Supplementary Materials
Textured 3D Regenerative Morphing with 3D Diffusion Prior

Songlin Yang, Yushi Lan, Honghua Chen, Xingang Pan
S-Lab, Nanyang Technological University
sl.yang888@outlook.com, {yushi001, honghua.chen, xingang.pan}@ntu.edu.sg

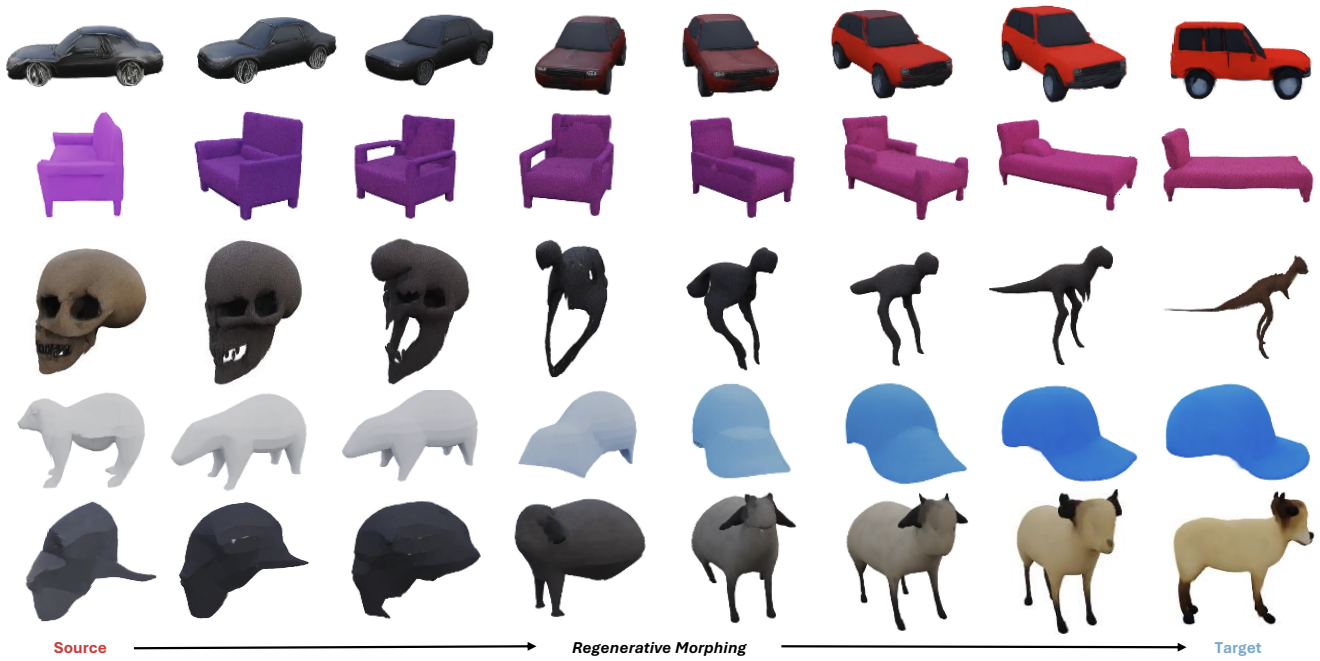

Figure 1. More 3D morphing sequences generated by our method. ***More video results can be found here.***

To experimentally validate the rationale behind the motivations discussed in our manuscript and to provide additional details that could not be elaborated on due to manuscript space limitations, we have carefully prepared comprehensive supplementary materials for reference (***Click on the index to directly access the corresponding content***).

1. More Qualitative Experimental Results

To better demonstrate the effectiveness of our method, we provide additional results in Fig. 1, Fig. 2, and here (video results for 360° display). Our method can handle object pairs with significant differences in shape structure and texture, providing a novel research direction for future exploration of more complex objects and adaptive morphing

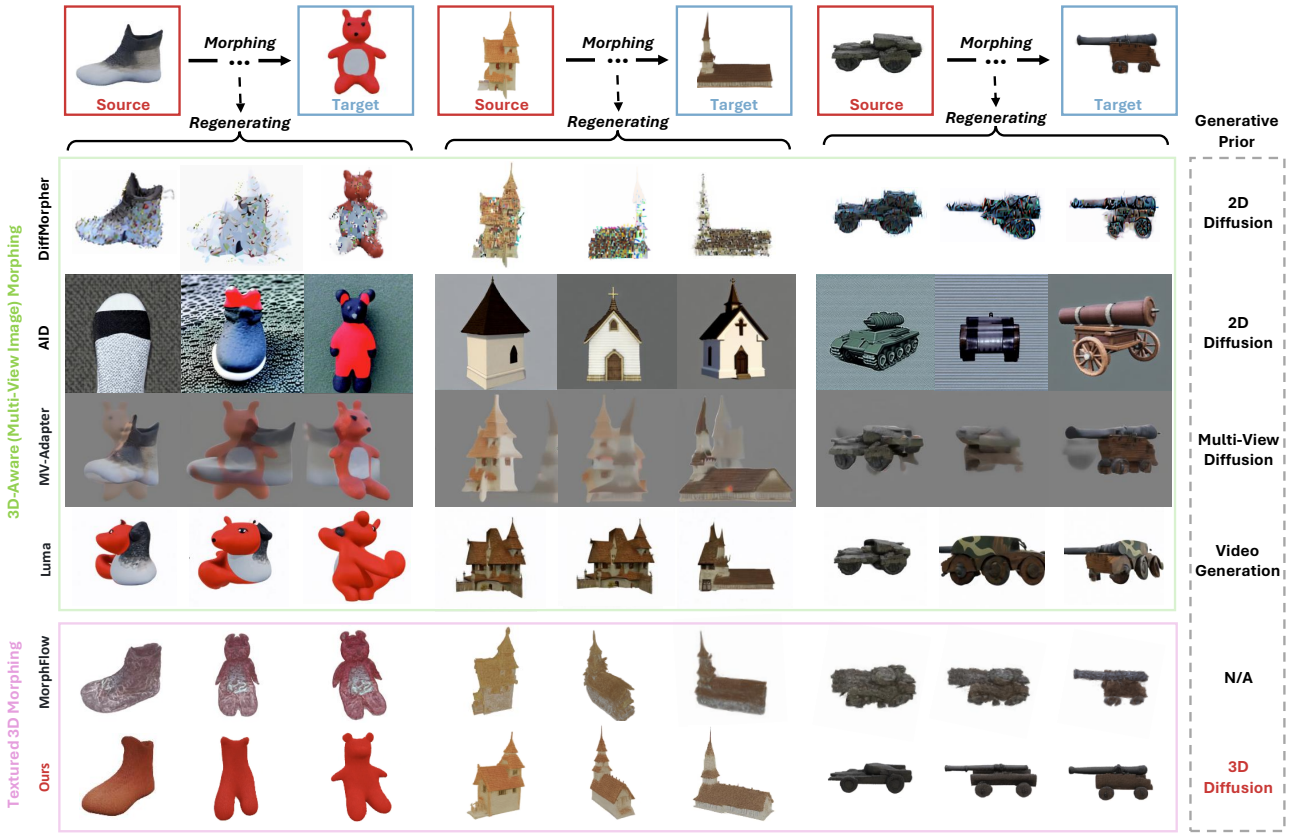

Figure 2. More qualitative comparisons of different methods from tasks, morphing tricks, and generative priors. ***More video results can be found here.***

methods.

2. 3D Generation Model: Gaussian Anything

Gaussian Anything [13] introduces a 3D generation framework built on a point cloud-based 3D latent space. The 3D Variational Autoencoder (VAE) (See 2.1) efficiently encodes 3D data into a dynamic latent space, which is subsequently decoded into detailed Surfel Gaussians. Diffusion models (See 2.2) trained on this compacted latent space achieve remarkable results in 3D generation and editing conditioned on text, as well as in generating high-quality 3D content from images on diverse real-world datasets. For more implementation details, please see their project page.

2.1. Point-Cloud Structured 3D VAE

A 3D VAE is introduced that takes multi-view posed RGB-D (Depth)-Normal renderings as input. These renderings are easy to generate and provide a rich set of 3D attributes corresponding to the input object. Each view's information is concatenated along the channel dimension and efficiently encoded using a scene representation transformer [17], producing a compact latent representation of the 3D input. Rather than directly applying this latent representation to diffusion learning, the model's innovative method transforms unordered tokens into a shape that mirrors the 3D input. This transformation is achieved by cross-attending [11] the latent set with a sparse point cloud sampled from the 3D shape. This point-cloud structured latent space significantly aids in disentangling shape and texture, as well as enabling 3D editing. Subsequently, a DiT-based 3D decoder [14] progressively decodes and upscales the latent point cloud into a dense set of Surfel Gaussians [9], which are rasterized into high-resolution renderings to guide the 3D VAE training.

2.2. Cascaded 3D Generation with Flow Matching

After the 3D VAE is trained, they conduct cascaded latent diffusion modeling on the latent space through flow matching [1] using the DiT [14] framework. To encourage better shape-texture disentanglement, a point cloud diffusion model is first trained to carve the overall layout of the input shape. Then, a point cloud feature diffusion model is cascaded to output the corresponding feature conditioned on the generated point cloud. The generated featured point cloud is then decoded into Surfel Gaussians [9] via pre-

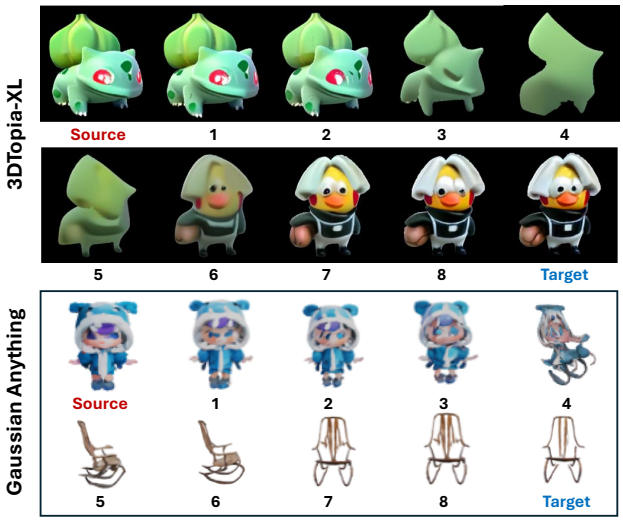

Figure 3. Evaluation of 3D generative model capabilities. Based on accessibility, we tested the interpolation performance of projects with available training code and model details, namely 3DTopia-XL [3] and Gaussian Anything [13] in the image-to-3D setting. We found that while 3DTopia-XL generates high-quality 3D assets, its latent space lacks reasonable generative capabilities, as evidenced by the interpolation results between the 3rd and 6th samples.

trained VAE for downstream applications.

2.3. Implementation Details

The 3D diffusion prior [13] is trained on the G-Objaverse [4, 15] dataset. Its geometry and texture diffusion models are based on the DiT architecture [2], which consists of 24 layers, 16 attention heads, and a 1024-dimensional hidden space. The sparse point cloud $\mathbf{z}_G$ has a size of $M \times 3$ (with $M = 768$), and the corresponding feature $\mathbf{z}_T$ has dimensions $M \times 10$.

3. How to Align/Prepare the Input 3D and Images with Gaussian Anything?

For textured 3D representations, multi-view RGB, depth, and normal images can be directly rendered, and then the corresponding latent can be obtained using the 3D VAE of Gaussian Anything. For a single image, two methods are possible: (a) The 2D image can be lifted to multi-view using a multi-view generation model [18], and then a renderable textured 3D model can be trained from these multi-view images, or (b) A direct image-to-3D method [10] can be used to obtain the textured 3D model.

4. How to Choose Appropriate 3D Diffusion Models for 3D Morphing?

Selecting an appropriate 3D generative model is foundational for texured 3D regenerative morphing, as it determines (a) the range of 3D object categories that can be handled and (b) the ability to integrate diverse information for generating smooth interpolation sequences. We followed four criteria when selecting a 3D generative model for our research:

(a) Accessibility: Training 3D generative models is highly resource-intensive, and high-quality models capable of generating diverse outputs are often proprietary assets, accessible only through APIs. This limits our ability to probe the internal characteristics and potential issues of such models. Therefore, an ideal 3D generative model should be open-sourced, including all testing and training files, datasets, and model checkpoints. Based on this criterion, we selected Gaussian Anything [13] and 3DTopia-XL [3] as our potential target models. It is worth noting that, prior to the ICCV deadline, the most anticipated text-to-3D models—Trellis [23] and CLAY [26]—have not been open-sourced, preventing us from conducting experiments on them.

(b) Fairness: For data-driven studies, fairness is reflected in the use of publicly available datasets for training, ensuring future researchers can build on our work with a well-established baseline or benchmark. The Objaverse [4] dataset, currently one of the most widely adopted 3D datasets, is particularly suitable for academic research. Thus, we prefer Gaussian Anything [13], which is trained on Objaverse.

(c) Generation Quality: 3D generative modeling has become one of the most competitive and rapidly evolving research areas in recent years, with many papers showcasing impressive results. However, unlike 2D images or videos, 3D training data is harder to collect at scale, and no existing model can perfectly generate a full range of 3D objects. Therefore, we prioritized models capable of generating a wide variety of objects, ideally including categories such as animals, buildings, furniture, food, transportation, and plants. After evaluating the performance of several state-of-the-art 3D generative models on image-to-3D and text-to-3D tasks, we selected Gaussian Anything [13] as our research model.

(d) Preliminary Interpolation Feasibility: For models with strong generative capabilities, the quickest way to determine their suitability for 3D morphing research is to conduct basic interpolation tests. Not all generative models can effectively fuse different information for interpolation. Among the tested models (as shown in Fig. 3), Gaussian Anything [13] demonstrated the best interpolation performance, making it the most suitable choice for our study.

5. Baseline Methods and Implementation Details

5.1. DiffMorpher

Given two images, DiffMorpher [25] uses two LoRAs [8] to fit the two images respectively. Then the latent noises for the two images are obtained via DDIM inversion [19]. The mean and standard deviation of the interpolated noises are adjusted through AdaIN. To generate an intermediate image, they interpolate between both the LoRA parameters and the latent noises via the interpolation ratio α. In addition, the text embedding and the K and V in self-attention modules are also replaced with the interpolation between the corresponding components. Using a sequence of α and a new sampling schedule, their method will produce a series of high-fidelity images depicting a smooth transition between the two given images. We followed the script and default parameter settings given by Diffmorpher and used their open-source code to produce the results.

5.2. AID

Similar to the DiffMorpher [25] framework, AID [7] removes the LoRA fitting and introduces the following additional modifications: (a) Replacing both cross-attention and self-attention mechanisms during interpolated image generation with fused interpolated attention; (b) Selecting interpolation coefficients using a Beta prior; (c) Injecting prompt guidance into the fused interpolated cross-attention. We implemented the generation of relevant results based on the code of Stable Diffusion 1.5 [16], and all settings follow the default settings of AID. More details can be found on their project page.

5.3. MV-Adapter

MV-Adapter [10] is a versatile plug-and-play and state-of-the-art adapter that turns existing pre-trained text-to-image (T2I) diffusion models to multi-view image generators. We generated image morphing results based on their Image-to-Multiview code and Stable Diffusion 2.1. The only change is that we linearly interpolated the condition features of the source image and target image extracted by their image encoder according to different morphing weights.

5.4. Luma

The Dream Machine of Luma AI is based on the DiT [14] video generation architecture, capable of generating high-quality videos with 120 frames in just 120 seconds, enabling rapid creative iteration. It understands physical interactions, ensuring that the generated video characters and scenes maintain consistency and physical accuracy. We accessed their API and utilized the video generation function to generate intermediate video frames by providing the source image as the first frame and the target image as the last frame. For instance, for the "polar bear" to "wooden stool" morphing video generation, the guiding prompt we used is: *"Morph a polar bear into a wooden stool, smoothly interpolating both geometry and texture, with the object always remaining at the center of the frame."*

Table 1. Task setting comparison.

	Shape	Texture	Aligned Dataset	Out-of-Domain
MeshUp	✓	×	No Need	✓
SRIF	✓	×	No Need	✓
NeuroMorph	✓	×	Need	×
CharacterMixer	✓	×	Need	×
MorphFlow	✓	✓	No Need	✓
Ours	✓	✓	No Need	✓

5.5. MorphFlow

MorphFlow [22] introduces an optimization-based method for multi-view regenerative morphing. The method does not assume prior knowledge of the categories or affinities between the source and target images, nor does it rely on predefined correspondences. By utilizing optimal transport, the method interpolates a volume for rendering smooth multi-view transitions. Additionally, a rigid transformation is incorporated to preserve structure during the morphing process. The method is highly efficient, learning and rendering a morphing renderer from scratch in just 30 minutes, with the ability to generate a novel-view morph per second during morphing and rendering. We first obtain the multi-view images along with their corresponding COLMAP camera annotations, and then generate the morphing output using their open-source code with the default parameter settings.

6. Related Works That You Might Confuse: Comparisons with Shape Morphing Methods and Out-of-Setting Clarification

As shown in Tab. 1, our setting focuses on textured 3D morphing, a task currently shared only with MorphFlow [22]. Earlier works like NeuroMorph [5] and CharacterMixer [24] were trained on aligned datasets, essentially learning in-domain, topology-aligned correspondences between 3D data. However, these methods fail to generalize such correspondences to out-of-domain 3D representations. Other methods, such as MeshUp [12] and SRIF [20], explored shape morphing by leveraging generative priors. While they recognized the importance of generative priors for improving generalization in morphing tasks, their work was limited to shape-only morphing and did not release source code. We qualitatively compared results from their manuscripts with ours, as shown in Fig. 4, demonstrating that our method not only performs morphing between textured 3D representations with similar topologies (e.g., Mario and Luigi) but also handles morphing between representations with significant category differences (e.g., a boot and a teddy bear).

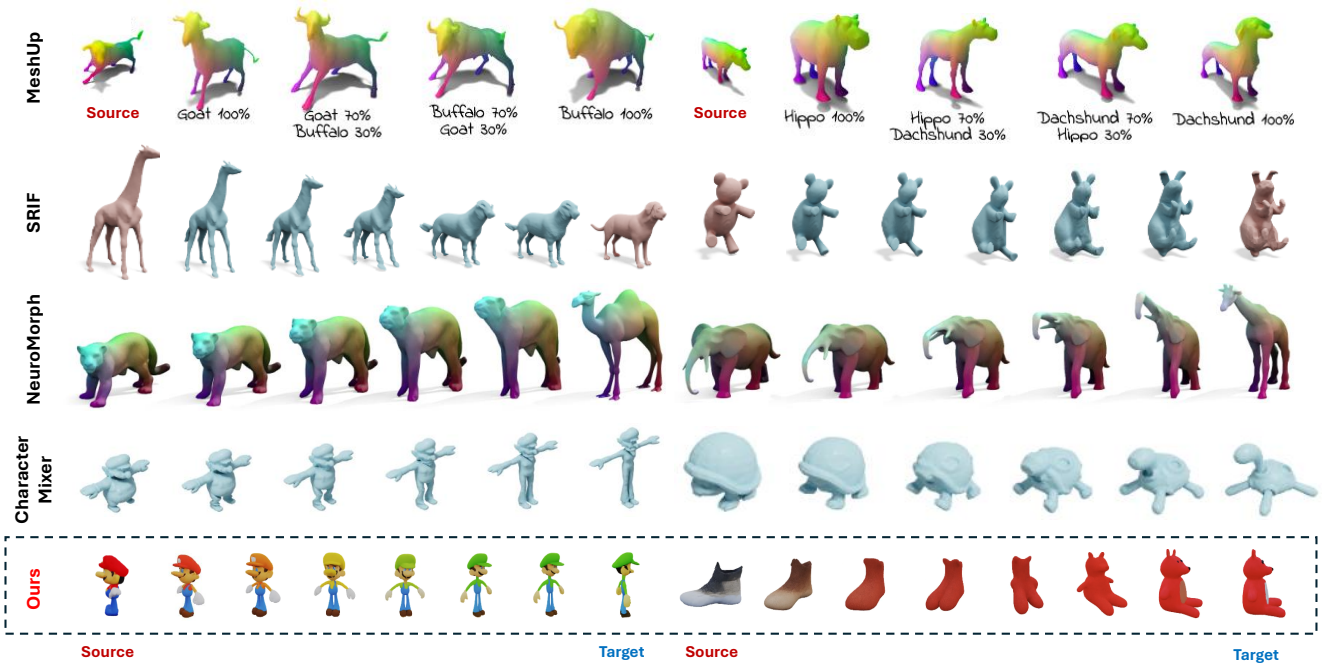

Figure 4. Comparison of related methods. Our method focuses on textured 3D morphing, whereas MeshUp [12], SRIF [20], NeuroMorph [5], and CharacterMixer [24] are limited to shape-only 3D morphing. Note that all results outside the dashed boxes are sourced from their respective manuscripts.

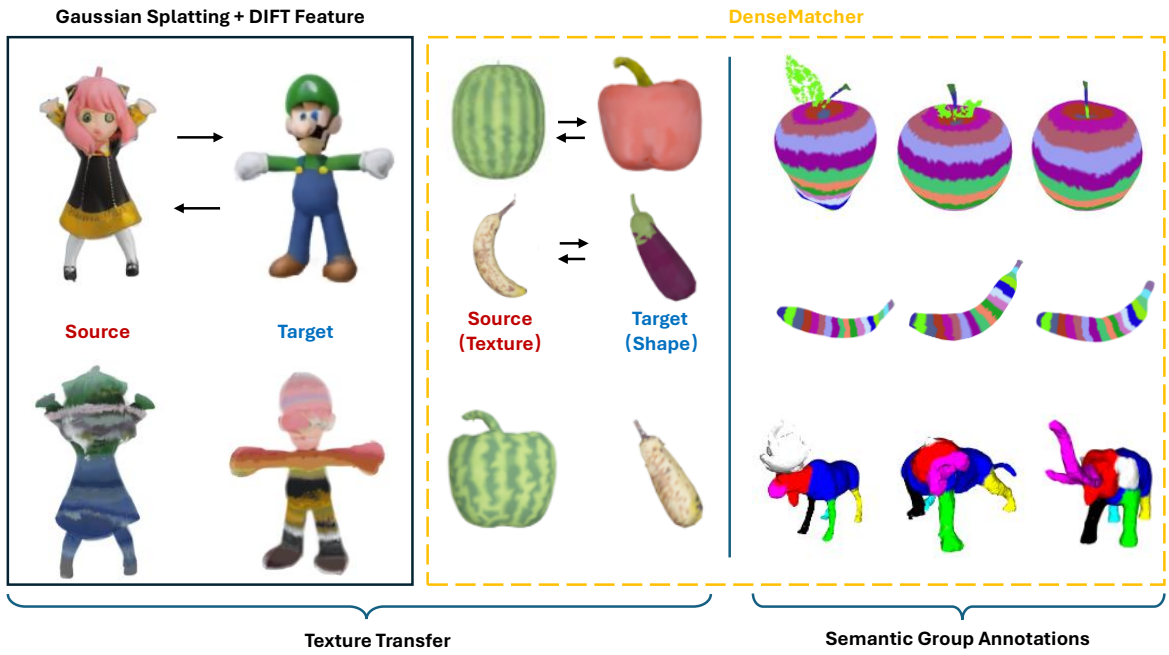

Figure 5. Explicit correspondence. Using explicit correspondence for morphing presents two major challenges: first, obtaining semantic features for tens of thousands of points is extremely difficult; second, the correspondences obtained are typically part-wise, which is inadequate for morphing tasks that require dense correspondences. Note that the results within the yellow dashed boxes are from DenseMatcher [27] manuscript.

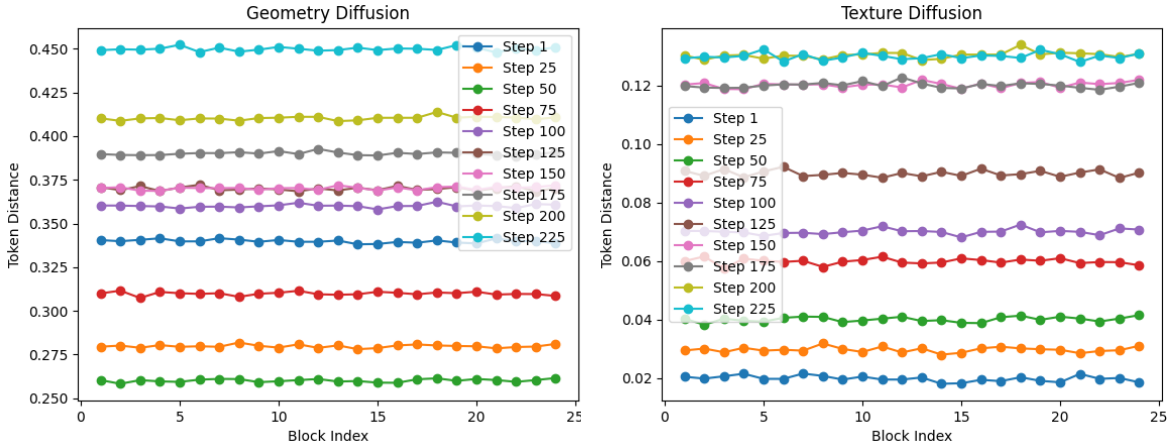

Figure 6. Differences between tokens in two aligned 3D objects from different blocks. At the same time step, there was no notable difference in token distances between the object and the scaled object across various blocks.

Table 2. Beta distribution samples for different alpha and beta (ordered from smallest to largest).

Alpha, Beta	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1, 1	0.19	0.23	0.25	0.34	0.39	0.50	0.52	0.69	0.76	0.77	0.81	0.83	0.84	0.85	0.88	0.89	0.92	0.94	0.94	0.96
3, 3	0.16	0.36	0.39	0.50	0.52	0.54	0.63	0.71	0.73	0.73	0.74	0.77	0.77	0.80	0.83	0.84	0.85	0.88	0.88	0.90
5, 5	0.23	0.30	0.34	0.35	0.44	0.49	0.56	0.66	0.70	0.71	0.73	0.75	0.78	0.80	0.82	0.83	0.85	0.86	0.88	0.89
15, 15	0.36	0.37	0.42	0.42	0.44	0.49	0.52	0.57	0.57	0.58	0.60	0.62	0.63	0.64	0.66	0.67	0.69	0.70	0.72	0.73
20, 20	0.38	0.42	0.48	0.48	0.53	0.56	0.57	0.59	0.56	0.59	0.61	0.63	0.64	0.66	0.67	0.68	0.70	0.71	0.72	0.74
10, 15	0.34	0.37	0.39	0.41	0.42	0.43	0.52	0.52	0.57	0.56	0.57	0.59	0.60	0.61	0.63	0.65	0.66	0.67	0.68	0.69
15, 10	0.37	0.48	0.52	0.55	0.57	0.58	0.59	0.67	0.70	0.80	0.80	0.82	0.84	0.85	0.86	0.88	0.89	0.90	0.91	0.92
5, 15	0.14	0.17	0.18	0.20	0.23	0.24	0.29	0.32	0.38	0.41	0.42	0.44	0.46	0.48	0.49	0.50	0.51	0.53	0.54	0.55
15, 5	0.55	0.58	0.64	0.66	0.68	0.69	0.70	0.80	0.89	0.89	0.90	0.92	0.93	0.94	0.95	0.96	0.97	0.98	0.98	1.00

It is important to emphasize that our setting focuses on textured 3D morphing, which requires fulfilling two conditions: (1) the representation must be textured 3D (such as textured point clouds, textured meshes, or volumetric representations like Gaussian splatting and neural radiance fields); (2) both a source and a target must be specified, as we need to generate a smooth and plausible sequence. In contrast, text-based 3D editing tasks like Instruct-nerf2nerf [6], which do not ensure intermediate state plausibility and lack a specified target, fall outside the scope of our task setting.

7. Semantics of the Same Position in Different Blocks

We also measured the differences between tokens from different blocks by testing with scaled 3D object pairs, as shown in Fig. 6. We found that, at the same time step, there was no significant difference in the token distances between the object and the scaled object across different blocks. Furthermore, the computational cost of Token Reordering and Low-Frequency Enhancement across all blocks is manageable. More importantly, performing manipulation across all block helps preserve the model's inherent capabilities, making our method more robust to input objects. However, our work can also inspire future research in 3D generation, such as exploring improved semantics-aware training methods for 3D generation.

8. Exploratory Experiments with Explicit Correspondence

Inspired by traditional shape morphing and image morphing, our initial method aimed to establish explicit correspondences between textured 3D representations. Specifically, we sought to assign DIFT [21] features to each Gaussian [9]. This method was based on the fact that DIFT features have been validated to provide 3D correspondence for the same object from different viewpoints and semantic correspondence across objects, making this introduction reasonable.

However, we overlooked a key issue: the large number of 3D points, which led to two main drawbacks: (a) high computational cost and (b) the feature handling, which works well for single-point-to-single-point searches from a single viewpoint in 2D foundation models, becomes difficult when attempting to preserve the feature's approximate consistency across different viewpoints.

Table 3. Prompts for testing cases.

Index	Objects	Prompts
1	Stool	*"A wooden tripod stool."*
2	Chair	*"A blue plastic chair."*
3	Llama	*"A realistic 3D model of a llama."*
4	Dog	*"A realistic 3D model of a Husky dog with a big head"*
5	Pumpkin	*"A flat, orange, pixelated Lego pumpkin with a green stem."*
6	Mushroom	*"A light green mushroom."*
7	Car	*"A sleek car with smooth curves, shiny metallic surface, and detailed wheels."*
8	Truck	*"A large red truck with a spacious cargo bed, sturdy wheels, and a robust front grille."*
9	Lounge Sofa	*"A purple lounge sofa."*
10	Massage Sofa	*"A pink medieval-style massage sofa with intricate carvings, plush upholstery, and a comfortable, luxurious design."*
11	Mario	*"A cartoon-style Mario character with a red hat, blue overalls, white gloves, and a cheerful expression."*
12	Luigi	*"A cartoon-style Luigi character with a green hat, blue overalls, and a tall, thin build."*
13	Tank	*"A 3D model of a military tank with detailed textures."*
14	Teapot	*"A classic teapot."*
15	Bowl	*"A simple teal bowl."*
16	Fighter Jet	*"A sleek fighter jet with sharp aerodynamic lines, detailed metallic surface, and camouflage paint."*
17	Seagull	*"A seagull with detailed wings, a sleek body, and a realistic beak, in natural white and gray colors."*
18	Cannon	*"A cannon with a long, cylindrical barrel mounted on a wooden carriage."*
19	House	*"A toy house in a fairy-tale style with whimsical architecture, pastel colors, and charming details like a crooked chimney and flower decorations."*
20	Church	*"A classic church with tall spires, arched windows, and a large central entrance."*
21	Skull	*"A 3D model of a human skull."*
22	Animal Skull	*"A 3D low poly model of an animal skull with gray appearance."*
23	Dinosaur	*"A dinosaur with a large, muscular body, a long tail, and realistic skin texture."*
24	Teddy Bear	*"A red teddy bear."*
25	Polar Bear	*"A low poly model of a polar bear with simplified geometric shapes and flat surfaces, featuring a white body and strong build."*
26	Cow	*"A simple cow model with a stocky body, short legs, and small horns."*
27	Cap	*"A blue baseball cap."*
28	Helmet	*"A medieval helmet with a rounded metal shell and a faceplate."*
29	Donut	*"A donut with a round shape, a hole in the center, and a sugary glaze."*
30	Ice Cream	*"Pink ice cream with a creamy texture, served in a cone or cup."*
31	Hydrant	*"A fire hydrant with a cylindrical body, typically painted in bright red."*
32	Drum	*"A metal oil drum with a cylindrical shape, a top and bottom lid, and a rugged surface."*
33	Vase	*"A light purple vase."*
34	Boot	*"A snow boot with a thick, insulated lining, waterproof exterior, and durable sole for winter conditions."*
35	Fish	*"A chubby fish with a vibrant green body."*

After extensive optimization and adjustment of parameters, we obtained a mapping and performed texture transfer between two cartoon characters. We found that the learned correspondence was inaccurate and exhibited a "layered" characteristic, which closely resembled the patterns discovered by DenseMatcher [27]. However, this correspondence is unsuitable for morphing and would require substantial research effort to address. Therefore, we are more inclined to pursue morphing research based on promising 3D generative models with implicit correspondence.

9. Testing Protocol and Cases

9.1. Quantitative Test Details

For the FID test, the reference images consist of 3000 samples, which were obtained by rendering 15 sets of 3D pairs from 100 different viewpoints. The test images consist of 3000 samples, generated by each method using the same 3D pairs. The remaining quantitative metrics are obtained from testing on the 15 pairs of data.

When evaluating the structural and semantic plausibility of the generated images using GPT-4o, we input the results generated by six methods on the same 3D pairs into GPT-4o for comparison and scoring (similar to the images in Fig. 6). Meanwhile, we provide a guiding prompt that instructs GPT-4o to engage in a step-by-step reasoning process during the evaluation, enhancing both the interpretability and accuracy of the assessment: *"What I am doing now is morphing with textured 3D representations, the purpose is to generate an intermediate interpolation sequence, and at the same time require the transition from source to target to be smooth and reasonable. Now I have six methods, where the first row will give the source and target, and each of the remaining rows is a method. Columns 1-3 are the first test example (morphing from teapot to bowl), columns 4-6 are the second test example (morphing from polar bear to wooden stool), and columns 7-9 are the third test example (morphing from pumpkin to mushroom). Please help me score these methods for the generated intermediate morphing results in terms of shape rationality and semantic rationality, 0 is the lowest score and 1 is the highest score, and give the discrimination results."*

9.2. Beta Distribution

We present sampling results from various Beta distributions for a clear comparison. When alpha $<$ beta, the samples are centered closer to 0 (the source object). Conversely, when alpha $>$ beta, the samples are more biased toward 1 (the target object). When alpha equals beta, larger values concentrate the samples around 0.5. Empirical observations show that the most noticeable morphing occurs near alpha =

Table 4. The raw statistics for user study (Part 1).

	Teapot - Bowl		Polar Bear - Stool		Pumpkin - Mushroom		Teddy Bear - Boot		Mario - Luigi		Average	
	STP-U↑	SEP-U↑	STP-U↑	SEP-U↑	STP-U↑	SEP-U↑	STP-U↑	SEP-U↑	STP-U↑	SEP-U↑	STP-U↑	SEP-U↑
DiffMorpher	0.25	0.20	0.13	0.33	0.60	0.50	0.75	0.40	0.70	0.30	0.49	0.35
AID	0.15	0.53	0.40	0.27	0.50	0.60	0.25	0.32	0.70	0.60	0.40	0.46
MV-Adapter	0.65	0.67	0.07	0.40	0.10	0.50	0.35	0.20	0.02	0.45	0.24	0.44
Luma	0.50	0.13	0.73	0.60	0.10	0.25	0.60	0.60	0.00	0.25	0.39	0.25
MorphFlow	0.45	0.47	0.67	0.53	0.70	0.40	0.30	0.52	0.40	0.40	0.50	0.46
Ours	**1.00**	**1.00**	**0.40**	**0.87**	**1.00**	**0.75**	**0.75**	**0.96**	**1.00**	**1.00**	**0.83**	**0.92**

Table 5. The raw statistics for user study (Part 2).

	Animal Skull - Cow		Car - Truck		House - Church		Chair - Donut		Tank - Cannon		Average	
	STP-U↑	SEP-U↑	STP-U↑	SEP-U↑	STP-U↑	SEP-U↑	STP-U↑	SEP-U↑	STP-U↑	SEP-U↑	STP-U↑	SEP-U↑
DiffMorpher	0.20	0.20	0.36	0.40	0.20	0.25	0.45	0.40	0.70	0.00	0.38	0.25
AID	0.50	0.40	0.28	0.55	0.30	0.65	0.30	0.80	0.40	0.33	0.36	0.55
MV-Adapter	0.20	0.27	0.28	0.40	0.10	0.10	0.45	0.20	0.00	0.33	0.21	0.26
Luma	0.30	0.60	0.52	0.45	0.70	0.35	0.20	0.00	0.50	0.67	0.44	0.41
MorphFlow	0.80	0.53	0.56	0.30	0.70	0.65	0.60	0.60	0.40	0.67	0.61	0.55
Ours	**1.00**	**1.00**	**1.00**	**0.90**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**0.98**

0.5. Several strategies, such as resampling based on variation, can be employed. To ensure fairness in comparison, all methods use morphing based on the sampling results with alpha = beta = 5.

9.3. Prompts for Testing Cases

The strength of our method lies in its ability to perform morphing on diverse cross-category or same-category 3D object pairs. This capability was carefully considered during the selection of test cases, which were primarily categorized into furniture, vehicles, plants, humanoid objects, and animals. Moreover, our method goes beyond shape-only morphing, as we also aim to validate its effectiveness on diverse and richly textured color variations. As such, the color range in our test cases is intentionally broad to ensure comprehensive evaluation. More details about the test cases and their corresponding prompts can be found in Tab. 3.

10. Raw Statistics of User Study

We recruited over 20 volunteers for a user study, where they were asked to rank morphing sequences generated by six methods for the same pairs of test samples. Presenting the results of different methods simultaneously allows users to clearly observe their differences, making the comparison more fair. The 3D object pairs and results are presented in Tab. 4 and Tab. 5.

11. Failure Cases and Analysis

Our work introduces a novel approach to classic shape morphing by leveraging diffusion model priors to overcome limitations imposed by small untextured datasets and the extensive correspondence matching previously required. However, through extensive testing, we have identified unresolved issues and failure cases that will guide future research.

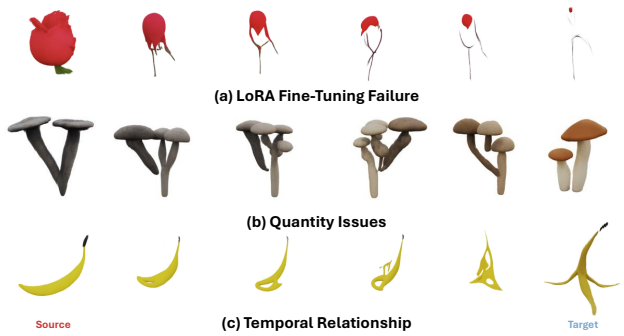

Figure 7. Failure cases occur with challenging morphing pairs.

As shown in Fig. 7, here are three common failure cases: (1) LoRA fine-tuning failure: When fine-tuning the LoRA model, the objects may fall outside the training data distribution of the foundational 3D generation model, leading to the LoRA model's inability to learn correctly. (2) Quantity issues: Due to the lack of good alignment between text and 3D data, objects that require numerical descriptions may experience morphological breakdowns during morphing due to ambiguity. (3) Temporal relationship generation: Since the foundational 3D generation model is trained on limited 3D data, it lacks the rich semantic understanding of 2D generative models, making it difficult to infer temporal relationships, such as turning a banana into a banana peel. However, the fundamental reason lies in the lack of sufficient training data for the foundational model, which hinders the modeling of a semantically rich and smoothly transitioning implicit space. However, our approach provides insights to address these limitations and will continue to improve its handling of these challenging pairs in the future.

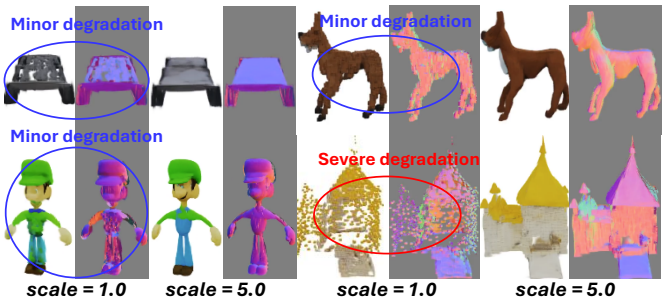

Figure 8. More cases of surface improvement (GaussianAnything).

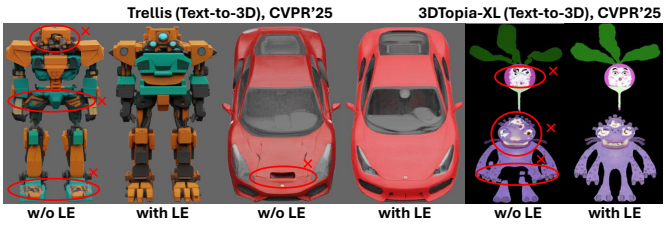

Figure 9. More cases of surface improvement (Trellis and 3DTopia-XL).

Table 6. Quantitative ablation study of texture diffusion.

Texture Diffusion Time Step	5 Steps			40 Steps			80 Steps		
	FID↓	PPL↓	PDV↓	FID↓	PPL↓	PDV↓	FID↓	PPL↓	PDV↓
Basic+AF	105.62	4.44	0.0110	112.70	4.82	0.0173	130.65	4.97	0.0168
Basic+AF+TR	9.41	3.37	0.0024	62.08	4.79	0.0120	70.39	4.80	0.0149
Basic+AF+LE	7.08	3.15	0.0009	50.70	4.42	0.0099	52.92	4.65	0.0125
Basic+AF+TR+LE	**6.36**	**3.02**	**0.0001**	48.17	3.95	0.0062	49.28	4.42	0.0090

12. Surface Improvement by Low-Frequency Enhancement

As shown in Fig. 8 and Fig. 9, we compiled additional results reflecting varying levels of surface generation quality, including outputs from GaussianAnything, Trellis, and 3DTopia-XL.

13. Ablation Study of Texture Diffusion

As shown in Tab. 6, we found that 5 steps already yield good texture interpolation, while more steps tend to cause color distortion.

References

[1] Michael S Albergo, Nicholas M Boffi, and Eric Vanden-Eijnden. Stochastic interpolants: A unifying framework for flows and diffusions. *arXiv preprint arXiv:2303.08797*, 2023. 2

[2] Junsong Chen, Jincheng Yu, Chongjian Ge, Lewei Yao, Enze Xie, Yue Wu, Zhongdao Wang, James Kwok, Ping Luo, Huchuan Lu, et al. Pixart-α: Fast training of diffusion transformer for photorealistic text-to-image synthesis. *ICLR*, 2024. 3

[3] Zhaoxi Chen, Jiaxiang Tang, Yuhao Dong, Ziang Cao, Fangzhou Hong, Yushi Lan, Tengfei Wang, Haozhe Xie, Tong Wu, Shunsuke Saito, et al. 3dtopia-xl: Scaling high-quality 3d asset generation via primitive diffusion. *arXiv preprint arXiv:2409.12957*, 2024. 3

[4] Matt Deitke, Dustin Schwenk, Jordi Salvador, Luca Weihs, Oscar Michel, Eli VanderBilt, Ludwig Schmidt, Kiana Ehsani, Aniruddha Kembhavi, and Ali Farhadi. Objaverse: A universe of annotated 3d objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13142–13153, 2023. 3

[5] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7473–7483, 2021. 4, 5

[6] Ayaan Haque, Matthew Tancik, Alexei A Efros, Aleksander Holynski, and Angjoo Kanazawa. Instruct-nerf2nerf: Editing 3d scenes with instructions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19740–19750, 2023. 6

[7] Qiyuan He, Jinghao Wang, Ziwei Liu, and Angela Yao. Aid: Attention interpolation of text-to-image diffusion. *NeurIPS*, 2024. 4

[8] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021. 4

[9] Binbin Huang, Zehao Yu, Anpei Chen, Andreas Geiger, and Shenghua Gao. 2d gaussian splatting for geometrically accurate radiance fields. In *ACM SIGGRAPH 2024 conference papers*, pages 1–11, 2024. 2, 6

[10] Zehuan Huang, Yuanchen Guo, Haoran Wang, Ran Yi, Lizhuang Ma, Yan-Pei Cao, and Lu Sheng. Mv-adapter: Multi-view consistent image generation made easy. *arXiv preprint arXiv:2412.03632*, 2024. 3, 4

[11] Zixuan Huang, Justin Johnson, Shoubhik Debnath, James M Rehg, and Chao-Yuan Wu. Pointinfinity: Resolution-invariant point diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10050–10060, 2024. 2

[12] 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. 4, 5

[13] Yushi Lan, Shangchen Zhou, Zhaoyang Lyu, Fangzhou Hong, Shuai Yang, Bo Dai, Xingang Pan, and Chen Change Loy. Gaussiananything: Interactive point cloud latent diffusion for 3d generation. In *ICLR*, 2025. 2, 3

[14] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4195–4205, 2023. 2, 4

[15] Lingteng Qiu, Guanying Chen, Xiaodong Gu, Qi Zuo, Mutian Xu, Yushuang Wu, Weihao Yuan, Zilong Dong, Liefeng

Bo, and Xiaoguang Han. Richdreamer: A generalizable normal-depth diffusion model for detail richness in text-to-3d. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9914–9925, 2024. 3

[16] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. 4

[17] Mehdi SM Sajjadi, Henning Meyer, Etienne Pot, Urs Bergmann, Klaus Greff, Noha Radwan, Suhani Vora, Mario Lučić, Daniel Duckworth, Alexey Dosovitskiy, et al. Scene representation transformer: Geometry-free novel view synthesis through set-latent scene representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6229–6238, 2022. 2

[18] Yichun Shi, Peng Wang, Jianglong Ye, Mai Long, Kejie Li, and Xiao Yang. Mvdream: Multi-view diffusion for 3d generation. *arXiv preprint arXiv:2308.16512*, 2023. 3

[19] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020. 4

[20] 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 Conference Papers*, pages 1–11, 2024. 4, 5

[21] Luming Tang, Menglin Jia, Qianqian Wang, Cheng Perng Phoo, and Bharath Hariharan. Emergent correspondence from image diffusion. *Advances in Neural Information Processing Systems*, 36:1363–1389, 2023. 6

[22] Chih-Jung Tsai, Cheng Sun, and Hwann-Tzong Chen. Multiview regenerative morphing with dual flows. In *European Conference on Computer Vision*, pages 492–509. Springer, 2022. 4

[23] 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. 3

[24] Xiao Zhan, Rao Fu, and Daniel Ritchie. Charactermixer: Rig-aware interpolation of 3d characters. In *Computer Graphics Forum*, page e15047. Wiley Online Library, 2024. 4, 5

[25] Kaiwen Zhang, Yifan Zhou, Xudong Xu, Bo Dai, and Xingang Pan. Diffmorpher: Unleashing the capability of diffusion models for image morphing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7912–7921, 2024. 4

[26] Longwen Zhang, Ziyu Wang, Qixuan Zhang, Qiwei Qiu, Anqi Pang, Haoran Jiang, Wei Yang, Lan Xu, and Jingyi Yu. Clay: A controllable large-scale generative model for creating high-quality 3d assets. *ACM Transactions on Graphics (TOG)*, 43(4):1–20, 2024. 3

[27] Junzhe Zhu, Yuanchen Ju, Junyi Zhang, Muhan Wang, Zhecheng Yuan, Kaizhe Hu, and Huazhe Xu. Densematcher: Learning 3d semantic correspondence for category-level manipulation from a single demo. *arXiv preprint arXiv:2412.05268*, 2024. 5, 7