kannan. *Foundations of Data Science*. Cambridge University Press, 2020. 4
- [2] Tim Brooks, Aleksander Holynski, and Alexei A. Efros. Instructpix2pix: Learning to follow image editing instructions. In *CVPR*, pages 18392–18402, 2023. 2
- [3] Chen-Hao Chao, Wei-Fang Sun, Bo-Wun Cheng, and Chun-Yi Lee. On investigating the conservative property of score-based generative models. In *ICML*, pages 4076–4095, 2023. 3
- [4] Chaofeng Chen, Jiadi Mo, Jingwen Hou, Haoning Wu, Liang Liao, Wenxiu Sun, Qiong Yan, and Weisi Lin. Topiq: A top-down approach from semantics to distortions for image quality assessment. *IEEE TIP*, 2024. 6
- [5] Junhao Chen, Manyi Li, Zherong Pan, Xifeng Gao, and Changhe Tu. Varying manifolds in diffusion: From time-varying geometries to visual saliency. *arXiv preprint arXiv:2406.18588*, 2024. 4
- [6] Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. Stargan v2: Diverse image synthesis for multiple domains. In *CVPR*, pages 8188–8197, 2020. 6
- [7] Carl De Boor. *A practical guide to splines*. Springer New York, 1978. 5
- [8] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat GANs on image synthesis. In *NeurIPS*, pages 8780–8794, 2021. 2
- [9] Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit Haim Bermano, Gal Chechik, and Daniel Cohen-or. An image is worth one word: Personalizing text-to-image generation using textual inversion. In *ICLR*, 2023. 2
- [10] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NeurIPS*, 2014. 2
- [11] Jiayi Guo, Xingqian Xu, Yifan Pu, Zanlin Ni, Chaofei Wang, Manushree Vasu, Shiji Song, Gao Huang, and Humphrey Shi. Smooth diffusion: Crafting smooth latent spaces in diffusion models. In *CVPR*, pages 7548–7558, 2024. 2, 7
- [12] Ligong Han, Yinxiao Li, Han Zhang, Peyman Milanfar, Dimitris N. Metaxas, and Feng Yang. Svdif: Compact parameter space for diffusion fine-tuning. In *ICCV*, pages 7289–7300, 2023. 2
- [13] Qiyuan He, Jinghao Wang, Ziwei Liu, and Angela Yao. Aid: Attention interpolation of text-to-image diffusion. *NeurIPS*, 2024. 2, 7
- [14] Amir Hertz, Kfir Aberman, and Daniel Cohen-Or. Delta denoising score. In *ICCV*, pages 2328–2337, 2023. 2
- [15] Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-or. Prompt-to-prompt image editing with cross-attention control. In *ICLR*, 2023. 2
- [16] Amir Hertz, Andrey Voynov, Shlomi Fruchter, and Daniel Cohen-Or. Style aligned image generation via shared attention. In *CVPR*, pages 4775–4785, 2024. 2
- [17] 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. *NeurIPS*, 30, 2017. 6
- [18] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021. 4
- [19] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, 2020. 2, 3
- [20] Aapo Hyvärinen. Estimation of non-normalized statistical models by score matching. *JMLR*, 6:695–709, 2005. 3
- [21] András Jankovics. Automatic image morphing. github.com/jankovicsandras/autoimagemorph, 2022. 2
- [22] Justin Johnson, Bharath Hariharan, Laurens Van Der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. CLEVR: A diagnostic dataset for compositional language and elementary visual reasoning. In *CVPR*, pages 2901–2910, 2017. 5, 6
- [23] Tero Karras. Progressive growing of gans for improved quality, stability, and variation. *ICLR*, 2018. 6
- [24] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *CVPR*, pages 4401–4410, 2019. 6
- [25] Oren Katzir, Or Patashnik, Daniel Cohen-Or, and Dani Lischinski. Noise-free score distillation. In *ICLR*, 2024. 4
- [26] Bahjat Kawar, Shiran Zada, Oran Lang, Omer Tov, Huiwen Chang, Tali Dekel, Inbar Mosseri, and Michal Irani. Imagic: Text-based real image editing with diffusion models. In *CVPR*, pages 6007–6017, 2023. 2
- [27] Jacek Kierzenka and Lawrence F Shampine. A BVP solver based on residual control and the maltab pse. *ACM Transactions on Mathematical Software (TOMS)*, 27(3):299–316, 2001. 4
- [28] Seung Wook Kim, Karsten Kreis, Daiqing Li, Antonio Torralba, and Sanja Fidler. Polymorphic-gan: Generating aligned samples across multiple domains with learned morph maps. In *CVPR*, pages 10630–10640, 2022. 2
- [29] Nupur Kumari, Bingliang Zhang, Richard Zhang, Eli Shechtman, and Jun-Yan Zhu. Multi-concept customization of text-to-image diffusion. In *CVPR*, pages 1931–1941, 2023. 2
- [30] Seungyong Lee, George Wolberg, Kyung Yong Chwa, and Sung Yong Shin. Image metamorphosis with scattered feature constraints. *IEEE TVCG*, 2(4):337–354, 1996. 2
- [31] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *ICML*, 2022. 6

- [32] Jing Liao, Rodolfo S. Lima, Diego Nehab, Hugues Hoppe, Pedro V. Sander, and Jinhui Yu. Automating image morphing using structural similarity on a halfway domain. *ACM TOG*, 33(5):168:1–168:12, 2014. 2
- [33] Sanghun Park, Kwanggyoon Seo, and Junyong Noh. Neural crossbreed: neural based image metamorphosis. *ACM TOG*, 39(6):224:1–224:15, 2020. 2
- [34] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. SDXL: Improving latent diffusion models for high-resolution image synthesis. In *ICLR*, 2024. 2
- [35] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. In *ICLR*, 2023. 4
- [36] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, pages 8748–8763, 2021. 4, 6
- [37] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, pages 10684–10695, 2022. 1, 2, 3, 4, 5
- [38] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *CVPR*, pages 22500–22510. IEEE, 2023. 2
- [39] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *CVPR*, pages 22500–22510, 2023. 6
- [40] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. In *NeurIPS*, 2022. 2, 4
- [41] Steven M Seitz and Charles R Dyer. View morphing. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 21–30, 1996. 2
- [42] Dror Simon and Aviad Aberdam. Barycenters of natural images - constrained wasserstein barycenters for image morphing. In *CVPR*, pages 7907–7916, 2020. 2
- [43] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *ICML*, pages 2256–2265, 2015. 1
- [44] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *ICLR*, 2021. 3
- [45] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. In *NeurIPS*, 2019. 4
- [46] Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *ICLR*, 2021. 1
- [47] Charles Stein. A bound for the error in the normal approximation to the distribution of a sum of dependent random variables. In *Proceedings of the Sixth Berkeley Symposium on Mathematical Statistics and Probability, Volume 2: Probability Theory*, pages 583–603. University of California Press, 1972. 1, 3
- [48] Narek Tumanyan, Michal Geyer, Shai Bagon, and Tali Dekel. Plug-and-play diffusion features for text-driven image-to-image translation. In *CVPR*, pages 1921–1930, 2023. 2
- [49] Andrey Voynov, Qinghao Chu, Daniel Cohen-Or, and Kfir Aberman. P+: extended textual conditioning in text-to-image generation. *arXiv preprint arXiv:2303.09522*, 2023. 2
- [50] Clinton Wang and Polina Golland. Interpolating between images with diffusion models. *ICML Workshop on Deployment Challenges for Generative AI*, 2023. 2, 6
- [51] George Wolberg. Image morphing: a survey. *The visual computer*, 14(8-9):360–372, 1998. 2
- [52] Zhaoyuan Yang, Zhengyang Yu, Zhiwei Xu, Jaskirat Singh, Jing Zhang, Dylan Campbell, Peter Tu, and Richard Hartley. IMPUS: Image Morphing with Perceptually-Uniform Sampling using Diffusion Models. In *ICLR*, 2024. 2, 6, 7
- [53] Kaiwen Zhang, Yifan Zhou, Xudong Xu, Bo Dai, and Xingang Pan. DiffMorpher: Unleashing the capability of diffusion models for image morphing. In *CVPR*, pages 7912–7921, 2024. 2, 6, 7
- [54] PengFei Zheng, Yonggang Zhang, Zhen Fang, Tongliang Liu, Defu Lian, and Bo Han. Noisediffusion: Correcting noise for image interpolation with diffusion models beyond spherical linear interpolation. In *ICLR*, 2024. 7
- [55] Lei Zhu, Yan Yang, Steven Haker, and Allen Tannenbaum. An image morphing technique based on optimal mass preserving mapping. *IEEE TIP*, 16(6):1481–1495, 2007. 2
- [56] Bhushan Zope and Soniya B Zope. A survey of morphing techniques. *International Journal of Advanced Engineering, Management and Science*, 3(2):239773, 2017. 2