Wukong's 72 Transformations: High-fidelity Textured 3D Morphing via Flow Models

Minghao Yin[1] **Yukang Cao**[2] **Kai Han**[1*]
[1]Visual AI Lab, The University of Hong Kong [2]S-Lab, Nanyang Technological University
yinmh@connect.hku.hk yukang.cao@ntu.edu.sg kaihanx@hku.hk

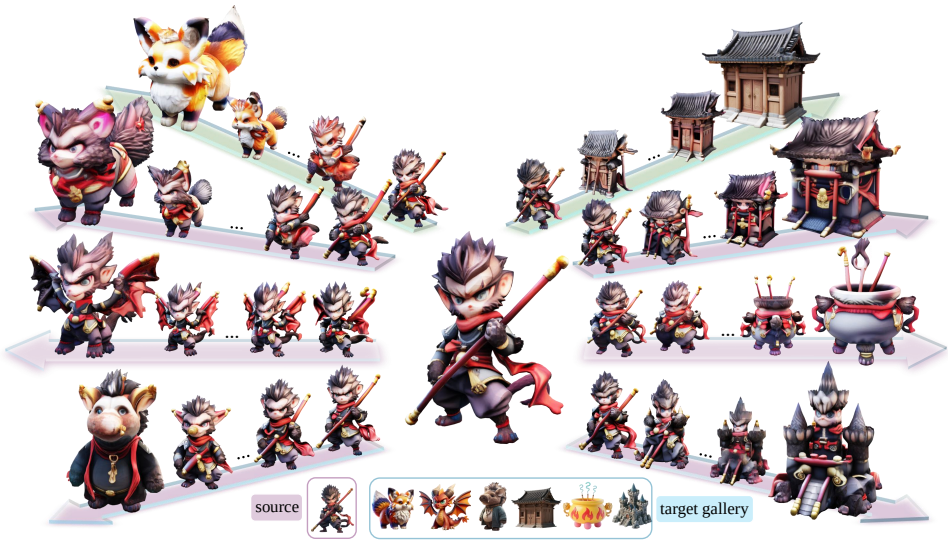

Figure 1: **High-fidelity textured 3D morphing of Wukong.** Taking an image of Wukong (bottom left) as the source and an image of another character (bottom right) as the target, we demonstrate two types of *textured 3D morphing* by our method: (i) Purple arrows indicate texture-controlled morphing, where the geometric structure changes while preserving detailed textures from the source; (ii) Green arrows indicate textured 3D morphing guided by the target prompt.

Abstract

We present WUKONG, a novel training-free framework for high-fidelity textured 3D morphing that takes a pair of source and target prompts (image or text) as input. Unlike conventional methods—which rely on manual correspondence matching and deformation trajectory estimation (limiting generalization and requiring costly preprocessing)—WUKONG leverages the generative prior of flow-based transformers to produce high-fidelity 3D transitions with rich texture details. To ensure smooth shape transitions, we exploit the inherent continuity of flow-based generative processes and formulate morphing as an optimal transport barycenter problem. We further introduce a sequential initialization strategy to prevent abrupt geometric distortions and preserve identity coherence. For faithful texture preservation, we propose a similarity-guided semantic consistency mechanism that selectively retains high-frequency details and enables precise control over blending dynamics. This avoids common artifacts like oversmoothing while maintaining semantic fidelity. Extensive quantitative and qualitative evaluations demonstrate that WUKONG significantly outperforms state-of-the-art methods,

*Corresponding author.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

achieving superior results across diverse geometry and texture variations. Project page: https://visual-ai.github.io/wukong

1 Introduction

3D morphing techniques (Seitz and Dyer, 1996; Shechtman et al., 2010; Tsai et al., 2022; Kim et al., 2024) aim to produce smooth transitions between the source object and target object by gradually altering 3D attributes such as shape and texture. These methods have broad applications in gaming, animation, and cinematic transitions. Existing approaches primarily focus on geometric transformations, relying on correspondence matching (Deng et al., 2023) and deformation trajectory estimation (Eisenberger et al., 2021). However, most existing methods are limited to untextured 3D meshes, leaving textured 3D morphing a largely under-explored problem. Moreover, their effectiveness is constrained by the limited availability of large-scale 3D datasets with source-target pairs, often resulting in unsatisfactory outputs when applied to unseen data. To bridge this gap, we introduce WUKONG, a method for *high-fidelity textured 3D morphing from a pair of image or text prompts*. This name is inspired by the legendary character Wukong and his 72 earthly transformations, as exemplified by the morphing result in Fig. 1.

With recent advances in large-scale 3D generation and reconstruction (Xiang et al., 2024; Tochilkin et al., 2024; Li et al., 2025), one may consider to address the 3D morphing problem by training a feed-forward network to model the transitions between two input objects. However, constructing paired 3D data for training based on existing large-scale 3D data (Deitke et al., 2024) is non-trivial, both technically and economically. Therefore, such a straightforward approach is infeasible.

Recently, 2D image morphing has achieved remarkable success (Qiyuan et al., 2024; Zhang et al., 2024a), driven by advances in 2D diffusion models (*e.g.*, Stable Diffusion (Rombach et al., 2022)). Inspired by this progress, we aim to extend these successes to 3D – morphing not only shapes but also textures – without relying on large-scale paired data. In other words, we aim to develop a training-free framework for textured 3D morphing by leveraging the strong priors of generative models. However, building such a framework poses significant challenges due to the absence of large-scale 3D data with continues shape and texture transitions.

In paper, we propose WUKONG, a novel framework for high-fidelity textured 3D morphing that takes image or text pairs as input to delineate source and target objects. Built upon a pre-trained flow-based transformer (Xiang et al., 2024) for 3D generation, WUKONG bridges the gap between 3D shapes and image/text conditions. Leveraging the deterministic property of flow models (where intermediate states are not stochastic), we derive morphing trajectories by solving a free-support Wasserstein barycenter problem. Additionally, we introduce a sequential initialization strategy to enhance the smoothness of the transitions along the barycentric trajectory. This design also allows us to handle the texture morphing in 3D in a similar fashion. However, this is not sufficient to produce faithful texture. As naïve interpolation leads to undesirable blending artifacts (*e.g.*, loss of facial details and color patterns). To address this, we propose a similarity-guided consistency mechanism that selectively preserves high-frequency texture details while providing finer control over transition dynamics.

The main contributions of this work are as follows: (i) We propose WUKONG, a novel framework for high-fidelity textured 3D morphing that takes a pair of image or text prompts as input. By leveraging a flow-based generative model as a prior, WUKONG enables smooth and controllable shape and texture interpolation. (ii) We introduce a method to derive faithful intermediate morphing states by solving an optimal transport barycenter problem, further augmented by a sequential initialization strategy to facilitate smooth transitions. (iii) To address texture degradation in morphing, we propose a similarity-guided semantic consistency mechanism that selectively preserves high-frequency texture details, enabling finer control over texture transitions. (iv) Through extensive experiments on diverse 3D morphing scenarios—spanning different object categories with varying geometry and significant texture changes—we demonstrate that WUKONG outperforms existing methods, establishing a new state-of-the-art in textured 3D morphing.

2 Related work

2D morphing Classical image morphing (Liao et al., 2014; Beier and Neely, 1992; Darabi et al., 2012; Shechtman et al., 2010) typically involves three key steps: (1) finding feature-based correspondence; (2) mapping between two images using optimization frameworks; and (3) auxiliary techniques to ensure smooth transitional continuity. However, strong priors and limited model expressiveness

often lead to ghosting artifacts. With growing data availability, data-driven morphing methods (Fish et al., 2020; Averbuch-Elor et al., 2016) shifted from explicit mappings to learning over dataset distributions using deep models. However, their generalization is limited by dataset-specific training. Recent approaches like DiffMorpher (Zhang et al., 2024a) address this by leveraging pre-trained diffusion models, enabling more flexible morphing. DiffMorpher achieves morphing by interpolating noise, conditional inputs, and selectively blending model parameters, enabling strong shape and texture transitions. Inspired by this idea, we propose a novel method that replaces linear interpolation with an optimal transport-based strategy, offering a more principled and effective interpolation of latent conditions.

3D generation To achieve 3D generation, two dominant paradigms have emerged: (1) Distilling useful priors from pretrained generative models into a 3D feed-forward reconstruction algorithm; (2) Training a unified 3D generative model from scratch. In the first paradigm, knowledge distillation from large generative models occurs via gradient-based or data-based methods. Data distillation fine-tunes 2D models to generate multi-view images (Yu et al., 2024; Shi et al., 2024; Shriram et al., 2025), which are then reconstructed into 3D using methods like Gaussian splatting (Kerbl et al., 2023). Gradient distillation, exemplified by Score Distillation Sampling (Poole et al., 2023), guides 3D optimization directly. However, both lack a latent 3D space, limiting structural control. To bridge this gap, native 3D generative models have emerged (Lan et al., 2025; Zeng et al., 2022; Zhang et al., 2024b), typically combining a VAE for dimensionality reduction with a 3D generative model. However, most are limited to either explicit (*e.g.*, point clouds, voxels, meshes) or implicit (*e.g.*, neural fields, Gaussians) formats. Trellis (Xiang et al., 2024) addresses this by introducing a Structured Latent Representation (SLAT) for flexible multi-format generation. We inherit Trellis's versatility to support diverse 3D modalities.

3D morphing Early 3D morphing research centered on shape transition (Tam et al., 2013), following a three-step pipeline: finding correspondence, modeling the mapping, and refinement. Classical methods employed techniques like Wasserstein distance (Tsai et al., 2022; Solomon et al., 2015; Sorkine and Alexa, 2007; Ren et al., 2020; Eisenberger et al., 2021). Despite their contributions, these methods suffer from oversimplified assumptions and high input sensitivity. Recent learning-based methods improve robustness by integrating foundation model priors. NSSM (Morreale et al., 2024) uses DINOv2 (Oquab et al., 2023) for 3D correspondence; SRIF (Sun et al., 2024) incorporates a diffusion prior from DiffMorpher (Zhang et al., 2024a). Emerging works (Gao et al., 2023; Yang et al., 2025; Michel et al., 2022) explore text-driven morphing using CLIP (Radford et al., 2021), though limited by CLIP's coarse latent space. Other efforts leverage topology-aligned datasets (Eisenberger et al., 2021) with in-domain training to enhance semantic consistency (Yumer et al., 2015). Shape-only 3D morphing has limited practical use without texture and appearance cues. A recent concurrent work (Yang et al., 2025) explores morphing textured 3D representations using generative models, bypassing explicit correspondence computation. However, it mainly focuses on parameter fusion and pays less attention to latent condition interpolation. In contrast, our method emphasizes a principled interpolation schedule for latent conditions. Additionally, we adopt a flow-based model with a unified 3D representation, enabling flexible output across diverse formats.

3 Method

3.1 Overview and formulation

Given two input prompts (P_{source}, P_{target}) in image or text form, our method learns an interpolation function $\mathbf{I}$ to generates a coherent sequence of 3D textured meshes $\mathcal{G} = \{\mathbf{G}_\alpha\}_{\alpha=0}^{J+1}$, which forms a smooth morphing trajectory between the two concepts. Among them, $\mathbf{G}_0$ and $\mathbf{G}_{J+1}$ denote the 3D generation of the input prompts P_{source} and P_{target}, respectively. Formally, the problem can be formulated as:

$$\mathbf{G}_\alpha = \Phi(\mathbf{I}(\mathcal{E}(P_{\text{source}}), \mathcal{E}(P_{\text{target}}), \alpha)), \tag{1}$$

where Φ denotes a flow-based 3D generator, $\mathcal{E}$ represents the encoders that embed the prompts into the latent space. For feature extraction, we use DINOv2 (Oquab et al., 2023) as the encoder $\mathcal{E}$ for image prompts and CLIP (Radford et al., 2021) for textual inputs.

In the rest of this section, we will discuss the details of our employed flow-based 3D generator Φ and how we formulate the interpolation function $\mathbf{I}$. From a general perspective, an effective interpolation strategy in 3D morphing should satisfy two key requirements: (1) smooth shape transitions that

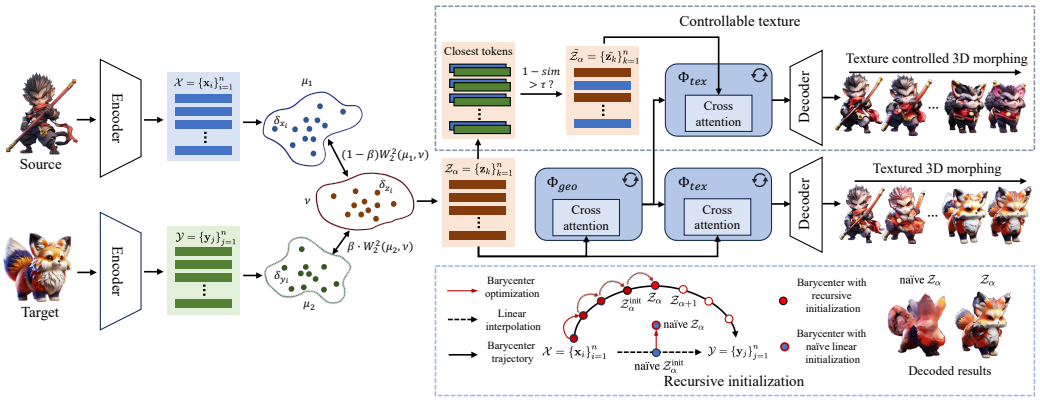

Figure 2: **Our WUKONG framework.** Given a source and a target (image or text), we extract features using pretrained encoders and treat the condition tokens as empirical distributions. We compute their Wasserstein barycenter with weights $(1-\beta)$ and β to obtain interpolated condition tokens $\mathcal{Z}$. These are fed into a shared geometry flow model Φ_{geo} and texture flow model Φ_{tex} to generate 3D outputs at different α values, producing textured 3D morphs. The top-right shows our texture-controlled morphing branch, and the bottom-right illustrates the recursive initialization strategy.

preserve identity and prevent abrupt changes, and (2) controllable texture blending. As depicted in Fig. 2, we propose to model these two properties with (1) shape interpolation via optimal transport barycenter and (2) controllable texture generation via selective interpolation.

3.2 Flow-based 3D generator

Recently, Trellis (Xiang et al., 2024) advances 3D generation by introducing a unified structured latent representation via rectified flow transformers (Xiang et al., 2024). It achieves high-fidelity outputs from image or text inputs by training a feed-forward pipeline, with the generation process at test time separated into three distinct phases:

(1) Geometric structure generation that generates sparse structure $\boldsymbol{p} = \{p_i\}_{i=1}^{L}$ from the condition $\mathcal{C}$, which is either text-embedded CLIP features (Radford et al., 2021) or image-encoded DINOv2 features (Oquab et al., 2023):

$$\{p_i\}_{i=1}^{L} = \Phi_{\text{geo}}(\mathcal{C}, t), \tag{2}$$

where Φ_{geo} denotes the flow transformer backbone, t is the timestep, and L represents the number of active voxels.

(2) Textured latent generation that generates latents $\boldsymbol{h} = \{h_i\}_{i=1}^{L}$ given the structure $\{p_i\}_{i=1}^{L}$:

$$\{h_i\}_{i=1}^{L} = \Phi_{\text{tex}}(\boldsymbol{p}, \gamma(\boldsymbol{x}), \mathcal{C}, t), \tag{3}$$

where $\gamma(\boldsymbol{x})$ is the positional embedded points.

(3) Latent decoder that decodes the structured latents $\boldsymbol{s} = \{(h_i, p_i)\}_{i=1}^{L}$ into a 3D representation, which we opt for meshes in our experiments:

$$\mathcal{D}_{\text{mesh}}(\boldsymbol{s}) \rightarrow \{\{w_i^j, d_i^j\}_{j=1}^{64}\}_{i=1}^{L}, \tag{4}$$

where $w_i^j \in \mathbb{R}^{45}$ are the flexible parameters in FlexiCubes (Shen et al., 2023) and $d_i^j \in \mathbb{R}^8$ are signed distance values for the eight vertices of the corresponding voxel. Note that our employed 3D generator (Xiang et al., 2024) also supports output in both 3D Gaussian Splatting (Kerbl et al., 2023) and NeRF (Martin-Brualla et al., 2021) formats, and our method naturally inherits this capability.

For our experiment, we utilize the pre-trained Trellis model as the backbone network due to its demonstrated efficiency and high-quality outputs:

$$\Phi = \{\Phi_{\text{geo}}, \Phi_{\text{tex}}, \mathcal{D}_{\text{mesh}}\}. \tag{5}$$

To maintain distinct control over different attributes, we design separate interpolation functions for geometric and texture features:

$$\mathbf{I} = \{\mathbf{I}_{\text{geo}}, \mathbf{I}_{\text{tex}}\}. \tag{6}$$

3.3 Shape interpolation via optimal transport barycenter

Given $\mathcal{C}_0 = \mathcal{E}(P_{\text{source}})$ and $\mathcal{C}_{J+1} = \mathcal{E}(P_{\text{target}})$, which represent the conditioning features respectively encoded from the input prompts P_{source} and P_{target}, we first model them as discrete probability distributions in feature space by extracting their respective sets of feature tokens:

$$\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^n \subset \mathbb{R}^m, \quad \mathcal{Y} = \{\mathbf{y}_j\}_{j=1}^n \subset \mathbb{R}^m, \tag{7}$$

where m is the embedding dimension. We can then obtain their empirical distribution:

$$\mu_1 = \sum_{i=1}^n a_i \delta_{\mathbf{x}_i}, \quad \mu_2 = \sum_{j=1}^n b_j \delta_{\mathbf{y}_j}, \tag{8}$$

where $\delta_{\mathbf{x}_i}$ and $\delta_{\mathbf{y}_j}$ denote the Dirac measures located at the tokens $\mathbf{x}_i$ and $\mathbf{y}_j$, respectively. a_i and b_j are weights assigned to each token, which we set $a_i = b_j = 1/n$ in our experiments.

To obtain the geometric structure $\boldsymbol{p}_\alpha$ at an intermediate step α via the generator Φ_{geo}, we require its corresponding interpolated condition $\mathcal{Z}_\alpha$. Since Φ_{geo} is a deterministic mapping from conditions to structures, we propose to obtain $\mathcal{Z}_\alpha$ by solving a free-support Wasserstein barycenter problem:

$$\begin{aligned} \mathbf{I}_{\text{geo}} : \mathcal{Z}_\alpha = \arg\min_{\mathcal{Z}} [(1-\beta)\cdot \mathcal{W}_2^2(\mu_1, \nu) + \beta \cdot \mathcal{W}_2^2(\mu_2, \nu)], \quad \beta = \alpha/(J+1), \\ \mathcal{W}_2^2(\mu, \nu) = \inf_{\gamma \in \Gamma(\mu,\nu)} \int_{n\times n} d(x,y)^2 \, \mathrm{d}\gamma(x,y), \end{aligned} \tag{9}$$

where $\mathcal{W}_2^2(\mu, \nu)$ denotes the 2-Wasserstein distance, $\nu = \sum_{k=1}^n c_k \delta_{\mathbf{z}_k}$ is the target interpolated distribution with learnable support points $\{\mathbf{z}_k\}_{k=1}^n$ uniformly weighted by $c_k = 1/n$. The interpolated tokens $\mathcal{Z}_\alpha = \{\mathbf{z}_k\}_{k=1}^n$ generated through this process serve as input to our geometric structure generator Φ_{geo}, which produces the corresponding 3D structure $\boldsymbol{p}_\alpha$.

A critical design for solving the free-support Wasserstein barycenter problem is the initialization of support points $\{\mathbf{z}_k\}$. While a naïve linear interpolation approach, $\mathbf{z}_{k,\text{init}} = (1-\beta)\cdot \mathbf{x}_k + \beta \cdot \mathbf{y}_k$, might seem reasonable, we empirically observe that this often results in discontinuous or inconsistent interpolations. This is particularly evident when interpolating between either: (1) geometrically distant shapes, or (2) semantically divergent conditions (see Fig. 8). The failure arises because linear initialization overlooks the structure of token distributions, often leading the barycenter optimization to unstable or unrealistic interpolations.

We address this limitation with a sequential initialization scheme that guarantees smooth transitions along the barycenter trajectory. Specifically, for interpolation step $\alpha \in [0, J+1]$, we initialize its barycenter using the optimized solution from the previous step:

$$\mathbf{z}_{k,\text{init}}^{(\alpha)} = \begin{cases} \mathbf{x}_k, & \text{if } \alpha = 0 \\ \mathbf{z}_k^{(\alpha-1)}, & \text{otherwise.} \end{cases} \tag{10}$$

This recursive approach supports the barycenter evolves continuously on the data manifold, maintaining: (1) geometric coherence: support points adapt gradually, reducing abrupt deviations or poor local minima, and (2) semantic stability: intermediate shapes retain recognizable object parts and consistent structural semantics throughout the interpolation.

3.4 Controllable texture generation via selective interpolation

Although our shape-level interpolation generates semantically coherent 3D transitions, it tends to produce undesirable texture averaging, particularly for fine details like facial features, color patterns, and garment elements, due to its uniform application across all token regions. To overcome this limitation, we introduce a selective interpolation strategy that: (1) preserves high-frequency texture details through similarity-guided blending, while (2) maintaining smooth semantic transitions at the structural level. This approach enables controlled transfer of local texture attributes from one condition while harmoniously integrating global characteristics from the other.

Building upon the geometric interpolation introduced in Sec. 3.3, we first compute the free-support barycenter $\mathcal{Z}_\alpha = \{\mathbf{z}_k\}_{k=1}^n$ from Eq. 9. While this provides a general interpolation baseline, it may

dilute fine-grained details from the source. To mitigate this, we augment the interpolation with a semantic consistency evaluation between interpolated points $\mathbf{z}_k$ and the source tokens $\mathcal{X}$, $\mathcal{Y}$.

Specifically, for each barycenter point $\mathbf{z}_k$, we identify its closest source tokens $\mathbf{x}_i$ and $\mathbf{y}_j$ and compute the cosine similarity $\texttt{sim} = \cos(\mathbf{x}_i, \mathbf{y}_j)$. We then selectively retain high-frequency information from either $\mathbf{x}_i$ or the interpolated tokens based on similarity:

$$\mathbf{I}_{\text{tex}} : \tilde{\mathcal{Z}}_\alpha = \{\tilde{\mathbf{z}}_k\}, \quad \text{while} \quad \begin{cases} \tilde{\mathbf{z}}_k = \mathbf{z}_k, & \text{if } 1 - \texttt{sim} > \tau \\ \tilde{\mathbf{z}}_k = \mathbf{x}_i, & \text{otherwise.} \end{cases} \tag{11}$$

Here, $\tau \in [0, 1]$ is a pre-defined similarity threshold that determines whether semantic discrepancy is large enough to justify interpolation. See Fig. 6 for an analysis across different values of τ. This approach enables asymmetric texture fusion, where more visually salient or personalized texture features can be preserved from one condition, while maintaining semantically meaningful global structure via the barycenter.

The selectively refined condition tokens $\tilde{\mathcal{Z}}_\alpha$ are then passed into the 3D textured structured latents generator Φ_{tex} to produce the final latent representation $\boldsymbol{h}_\alpha$. The interpolated 3D textured mesh at step α can then be obtained through the decoding function Eq. 4:

$$\mathbf{G}_\alpha = \mathcal{D}_{\text{mesh}}(\boldsymbol{s}_\alpha), \quad \text{where } \boldsymbol{s}_\alpha = \{\boldsymbol{h}_\alpha, \boldsymbol{p}_\alpha\} = \{(h_{i,\alpha}, p_{i,\alpha})\}_{i=1}^{L}. \tag{12}$$

4 Experimental results

Implementation details We adopt the Trellis framework as our 3D generative model. Both the structure flow and texture flow models are implemented using rectified flow with 25 sampling steps each. The classifier-free guidance (CFG) strength is set to 3. DINOv2 (Oquab et al., 2023) and CLIP (Radford et al., 2021) are used for image and text feature extraction. Both the structure and texture flow-based generator Φ_{geo}, Φ_{tex} contain 21 cross-attention blocks, where interpolation is performed on every condition token before each cross-attention layer. The morphing coefficient α is uniformly sampled during morphing and we set $J = 6$ for experiments presented in the paper. For shape interpolation, we compute the token-wise barycenter using the free-support Wasserstein barycenter implemented by `ot.lp.free_support_barycenter` (Lindheim, 2023). We set the maximum number of optimization iterations to 100, and the convergence threshold (stop criterion) to 1×10^{-5}. We conduct experiments on an NVIDIA A100. Generating a single morphed 3D output takes approximately 30 seconds.

Metrics Following (Yang et al., 2025), we evaluate textured 3D morphing quality using metrics for fidelity, plausibility, and smoothness on their input pairs: (1) FID (Heusel et al., 2017) for visual fidelity; (2) STP-GPT and SEP-GPT for structural and semantic consistency; (3) GPT-4o (Hurst et al., 2024) for visual plausibility (0–1 score); (4) PPL (Karras et al., 2019) for perceptual smoothness; and (5) V-CLIP (Ma et al., 2022), which measures semantic alignment to "a smooth transformation from A to B" using cosine similarity in a joint embedding space.

Baseline methods for evaluation We compare our 3D textured morphing results against two baseline methods: 3DRM (Yang et al., 2025) and MorphFlow (Tsai et al., 2022). To further evaluate performance, we render the 3D meshes into 2D images and compare them with state-of-the-art 2D image morphing approaches, including DiffMorpher (Zhang et al., 2024a), AID (Qiyuan et al., 2024), MV-Adapter (Jin et al., 2024), and Luma (Luma Labs AI, 2025).

Table 1: **Quantitative comparison.**

Model	FID ↓	STP-GPT ↑	SEP-GPT ↑	PPL ↓	V-CLIP ↑
DiffMorpher	218.07	0.14	0.10	5.23	0.61
AID	115.72	0.46	0.62	4.68	0.74
MV-Adapter	120.93	0.44	0.49	7.29	0.67
Luma	95.49	0.69	0.65	7.37	0.70
MorphFlow	147.70	0.71	0.79	3.10	0.78
3DRM	6.36	0.93	0.88	3.02	0.84
Ours	**4.01**	**1.00**	**1.00**	**2.91**	**0.90**

4.1 Main results

4.1.1 Quantitative results

In Table 1, we compare our method against a range of baselines across multiple evaluation metrics. As can be observed, our method consistently outperforms existing approaches across all metrics, demonstrating superior quality and consistency in both shape and texture morphing. Specifically, we outperform state-of-the-art textured 3D morphing methods across the board, *i.e.*, 3DRM (Yang et al., 2025) and MorphFlow (Tsai et al., 2022), showing clear improvements in perceptual quality and semantic coherence. Furthermore, compared with image-based morphing methods (Zhang et al., 2024a; Qiyuan et al., 2024; Jin et al., 2024; Luma Labs AI, 2025), our approach also exhibits much stronger 3D consistency and higher fidelity in both appearance and structure. See Appendix B for more quantitative evaluations.

Figure 3: **3D morphing with image prompts.** The leftmost column shows source image prompts, while the rightmost column shows target prompts. Intermediate columns depict the morphing trajectory generated by our method.

Figure 4: **3D morphing with text prompts.** The leftmost column shows the source text descriptions, and the rightmost column shows the target prompts. Intermediate results visualize the smooth morphing trajectory generated by our method.

4.1.2 Qualitative results

Image-conditioned 3D morphing Fig. 3 shows image-conditioned 3D morphing results. Each row presents a smooth transition from source to target, with consistent shape and texture interpolation.

Intermediate frames preserve structure without distortions. Rows 3–5 highlight our method's ability to handle cross-category morphing with clear semantic consistency.

Text-conditioned 3D morphing Our method also enables 3D morphing conditioned on textual descriptions, allowing users to generate transitions directly from text prompts. As shown in Fig. 4, the results exhibit smooth transitions in both shape and texture, with intermediate outputs maintaining high fidelity and semantic alignment. Notably, the second row captures a precise castle-to-room transformation, while the fourth row demonstrates realistic face morphing with consistent structure.

Figure 5: **Qualitative comparison with current SOTA method 3DRM.** Rows 1 and 3 show 3DRM's generated results, while rows 2 and 4 display our method's outputs.

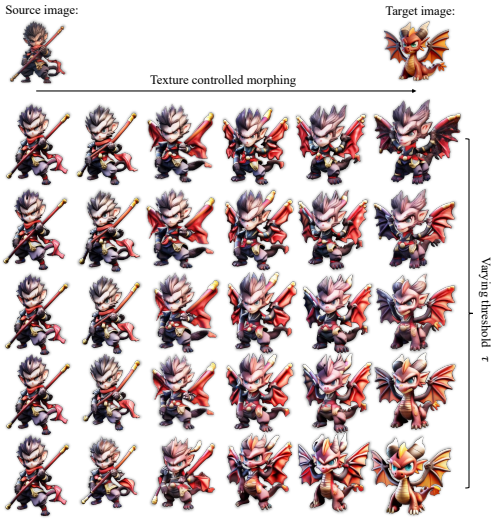

Figure 6: **Shape and texture interpolation results.** The figure shows shape interpolation arranged horizontally, with each row representing a morphing process at different interpolation thresholds, illustrating the smooth transition from the source image to the target image.

Figure 7: **Image-conditioned ablation study.** Top three rows compare textured 3D morphing: linear interpolation and no recursive initialization show shape and color artifacts, while ours is clean. Bottom two rows compare texture control: increasing source weight preserves details but distorts color; our selective interpolation maintains texture quality.

Comparison with existing methods We now compare our method with the current state-of-the-art in textured 3D morphing (3DRM (Yang et al., 2025)) in Fig. 5. Visually, our method delivers higher overall quality with significantly smoother transitions in both geometry and texture. Color variations in our morphing sequences also appear more continuous, and the interpolated shapes retain clear semantic coherence throughout. We further compare with MorphFlow, a recent textured 3D morphing approach, in the Appendix. See Appendix D for more visualizations and analysis.

Controllable texture morphing In Fig. 6, we demonstrate the controllability of WUKONG in textured 3D morphing through our proposed *selective texture interpolation*. Each row represents a

distinct morphing process with different interpolation thresholds, with the leftmost column fixed as the source shape. The results along the vertical axis (columns) are independent of each other. The far-left column does not show texture changes, as it represents the same 3D source generated from the input source image. By adjusting the selective interpolation threshold τ (from top to bottom), we can control the relative influence of source and target tokens during the morphing process. For instance, we can preserve source-dominant features such as facial identity, clothing styles, and color patterns in the intermediate 3D shapes, while still achieving smooth and semantically coherent shape transitions.

4.2 Ablation study

Ablation on shape interpolation In Fig. 7 (top three rows), we first present ablation studies evaluating different strategies for shape interpolation in image-conditioned 3D morphing. Specifically, (1) the first row shows results from directly applying linear interpolation to the condition tokens. While this produces smooth blending in token space, it results in semantically ambiguous intermediate shapes, incoherent textures, and noticeable color artifacts, revealing the limitations of naïve token-level averaging. (2) The second row presents our method using a linearly initialized free-support barycenter (i.e., linear initialization of support points). Although more flexible than fixed linear interpolation, it tends to converge to undesirable solutions, leading to distorted geometry and unstable textures in intermediate outputs. (3) In contrast, the third row shows our full method, where the barycenter is properly initialized and refined through iterative optimization. This results in high-quality morphs with semantically meaningful structure, smooth geometric transitions, and coherent color blending.

We further conduct ablation studies for shape interpolation under the setting of text-conditioned 3D morphing, as in Fig. 8. The results reveal the following: (1) Linear interpolation (row 1) produces unnatural artifacts, such as T-pose human figures, indicating that direct token blending often strays into semantically invalid regions. (2) Our method without recursive initialization (row 2) exhibits similar issues, underscoring the importance of proper initialization for barycenter optimization. Linear initialization fails to respect the underlying data manifold, often leading to unstable or artifact-prone results, as detailed in Sec. 3.3. (3) Our full method (row 3), with recursive initialization, successfully avoids these problems, delivering smooth and structural plausible shape transitions throughout the morph.

Figure 8: **Ablation study on text-conditioned 3D shape morphing.** Morphing from "T-rex" to "cat," the first row (linear interpolation) shows multiple T-pose human artifacts. The second row (without recursive initialization) still has some artifacts. The third row (full method) achieves smooth, coherent shape transitions.

In flow-based 3D generators, condition tokens guide the generation by modulating cross-attention at each layer. Linear interpolation between latent features explores intermediate regions of the latent space. However, when the source and target features are semantically distant (*e.g.*, from "T-Rex" to "cat" in Fig. 8) and lack appropriate supervision, the model's conditioning may become ambiguous or collapse into mode-averaged representations. This often results in "hallucinated" generic outputs, such as T-poses or default humanoid templates commonly seen in generative models. This issue highlights the importance of our barycentric interpolation and sequential initialization strategy, which

helps: (1) intermediate tokens remain on a semantically valid manifold; (2) abrupt jumps across unrelated modes are mitigated.

For quantatitive comparison, Table 2 provides further ablation results to validate the effectiveness of our method. The first row ("Linear") uses direct linear interpolation between latent tokens without any optimization or structural guidance. This naive baseline leads to unnatural intermediate shapes and visible texture artifacts. The second row ("w/o r-i") uses our barycenter-based optimization but removes the recursive initialization procedure. Although this improves upon linear interpolation by refining support points through optimization, the lack of a good initialization often causes the optimization to converge to suboptimal solutions, resulting in distortions and unstable transitions. The third row ("Ours*") corresponds to our full method applied on the TripoSG (Li et al., 2025) backbone. This configuration includes our recursive initialization scheme, which provides a strong prior for the barycenter and guides the optimization to more stable and meaningful intermediate representations. Compared to prior rows, "Ours*" produces smoother geometry transitions and better texture blending, even in challenging cases such as cross-category morphing. Notably, both "Ours*" and our default backbone implementation ("Ours") achieve the highest visual fidelity and semantic consistency across the morphing trajectory, demonstrating that our method generalizes well across different 3D generative models and significantly improves interpolation quality through careful design of both initialization and optimization procedures.

Table 2: **Ablation on shape interpolation.**

Model	FID ↓	STP-GPT ↑	SEP-GPT ↑	PPL ↓	V-CLIP ↑
Linear	14.73	0.86	0.85	3.06	0.81
w/o r-i	12.52	0.90	0.92	3.03	0.84
Ours*	4.16	0.96	**0.98**	2.95	**0.91**
Ours	**4.01**	**0.97**	0.95	**2.91**	0.90

Ablation on texture interpolation In Fig. 7 (rows 4 & 5), we demonstrate the importance of our selective interpolation strategy. In row 4, directly increasing source token weights via naïve linear interpolation leads to texture collapse, with disorganized colors and unclear semantics. In contrast, row 5 shows that our selective interpolation yields smooth morphing results while preserving distinctive source features like Wukong's appearance and attire. This demonstrates that selective control is crucial for high-quality and identity-preserving texture transitions.

Table 3: **Ablation on texture interpolation.**

Model	FID ↓	STP-GPT ↑	SEP-GPT ↑	PPL ↓	V-CLIP ↑
w/o interp	6.52	0.83	0.85	3.04	0.81
Linear	5.17	0.92	0.90	2.95	0.86
Ours	**4.20**	**1.00**	**1.00**	**2.91**	**0.90**

We conduct quantitative ablation experiments for texture interpolation strategy as shown in Table 3. The first row "w/o interp" refers to directly copying the source texture tokens across all steps without any interpolation. The second row "linear" refers to applying standard linear interpolation between source and target texture tokens. The third row "ours" corresponds to using our proposed selective interpolation strategy based on similarity thresholding. Our method achieve the highest visual fidelity and semantic consistency throughout the morphing trajectory, demonstrating that our method generalizes effectively across various 3D generative models. Additionally, it significantly enhances interpolation quality by carefully designing both the texture interpolation procedures.

5 Conclusion

We present a unified and flexible framework, WUKONG, for high-quality 3D morphing driven by minimal input—either in the form of image or text prompts. By leveraging a rectified flow-based generative model as a prior, our method enables semantically meaningful and structurally consistent shape and texture transitions. We reformulate the interpolation process using an optimal transport barycenter approach, and further enhance its stability and realism through a sequential initialization strategy. Additionally, our selective texture interpolation module offers fine-grained control over appearance, allowing users to preserve or blend semantic attributes as needed. Extensive experiments across diverse categories confirm the effectiveness of our design, with our method consistently outperforming prior state-of-the-art in both shape fidelity and texture consistency.

Acknowledgments This work is supported by Hong Kong Research Grant Council - General Research Fund (Grant No. 17213825). We would like to thank Tianshuo Yan for the invaluable help during the paper preparation.

References

Hadar Averbuch-Elor, Daniel Cohen-Or, and Johannes Kopf. Smooth image sequences for data-driven morphing. In *CGF*, 2016.

Thaddeus Beier and Shawn Neely. Feature-based image metamorphosis. *CG*, 1992.

Hang Dai, Nick Pears, William Smith, and Christian Duncan. Statistical modeling of craniofacial shape and texture. *IJCV*, 2020.

Soheil Darabi, Eli Shechtman, Connelly Barnes, Dan B Goldman, and Pradeep Sen. Image melding: Combining inconsistent images using patch-based synthesis. *TOG*, 2012.

Matt Deitke, Ruoshi Liu, Matthew Wallingford, Huong Ngo, Oscar Michel, Aditya Kusupati, Alan Fan, Christian Laforte, Vikram Voleti, Samir Yitzhak Gadre, et al. Objaverse-xl: A universe of 10m+ 3d objects. In *NeurIPS*, 2024.

Jiacheng Deng, Chuxin Wang, Jiahao Lu, Jianfeng He, Tianzhu Zhang, Jiyang Yu, and Zhe Zhang. Se-ornet: Self-ensembling orientation-aware network for unsupervised point cloud shape correspondence. In *CVPR*, 2023.

Marvin Eisenberger, Zorah Lahner, and Daniel Cremers. Smooth shells: Multi-scale shape registration with functional maps. In *CVPR*, 2020.

Marvin Eisenberger, David Novotny, Gael Kerchenbaum, Patrick Labatut, Natalia Neverova, Daniel Cremers, and Andrea Vedaldi. Neuromorph: Unsupervised shape interpolation and correspondence in one go. In *CVPR*, 2021.

Danielle Ezuz, Justin Solomon, and Mirela Ben-Chen. Reversible harmonic maps between discrete surfaces. *TOG*, 2019.

N. Fish, R. Zhang, L. Perry, D. Cohen-Or, E. Shechtman, and C. Barnes. Image morphing with perceptual constraints and STN alignment. In *CGF*, 2020.

William Gao, Noam Aigerman, Thibault Groueix, Vladimir G. Kim, and Rana Hanocka. TextDeformer: Geometry Manipulation using Text Guidance. In *SIGGRAPH*, 2023.

Yunhui Guo, Chaofeng Wang, Stella X Yu, Frank McKenna, and Kincho H Law. Adaln: a vision transformer for multidomain learning and predisaster building information extraction from images. *J Comput Civil Eng*, 2022.

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. In *NeurIPS*, 2017.

Ruqi Huang and Maks Ovsjanikov. Adjoint map representation for shape analysis and matching. In *CGF*, 2017.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.

Xiaojie Jin, Bowen Zhang, Weibo Gong, Kai Xu, Xueqing Deng, Peng Wang, Zhao Zhang, Xiaohui Shen, and Jiashi Feng. Mv-adapter: Multimodal video transfer learning for video text retrieval. In *CVPR*, 2024.

Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *CVPR*, 2019.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkuehler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *TOG*, 2023.

Hyunwoo Kim, Itai Lang, Noam Aigerman, Thibault Groueix, Vladimir G Kim, and Rana Hanocka. Meshup: Multi-target mesh deformation via blended score distillation. *arXiv preprint arXiv:2408.14899*, 2024.

Vladimir G Kim, Yaron Lipman, and Thomas Funkhouser. Blended intrinsic maps. *TOG*, 2011.

Yushi Lan, Shangchen Zhou, Zhaoyang Lyu, Fangzhou Hong, Shuai Yang, Bo Dai, Xingang Pan, and Chen Change Loy. GaussianAnything: Interactive Point Cloud Flow Matching For 3D Object Generation. In *ICLR*, 2025.

Yangguang Li, Zi-Xin Zou, Zexiang Liu, Dehu Wang, Yuan Liang, Zhipeng Yu, Xingchao Liu, Yuan-Chen Guo, Ding Liang, Wanli Ouyang, et al. Triposg: High-fidelity 3d shape synthesis using large-scale rectified flow models. *arXiv preprint arXiv:2502.06608*, 2025.

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. *TOG*, 2014.

Johannes von Lindheim. Simple approximative algorithms for free-support wasserstein barycenters. *COA*, 2023.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *ICLR*, 2023.

Warren S Loud. Differential equations. by an tikhonov, ab vasil'eva and ag sveshnikov. *Am Math Mon*, 1987.

Luma Labs AI. Luma dream machine: Ai-powered video content creation, 2025. Accessed: 2025-01-15.

Yiwei Ma, Guohai Xu, Xiaoshuai Sun, Ming Yan, Ji Zhang, and Rongrong Ji. X-clip: End-to-end multi-grained contrastive learning for video-text retrieval. In *ACM MM*, 2022.

Ricardo Martin-Brualla, Noha Radwan, Mehdi SM Sajjadi, Jonathan T Barron, Alexey Dosovitskiy, and Daniel Duckworth. Nerf in the wild: Neural radiance fields for unconstrained photo collections. In *CVPR*, 2021.

Oscar Michel, Roi Bar-On, Richard Liu, Sagie Benaim, and Rana Hanocka. Text2mesh: Text-driven neural stylization for meshes. In *CVPR*, 2022.

Luca Morreale, Noam Aigerman, Vladimir G. Kim, and Niloy J. Mitra. Neural Semantic Surface Maps. In *Eurographics*, 2024.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

Ben Poole, Ajay Jain, Jonathan T. Barron, and Ben Mildenhall. DreamFusion: Text-to-3D using 2D Diffusion. In *ICLR*, 2023.

He Qiyuan, Jinghao Wang, Ziwei Liu, and Angela Yao. Aid: Attention interpolation of text-to-image diffusion. *NeurIPS*, 2024.

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*, 2021.

Jing Ren, Simone Melzi, Maks Ovsjanikov, and Peter Wonka. Maptree: Recovering multiple solutions in the space of maps. *TOG*, 2020.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022.

Steven M Seitz and Charles R Dyer. View morphing. In *SIGGRAPH*, 1996.

Eli Shechtman, Alex Rav-Acha, Michal Irani, and Steve Seitz. Regenerative morphing. In *CVPR*, 2010.

Tianchang Shen, Jacob Munkberg, Jon Hasselgren, Kangxue Yin, Zian Wang, Wenzheng Chen, Zan Gojcic, Sanja Fidler, Nicholas Sharp, and Jun Gao. Flexible isosurface extraction for gradient-based mesh optimization. *TOG*, 2023.

Yichun Shi, Peng Wang, Jianglong Ye, Mai Long, Kejie Li, and Xiao Yang. MVDream: Multi-view Diffusion for 3D Generation. In *ICLR*, 2024.

Jaidev Shriram, Alex Trevithick, Lingjie Liu, and Ravi Ramamoorthi. RealmDreamer: Text-Driven 3D Scene Generation with Inpainting and Depth Diffusion. In *3DV*, 2025.

Justin Solomon, Fernando De Goes, Gabriel Peyré, Marco Cuturi, Adrian Butscher, Andy Nguyen, Tao Du, and Leonidas Guibas. Convolutional wasserstein distances: Efficient optimal transportation on geometric domains. *TOG*, 2015.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *ICLR*, 2021.

Olga Sorkine and Marc Alexa. As-rigid-as-possible surface modeling. In *SGP*, 2007.

Mingze Sun, Chen Guo, Puhua Jiang, Shiwei Mao, Yurun Chen, and Ruqi Huang. SRIF: Semantic shape registration empowered by diffusion-based image morphing and flow estimation. In *SIGGRAPH Asia*, 2024.

Gary K.L. Tam, Zhi Quan Cheng, Yu Kun Lai, Frank C. Langbein, Yonghuai Liu, David Marshall, Ralph R. Martin, Xian Fang Sun, and Paul L. Rosin. Registration of 3d point clouds and meshes: A survey from rigid to nonrigid. *TVCG*, 2013.

Maja Temerinac, Marco Reisert, and Hans Burkhardt. Shrec'07-protein retrieval challenge. *SMI*, 2007.

Dmitry Tochilkin, David Pankratz, Zexiang Liu, Zixuan Huang, Adam Letts, Yangguang Li, Ding Liang, Christian Laforte, Varun Jampani, and Yan-Pei Cao. Triposr: Fast 3d object reconstruction from a single image. *arXiv preprint arXiv:2403.02151*, 2024.

Chih-Jung Tsai, Cheng Sun, and Hwann-Tzong Chen. Multiview regenerative morphing with dual flows. In *ECCV*, 2022.

Jianfeng Xiang, Zelong Lv, Sicheng Xu, Yu Deng, Ruicheng Wang, Bowen Zhang, Dong Chen, Xin Tong, and Jiaolong Yang. Structured 3d latents for scalable and versatile 3d generation. *arXiv preprint arXiv:2412.01506*, 2024.

Songlin Yang, Yushi Lan, Honghua Chen, and Xingang Pan. Textured 3d regenerative morphing with 3d diffusion prior. *arXiv preprint arXiv:2502.14316*, 2025.

Wangbo Yu, Jinbo Xing, Li Yuan, Wenbo Hu, Xiaoyu Li, Zhipeng Huang, Xiangjun Gao, Tien-Tsin Wong, Ying Shan, and Yonghong Tian. Viewcrafter: Taming video diffusion models for high-fidelity novel view synthesis. *arXiv preprint arXiv:2409.02048*, 2024.

Mehmet Ersin Yumer, Siddhartha Chaudhuriy, Jessica K. Hodgins, and Levent Burak Kara. Semantic shape editing using deformation handles. *TOG*, 2015.

LAN Yushi, Shangchen Zhou, Zhaoyang Lyu, Fangzhou Hong, Shuai Yang, Bo Dai, Xingang Pan, and Chen Change Loy. Gaussiananything: Interactive point cloud flow matching for 3d generation. In *ICLR*, 2025.

Xiaohui Zeng, Arash Vahdat, Francis Williams, Zan Gojcic, Or Litany, Sanja Fidler, and Karsten Kreis. Lion: Latent point diffusion models for 3d shape generation. In *NeurIPS*, 2022.

Biao Zhang and Rico Sennrich. Root mean square layer normalization. In *NeurIPS*, 2019.

Kaiwen Zhang, Yifan Zhou, Xudong Xu, Xingang Pan, and Bo Dai. DiffMorpher: Unleashing the Capability of Diffusion Models for Image Morphing. In *CVPR*, 2024a.

Zhengbo Zhang, Li Xu, Duo Peng, Hossein Rahmani, and Jun Liu. Diff-Tracker: Text-to-Image Diffusion Models are Unsupervised Trackers. In *ECCV*, 2024b.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction clearly summarize the main contributions of the paper, accurately reflecting both the scope and the results presented.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The paper explicitly discusses the limitations of the proposed work. Although the main manuscript focuses on the core contributions, a dedicated section on limitations is provided in the supplementary material.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: The paper explicitly states all theoretical results and proofs.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The paper fully details experimental setups, datasets, hyperparameters, evaluation metrics, and procedures (including model architecture descriptions or replication instructions where applicable), enabling reproduction of results without relying on unreleased code/data.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: The paper provides publicly accessible links to code, along with detailed step-by-step instructions (including exact commands, environment specifications, and data preparation workflows)

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: The paper specify all training and test details.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: The paper reports error bars suitably and correctly.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The paper provide sufficient information on the computer resources.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: The research conducted in the paper conform with the NeurIPS Code of Ethics in every respect.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: The work focuses on foundational 3D generation research without direct applications to societal contexts, and there are no foreseeable direct paths to negative societal impacts such as disinformation or privacy risks, hence no broader impacts discussion is applicable.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.

- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The work focuses on 3D generation research using synthetic datasets, and thus does not involve releasing data or models with significant potential for misuse, making safeguards discussions inapplicable.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: Original code, data, and model owners are appropriately credited, with licenses and terms of use clearly stated and fully respected.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.

- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: New assets are well documented. The documentation includes details about training, license, and limitations.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The research does not involve crowdsourcing experiments or studies with human subjects, focusing solely on 3D generation methodology without participant interaction, thus the question is not applicable.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve research with human subjects or crowdsourcing, and thus does not require IRB approvals or discussions of participant risks.

    Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (https://neurips.cc/Conferences/2025/LLM) for what should or should not be described.

Wukong's 72 Transformations: High-fidelity Textured 3D Morphing via Flow Models
– *Appendix* –

A Model details

To enable high-fidelity textured 3D morphing, we build upon two pretrained flow-based transformer models introduced in Trellis (Xiang et al., 2024): structure flow model and SLat flow model, originally designed for unconditional 3D generation. The structure flow model operates on structured latent representation and follows a transformer-based architecture with 24 modulated transformer blocks with cross attentions. Each block contains self-attention, cross-attention, and feed-forward components, modulated via AdaLN (Guo et al., 2022) conditioning from a learned timestep embedding. Root Mean Square Normalization (RMSNorm) (Zhang and Sennrich, 2019) is applied to both the query and key representations prior to their use in the attention mechanism. The SLat flow model incorporates a hierarchical design with sparse 3D convolutional blocks and positional embeddings to encode spatial context. The transformer core comprises 24 modulated sparse transformer blocks with cross attentions, analogous in structure to the geometry model but enhanced with sparse attention and feed-forward operations. Additionally, the model includes dedicated output convolutional blocks for upsampling and decoding, ensuring fine-grained preservation and modulation of high-frequency texture details. We use the free-support Wasserstein barycenter solver from the POT library (`ot.lp.free_support_barycenter` (Lindheim, 2023)), which is based on linear programming (LP). The cost matrix is computed using the squared Euclidean distance in the CLIP (Radford et al., 2021) (for text) and DINOv2 (Oquab et al., 2023) (for image) embedding space.

B Ablation study

To ensure a fair, apples-to-apples comparison with 3DRM, we implemented our full morphing method on top of the GaussianAnything (Yushi et al., 2025) framework—the same 3D generator used in 3DRM (Yang et al., 2025), results are shown in Table 4. We denote this variant as "Ours*" in the table below. Across all evaluation metrics, including FID, PPL, V-CLIP, and GPT-based perceptual scores (STP-GPT, SEP-GPT), our method consistently outperforms 3DRM, even when both share the exact same backbone. This clearly demonstrates that the improvement is not solely due to the use of a stronger generator like Trellis, but rather stems from our core morphing algorithm. Note that the GPT-based results may differ from those presented in main paper, as they are computed using only the four methods reported here.

Table 4: **Quantitative comparison with GaussianAnything as backbone.**

Model	FID ↓	STP-GPT ↑	SEP-GPT ↑	PPL ↓	V-CLIP ↑
MorphFlow	147.70	0.38	0.41	3.10	0.78
3DRM	6.36	0.85	0.80	3.02	0.84
Ours*	5.15	0.93	0.91	2.94	0.87
Ours	**4.01**	**1.00**	**1.00**	**2.91**	**0.90**

Besides, We conduct quantitative evaluations with different threshold τ values and present the results below as shown in Table 5. We observe that the performance is robust across a reasonable range of thresholds (0.2-0.8). We set a default threshold 0.3 in our evaluation.

Table 5: **Ablation study on threshold τ.**

Threshold τ	0.2	0.3	0.4	0.6	0.8
FID ↓	4.54	4.20	4.17	4.25	4.49
PPL ↓	2.94	2.91	2.91	2.92	2.93
V-CLIP ↑	0.88	0.90	0.91	0.90	0.87

To evaluate the method's generalization to real 3D data, we conducted experiments using the Headspace dataset (Dai et al., 2020), which contains high-quality 3D face scans along with corresponding rendered RGB images. In our pipeline, we used these rendered images as inputs and passed them through the DINOv2 (Oquab et al., 2023) encoder to extract texture and semantic features for morphing. The outputs were generated by our standard pipeline without any architectural changes or fine-tuning. Despite relying on pretrained components, our method shows strong generalization to real-world 3D scans. Our method outperformed both MorphFlow (Tsai et al., 2022) and 3DRM (Our own implementation) (Yang et al., 2025) on the same evaluation protocol. Quantitative results are shown below in Table 6:

Table 6: **Quantitative results on Headspace dataset.**

Model	FID ↓	STP-GPT ↑	SEP-GPT ↑	PPL ↓	V-CLIP ↑
MorphFlow	95.24	0.53	0.47	3.22	0.84
3DRM	6.61	0.83	0.77	3.04	0.88
Ours	**3.97**	**1.00**	**1.00**	**2.88**	**0.96**

C Rectified flow models *vs.* diffusion models

Continuity The first reason we choose the flow model over the diffusion model for 3D morphing is its mathematically grounded continuity with respect to the interpolation parameter α. In flow-based generative models, the mapping from α to the output $F(\alpha)$ is deterministic and constructed via an invertible transformation $T(z;\alpha)$, typically defined by an ordinary differential equation (ODE). Under standard regularity conditions (e.g., Lipschitz continuity of the velocity field), the solution $T(z;\alpha)$ is guaranteed to be continuously differentiable with respect to α (Loud, 1987), ensuring that the morphing trajectory forms a smooth path in the output space. This deterministic nature makes it possible to precisely control intermediate shapes and textures, yielding consistent and artifact-free transitions.

In contrast, diffusion models are typically governed by stochastic differential equations (SDEs), which introduce randomness throughout the generative process. While deterministic sampling methods like DDIM (Song et al., 2021) exist and are widely used, the underlying denoising process in these models often follows a stochastic trajectory. Consequently, even with interpolated conditioning, the same value of α can yield different outputs across runs. This inherent variability makes it difficult to ensure continuity or precise control in the morphing sequence, particularly at intermediate points where uncertainties compound. In contrast, rectified flow models generate deterministic and unique interpolation paths, enabling forward integration with a guaranteed likelihood formulation. This property makes them more suitable for achieving smooth, stable, and controllable interpolation, which aligns with our need for consistency in the latent space.

Convexity and optimality There are theoretical guarantees for flow models like the rectified flow model in maintaining the convexity of data during the generation process. This linearity ensures that any intermediate sample lies within the convex hull of the endpoints, thereby preserving the convexity of the data. In practice, the trajectory is hard to remain straight. There is analysis (Liu et al., 2023) on the straightness error on the trajectory, which states that even an imperfect trajectory is close enough to straight lines and ensure convexity of data to some extent. Theorems in this analysis further emphasize the uniqueness and optimality of the solution of rectified flow in matching distributions under convex cost functions. For diffusion models, the backward process generates data from the prior but does not theoretically guarantee convexity preservation. These models focus on matching data distributions, not preserving geometric properties like convexity. There is no theatrical guarantee on the data convexity in the backward process. The noise term in the reverse-time SDE can easily violate the convexity of original data. Also, under the same condition as the rectified flow model, the path of diffusion models is not assured to be optimal. There exists certain crossing flows in the matching of two distributions, leading to features that are out-of-distribution in practice.

Inference speed The theoretical basis for the faster inference of flow models (such as rectified flow models) primarily stems from the geometric properties of their trajectories and the efficiency of numerical simulation. For rectified flow model used in this paper, it aims to make generation

Table 7: **Quantitative comparison with shape morphing methods.**

Metrics	MapTree	BIM	SmoothShells	NeuroMorph	SRIF	Ours
Dirichlet ↓	17.7309	12.4723	14.0198	22.0461	6.4702	**4.5163**
Cov. ↑	0.3967	0.4665	0.6275	0.1099	0.6418	**0.8510**

trajectories as straight as possible. For ideal straight-line flows, the trajectory between any two points $Z_0 \sim \pi_0$ and $Z_1 \sim \pi_1$ is given by the linear interpolation $Z_t = tZ_1 + (1-t)Z_0$. In this case, the drift field of the ODE is a constant $v(Z_t, t) = Z_1 - Z_0$, which can be solved exactly with a single Euler step: $Z_1 = Z_0 + v(Z_0, 0) \cdot 1$. This eliminates the need for time discretization errors. Even in non-ideal cases, the optimized trajectories are close to straight lines, significantly reducing the number of required steps (in this paper, we take 20 steps). Diffusion models use nonlinear, stochastic trajectories requiring many steps—typically 2,000 without sampling techniques or around 200 with them—to achieve good results. Also, the reverse SDE process requires noise sampling, leading to additional computation cost.

D Comparison with other methods

D.1 Comparison with other textured 3D method

We compare our method with existing textured 3D morphing approaches, including MorphFlow, 3DRM, and our own. While the main paper presents qualitative comparisons with 3DRM, here we additionally provide a side-by-side qualitative comparison with MorphFlow. As shown in the Fig. 9, rows 1 and 3 display results from MorphFlow, and rows 2 and 4 show our corresponding outputs. Our method produces noticeably clearer and more accurate shapes and textures, with smoother and more coherent morphing transitions. These visual improvements are consistent with the quantitative results reported in the main paper, further corroborating the effectiveness of our approach.

Figure 9: **Qualitative comparison with MorphFlow.**

D.2 Comparison with shape morphing methods

Although previous 3D shape morphing methods do not consider texture transformation, we provide a comparison focused solely on shape deformation. As shown in Table 7, we compare our method with several state-of-the-art shape morphing approaches, including MapTree (Ren et al., 2020), BIM (Kim et al., 2011), SmoothShells (Eisenberger et al., 2020), NeuroMorph (Eisenberger et al., 2021), and SRIF (Sun et al., 2024). Following the evaluation protocol from (Sun et al., 2024), we use the SHREC07 (Temerinac et al., 2007) dataset and report performance using Dirichlet energy (Ezuz et al., 2019) and Coverage (Huang and Ovsjanikov, 2017) metrics. Our method achieves superior results across both metrics, demonstrating more efficient and accurate shape interpolation.

For qualitative comparison, we show results against SRIF in Fig. 10, where rows 1, 3 and 5 show SRIF's outputs and rows 2, 4 and 6 show ours. Our morphing process is smoother and preserves finer details in intermediate shapes—for example, the gecko's toes in row 2 and the head structure in row 4. Additional comparison with NeuralMorph is presented in Fig. 11, where our results are again significantly more detailed and coherent.

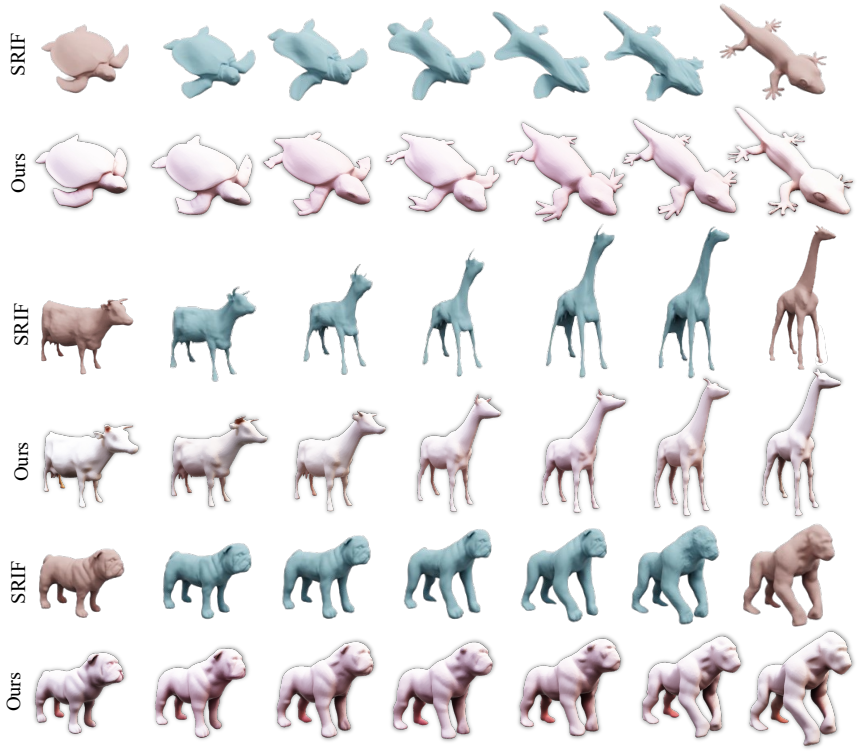

Figure 10: **Qualitative comparison with SRIF.**

Figure 11: **Qualitative comparison with NeuralMorph.**

To ensure that intermediate shapes remain faithful to both source and target, we introduce a shape-aware initialization. Specifically, we render both front and back views of the source and target objects and use these images as inputs to the flow model to extract an initial condition feature. This feature is then refined by minimizing the geometric difference between generated shapes and the original meshes, leading to accurate and consistent 3D representations throughout the morphing sequence.

E More results

Fig. 12 and Fig. 13 illustrate the textured 3D morphing process generated from "Wukong" to a variety of objects. The texture can be flexibly inherited from either the source or the target image, depending on the user preference. Fig. 14 and Fig. 15 further demonstrate morphing between additional object pairs, showcasing the versatility of our method. Notably, our method is capable of performing textured 3D morphing not only between geometrically complex objects, but also across different semantic categories, highlighting its superior robustness and generalizability.

F Broader impact

Our work introduces WUKONG , a training-free framework for high-quality textured 3D morphing, which significantly lowers the barrier to creating detailed and semantically consistent 3D transformations from simple prompts. This greatly reduces the efforts on 3D content creation for artists, designers, and educators, enabling broader access to advanced generative tools without requiring technical expertise in 3D modeling or animation. The ability to produce controllable and high-fidelity morphing sequences could benefit applications in virtual reality, digital storytelling, education, and creative industries. We hope our work inspires further research into controllable and efficient 3D generation techniques, and that it serves as a foundation for inclusive and creative applications of generative 3D content.

G Limitation

While our method achieves state-of-the-art performance in textured 3D morphing, several limitations remain. First, like existing morphing methods, our approach still encounters difficulties on cases involving extreme topological changes, such as splitting or merging parts. These scenarios remain a general challenge in the field and are not yet fully addressed by existing methods. Second, since our method operates without explicit 3D supervision or correspondence annotations, its results may be sensitive to ambiguities in the input prompts or inconsistencies in multi-view generation, especially when the input lacks structural clarity.

Figure 12: **Textured 3D morphing of Wukong (The Monkey King).**

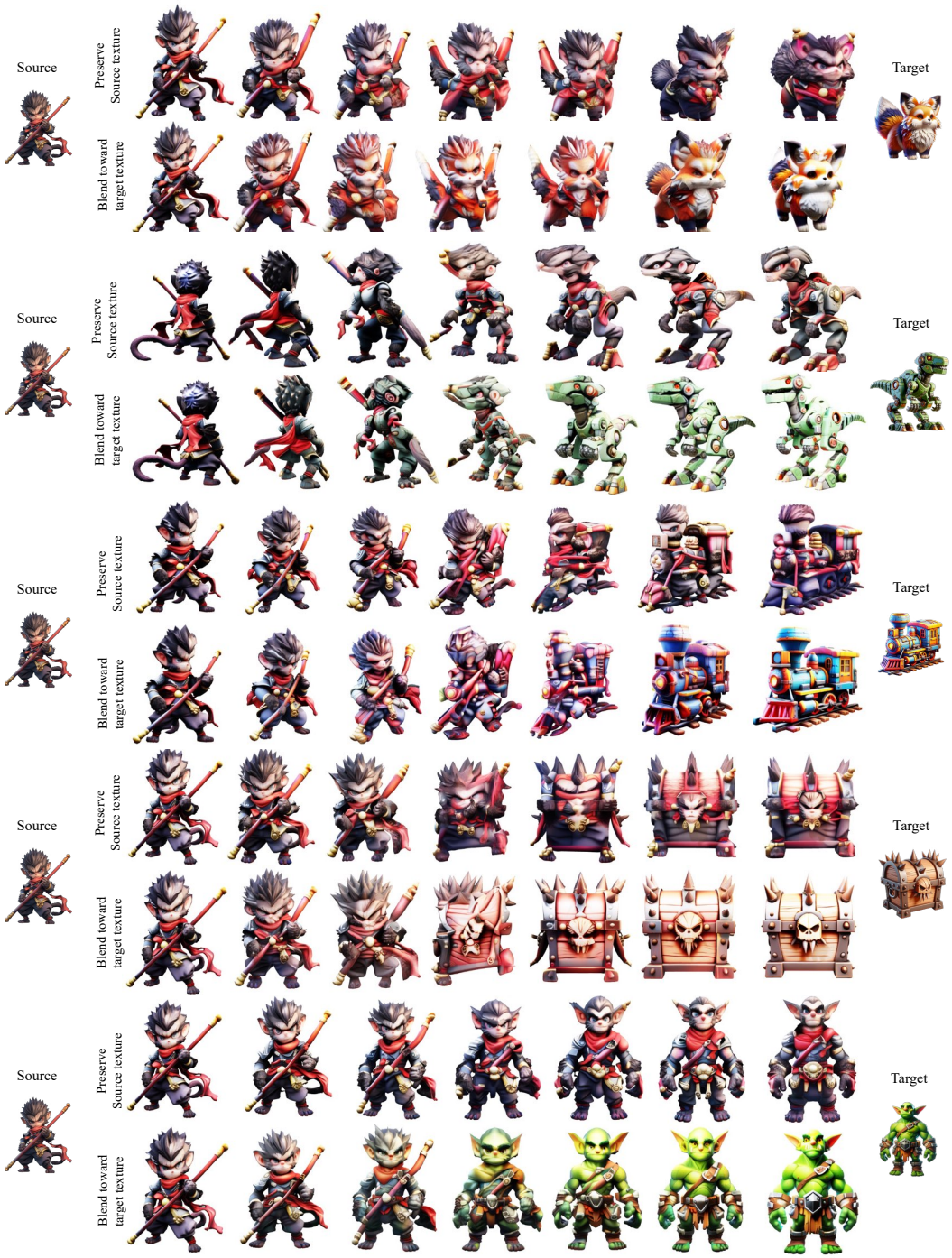

Figure 13: **Textured 3D morphing of Wukong (The Monkey King).**

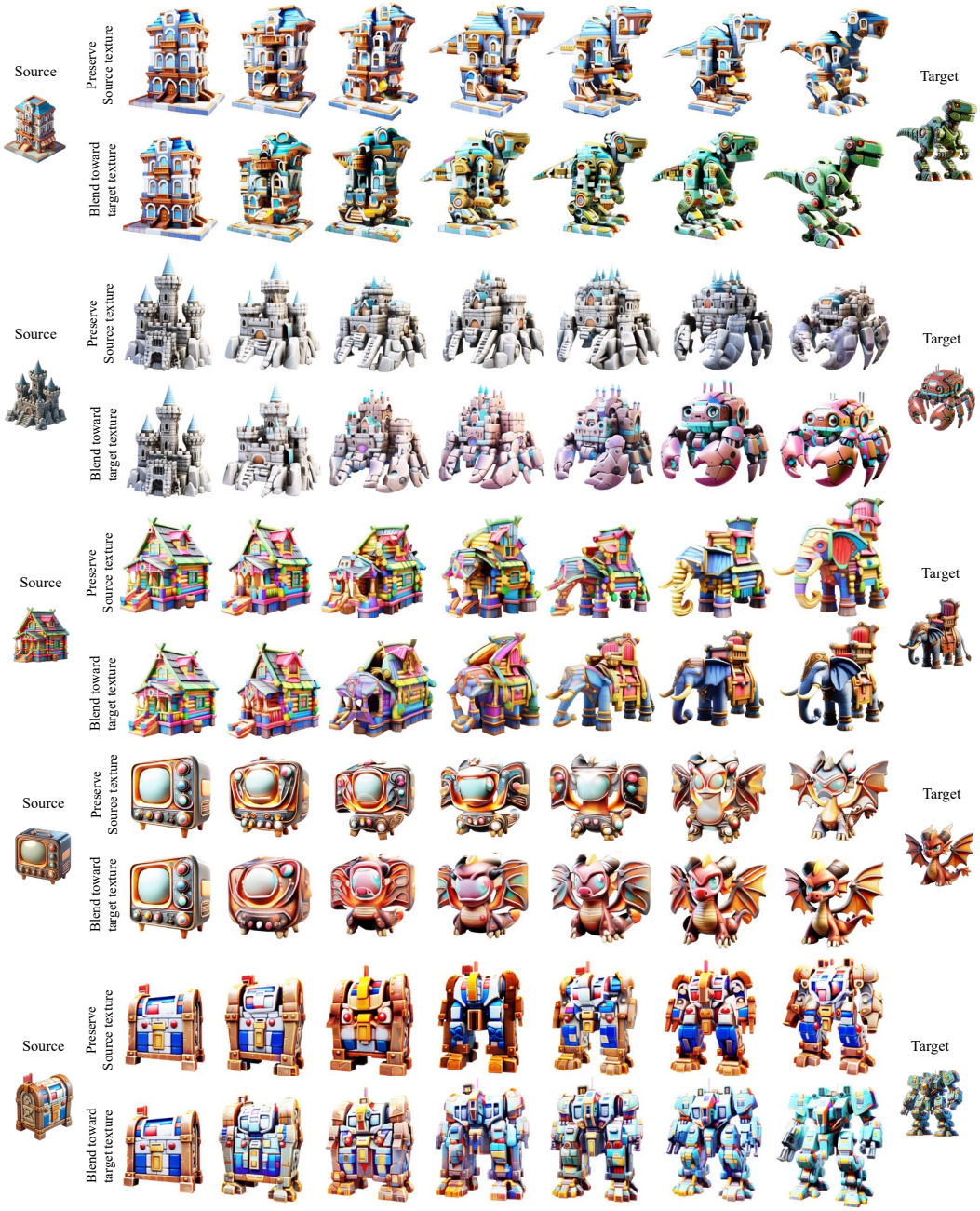

Figure 14: **Textured 3D morphing of different objects.**

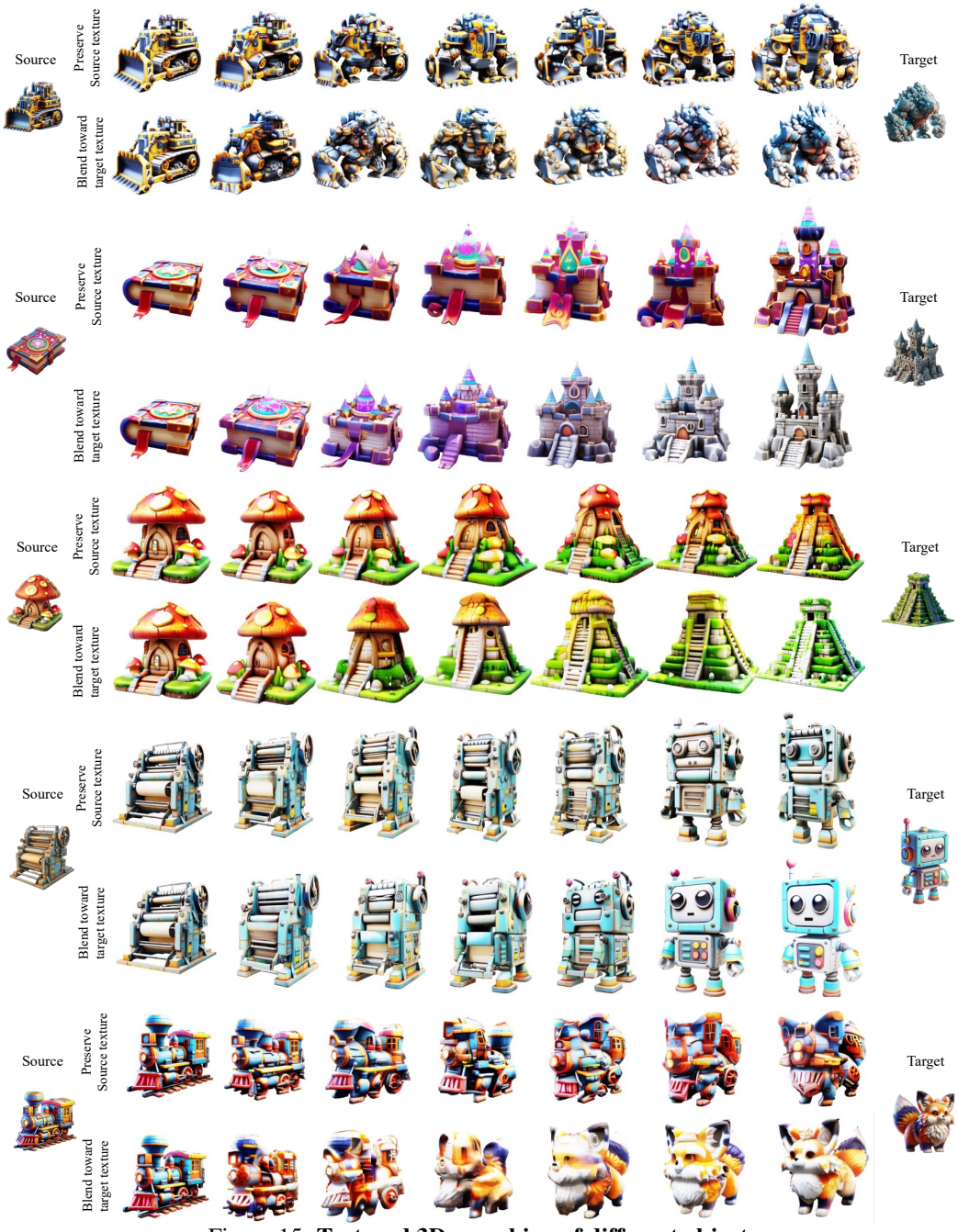

Figure 15: **Textured 3D morphing of different objects.**