

CaliTex: Geometry-Calibrated Attention for View-Coherent 3D Texture Generation

Chenyu Liu¹, Hongze Chen², Jingzhi Bao³, Lingting Zhu⁴, Runze Zhang⁵, Weikai Chen⁵, Zeyu Hu⁵, Yingda Yin⁵, Keyang Luo⁵, Xin Wang⁵

¹PKU, ²HKUST, ³CUHK(SZ), ⁴HKU, ⁵LIGHTSPEED



Figure 1. A collection of 3D objects textured by our method, demonstrating high-fidelity, seamless and geometry-aligned textures facilitated by our framework with geometry-calibrated attention.

Abstract

Despite major advances brought by diffusion-based models, current 3D texture generation systems remain hindered by cross-view inconsistency – textures that appear convincing from one viewpoint often fail to align across others. We find that this issue arises from attention ambiguity, where unstructured full attention is applied indiscriminately across tokens and modalities, causing geometric confusion and unstable appearance-structure coupling. To address this, we introduce CaliTex, a framework of geometry-calibrated attention that explicitly aligns attention with 3D structure. It introduces two modules: Part-Aligned Attention that enforces spatial alignment across se-

mantically matched parts, and Condition-Routed Attention which routes appearance information through geometry-conditioned pathways to maintain spatial fidelity. Coupled with a two-stage diffusion transformer, CaliTex makes geometric coherence an inherent behavior of the network rather than a byproduct of optimization. Empirically, CaliTex produces seamless and view-consistent textures and outperforms both open-source and commercial baselines.

1. Introduction

The automatic generation of 3D assets holds immense promise for content creation in gaming, film, and virtual

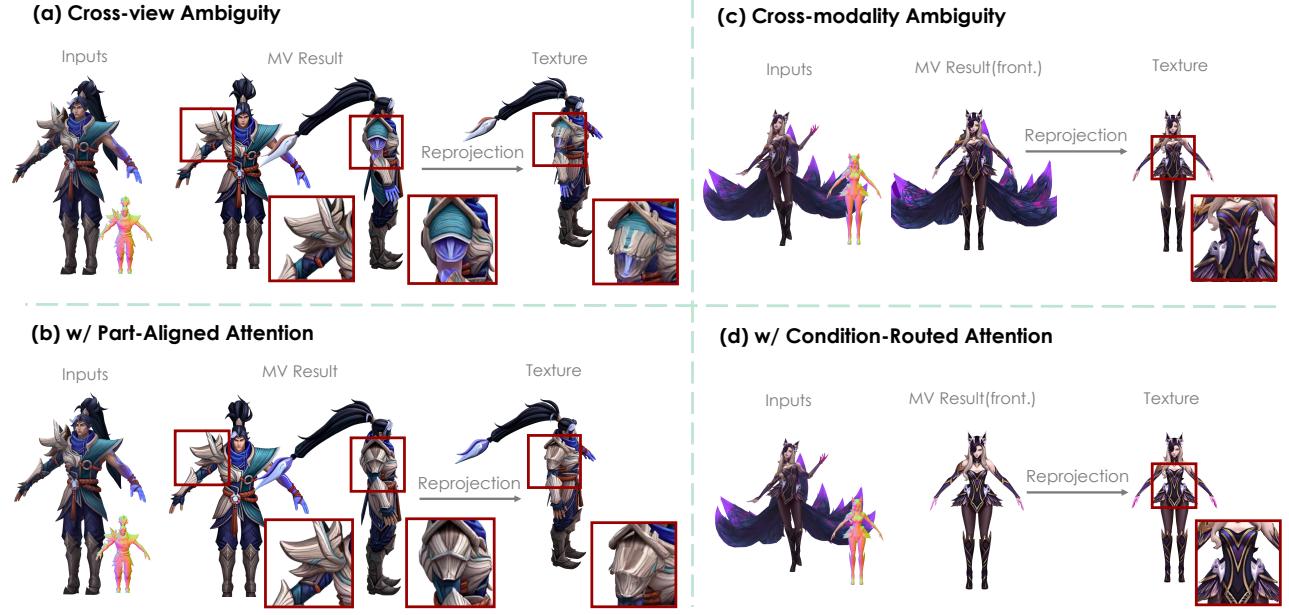


Figure 2. **Illustration issues caused by attention ambiguity and our proposed solutions.** Zoom in for more details. (a) The model confuses the left limb in the second view with the right limb, producing seams in the texture. (b) Our Part-Aligned Attention constrains attention computation within semantic parts, effectively eliminating cross-view inconsistency. (c) The model directly copies visually similar regions from the reference image, leading to misalignment with the geometry condition. (d) Our Condition-Routed Attention ensures geometry-aligned texture generation, correcting the distortion on the clothing, as highlighted in the bottom-right.

reality. Recent advances in native 3D generative modeling [21, 39, 50, 53] have significantly improved generation quality, marking a shift from traditional SDS-based pipelines [32, 40] to more powerful diffusion-based approaches. Most advanced techniques adopt a two-stage paradigm: geometry is first synthesized, followed by texture generation conditioned on the geometry. In this process, 2D generative priors from image diffusion models [1, 11, 31, 34, 52] are leveraged to synthesize multi-view images [28, 36, 53], which are then reprojected back onto the surface to construct textures. While this strategy delivers impressive appearance quality, it often collapses under cross-view inconsistency – the same surface region can appear differently across generated views, producing seams or blurring after reprojection. Shown in Fig. 2 (a,c), such artifacts do not stem from rendering, but rather from a deeper representation misalignment within the model itself.

Unlike conventional text-to-image models, where full attention across all noise and text tokens is both sufficient and necessary to capture a semantic context, texture generation involves a far more intricate interplay among noise tokens, geometry condition tokens, and reference image tokens. Current state-of-the-art methods [23, 58, 59] simply extend full attention indiscriminately across all tokens and views, presuming correspondences will emerge. However, this premise does not hold in practice.

Such a naïve full-attention design leads to two inherent

ambiguities that compromise texture consistency and detail. (i) *Cross-view Ambiguity* (Fig. 2 (a)). Tokens belonging to geometrically similar yet distinct regions, such as the left and right limbs, often attend to each other across views. Consequently, the model confuses distinct parts and generates nearly identical local textures for both. After weighted blending, these misaligned views are projected back onto inconsistent surface regions, resulting in seams and loss of spatial coherence. (ii) *Cross-modality Ambiguity* (Fig. 2 (c)). The model also suffers from unstable cross-modal attention: noise tokens alternatively attend to the reference image or the geometry condition. This results in either appearance overfitting (copying visual patterns) or geometric overreliance (losing appearance fidelity), producing superficially realistic but geometrically inconsistent textures.

We contend that geometric consistency does not arise automatically from training – it requires architectural calibration. Instead of introducing additional supervision or handcrafted priors, we reformulate attention itself to be geometry-aware – guiding the model to know *where to focus* and *how information flows* across modalities. Following this principle, we introduce CaliTEx, a framework of *Geometry-Calibrate Attention* that explicitly aligns attention with 3D structure at both spatial and informational levels. This calibration is instantiated through two complementary mechanisms (Fig. 2 (b,d)): *Part-Aligned Attention* and *Condition-Routed Attention*.

Part-Aligned Attention (PAA) mitigates cross-view ambiguity by imposing geometry-aware locality. The key idea is to constrain attention propagation within the same semantic part, preventing tokens from mistakenly attending to geometrically similar but distinct regions across views. Specifically, we use PartField [27] to decompose the mesh into semantic components, assigning each face a discrete part index. By rendering part-colored images from multiple viewpoints, we obtain part labels for every token, enabling us to group tokens that represent the same 3D part across different views. Cross-view attention is then computed within each group, while intra-view full attention is preserved to capture broader context. As shown in Fig. 2 (b), our method effectively eliminates cross-view inconsistency, yielding aligned fine details across different viewpoints.

Condition-Routed Attention (CRA) addresses cross-modality ambiguity by restructuring how information flows between reference and noise tokens. Instead of allowing noise and reference tokens to interact freely, CRA imposes a dual-pathway design: one branch computes attention between condition and reference tokens, capturing geometric priors from appearance cues, while the other links condition and noise tokens, injecting those geometry-aware features back into the generative process. This routing mechanism (Fig. 2 (d)) ensures that appearance information is always mediated by geometry, suppressing direct reference copying and maintaining consistent texture alignment with the underlying 3D surface.

Together, these mechanisms are implemented within a two-stage diffusion transformer: a single-view DiT first captures intra-view semantics under full attention, while a multi-view DiT equipped with PAA and CRA enforces cross-view and cross-modal coherence. This design not only stabilizes attention dynamics but also transforms geometric consistency from a post-hoc regularization into an intrinsic property of the generative process. Empirically, it yields seamless and geometrically faithful textures, even for complex assets exhibiting high symmetry or self-occlusion, as demonstrated in Fig. 2 (b,d). Our main contributions are summarized as follows:

- We present CaliTEx, a novel framework of geometry-calibrated attention that embeds geometric reasoning directly into attention computation.
- We propose part-aligned attention to enforce spatial alignment across views via part-level geometric priors.
- We introduce conditioned-routed attention to regulate reference-noise interaction through geometry-conditioned routing, achieving stable cross-modal alignment.
- We achieve high-fidelity 3D texture generation results that surpass existing state-of-the-art baselines.

2. Related Work

2.1. Texture Generation for 3D Objects

With the advent of diffusion-based generative models, texture generation for 3D objects has been extensively explored. Early approaches based on Score Distillation Sampling (SDS) [3, 5, 9, 24, 32, 47, 55] attempted to optimize 3D representations such as NeRF and 3D Gaussian Splatting [17, 30] by rendering images with sampled random noise and predicting the noise using a pre-trained 2D diffusion model. However, SDS methods often lead to oversaturated colors and suffer from the Janus problem. Recently, the rapid progress of text-conditioned image generation models [11, 20, 25] has inspired many works on 3D texture generation. Some methods [2, 33, 51] iteratively generate single-view images from arbitrary viewpoints, but often suffer from poor cross-view consistency. More recent approaches generate multiple views simultaneously, demonstrating strong quality [6, 15, 21, 23, 38, 50, 59]. However, they fail to achieve proper cross-view and view-geometry alignment when the geometries are highly complex.

Another line of work directly predicts 3D representations with colors [26, 41, 42, 48, 49]. UniTEX [23] introduces a two-stage pipeline: it first generates multi-view images with a diffusion model, followed by a transformer-based Large Texturing Model to predict the texture function in 3D space from both images and geometry. These approaches can inherently avoid issues caused by self-occlusion, but due to the scarcity of large-scale textured 3D training data, their outputs generally underperform compared to methods that leverage existing 2D image generation models in quality.

2.2. 3D Texture Alignment

The cross-view and view-geometry misalignments significantly affect texture quality, and several recent works have attempted to address these issues. SeqTex [50] directly generates UV-space textures alongside multi-view images, replacing the traditional back-projection pipeline. However, recent high-quality 3D generation models [4, 22, 39] typically rely on automatic UV unwrapping tools such as xatlas [46], where fragmented UV layouts often lead to degraded generation quality. Elevate3D [35] refines single-view generations iteratively to improve cross-view consistency and modifies geometry to maintain coherence with synthesized views, though this may inadvertently alter original geometric details. Romantex [12] introduces a 3D-aware RoPE to inject spatial information into the generative model and employs a geometry-related CFG technique to enhance alignment with geometry. AlignTex [56] strengthens geometry alignment by fusing image and geometry features within the diffusion process.

2.3. 3D Semantic Segmentation

Recent progress in image segmentation, especially with models such as SAM [19], has also driven advances in 3D semantic segmentation. Several methods use 2D segmentation priors from SAM to improve 3D semantic understanding [18, 27, 37, 44, 45]. Among them, PartField [27] trains a network with contrastive learning to predict a continuous 3D feature field, which is then clustered to obtain high-quality 3D segmentation efficiently. In contrast, P³-SAM [29] operates purely in the 3D domain, addressing the spatial inconsistency problem found in methods that depend on 2D segmentation labels. Several works have also explored part-level generation [10, 43] and part-level refinement in 3D generation tasks [4], which inspires us to incorporate semantic part priors into the 3D texture generation.

3. Method

3.1. Overview

The overview of our method is shown in Fig. 3. Following previous works [23, 38], we finetune a large-scale image generation network (FLUX.1-Kontext [20]) to synthesize six consistent views from an untextured mesh and a reference image. These generated views are then reprojected to form the texture map and inpainting is applied to fill occluded regions. The backbone of our framework is described in Section 3.2, which consists of two main components: Single-View DiT and Multi-View DiT. In the Single-View DiT stage, we perform batch-wise full attention to capture global information within individual views. In the Multi-View DiT stage, we unfold the batch dimension to perform multi-view attention, enabling information exchange across different views. Our implementations of the modules in Multi-View DiT stage are explained in detail in Section 3.3 and Section 3.4.

3.2. Multi-View Generation Network

In this section, we describe the general structure of our multi-view generation network based on Flux [1]. Given an untextured mesh and a reference image as inputs, we first render normal maps and canonical coordinate maps (CCMs) from the mesh under six predefined viewpoints. These two maps are averaged and then encoded into the latent space via a variational autoencoder (VAE), producing the geometric condition latent z_{cond} . Similarly, the reference image is encoded into z_{ref} , which is replicated six times to match the batch dimension. Let z_t denote the noise latent at timestep t . We concatenate the noise, condition, and reference latents along the sequence dimension, forming $\hat{z}_t \in \mathbb{R}^{6 \times 3L \times C}$, where L and C denote the token length per view and the feature dimension respectively, and the six views are tiled along the batch dimension. Next, a Single-View DiT performs batch-wise full attention to cap-

ture intra-view correspondences among the noise, condition, and reference latents.

Then, we flatten the batch dimension and concatenate all per-view noise latents, condition latents and the view-averaged reference latent to form the input of the next stage, denoted as $\tilde{z}_{\text{mv}} \in \mathbb{R}^{1 \times 13L \times C}$. It is then fed into the Multi-View DiT to enhance cross-view consistency.

Finally, the noise latent part z'_{img} is used as the prediction target, and the model is trained via the flow-matching objective $\mathcal{L}(\theta) = \mathbb{E}_{t, z_0, \epsilon} [\|z'_{\text{img}} - (\epsilon - z_0)\|^2]$ [25].

3.3. Condition-Routed Attention

Next, we describe the two modules in the Multi-View DiT stage. The input to this stage $\tilde{z}_{\text{mv}} \in \mathbb{R}^{1 \times 13L \times C}$ consists of three parts: the noise tokens and condition tokens from six views, along with the reference image tokens. Previous approaches [23] simply apply full attention to learn correlations among these modalities. However, this often leads to a cross-modality ambiguity problem, where noise tokens alternately attend to either the reference image or the geometry condition, lacking stable geometric alignment and producing artifacts in the generated textures.

To address this issue, we redesign the attention mechanism in the Multi-View DiT as a Condition-Routed Attention (CRA), composed of two parallel attention pathways. The key idea is to calibrate the visual prior from the reference image through the geometry condition, and then let the resulting geometry-aware features guide the noise latents, thereby improving alignment with geometric priors.

Specifically, we form two token groups: (1) a *condition–reference* group that fuses geometric and visual priors, and (2) a *noise–condition* group that guides the generation toward geometry-aligned outputs. We first compute the self-attention within the *condition–reference* group. Let the query, key, and value matrices derived from all tokens in this group be $(Q_{\text{c-r}}, K_{\text{c-r}}, V_{\text{c-r}})$. The attention is computed as:

$$\text{Attn}_{\text{c-r}} = \text{Softmax}\left(\frac{Q_{\text{c-r}}K_{\text{c-r}}^\top}{\sqrt{d}}\right)V_{\text{c-r}}, \quad (1)$$

where d denotes the feature dimension. Meanwhile, the attention within the *noise–condition* group is calculated as $\text{Attn}_{\text{n-c}}$, which will be detailed in Section 3.4. Finally, we merge the outputs from both groups as:

$$\text{Attn}_{\text{CRA}} = \text{Attn}_{\text{n-c}} \cup \text{Attn}_{\text{c-r}}. \quad (2)$$

where \cup denotes that the attention between each token pair is computed only once. The specific implementation of how two attention modules are merged is provided in the supplementary material. This dual-pathway attention operates across all 38 attention blocks of the Multi-View DiT. In each block, the visual priors from reference tokens are fused

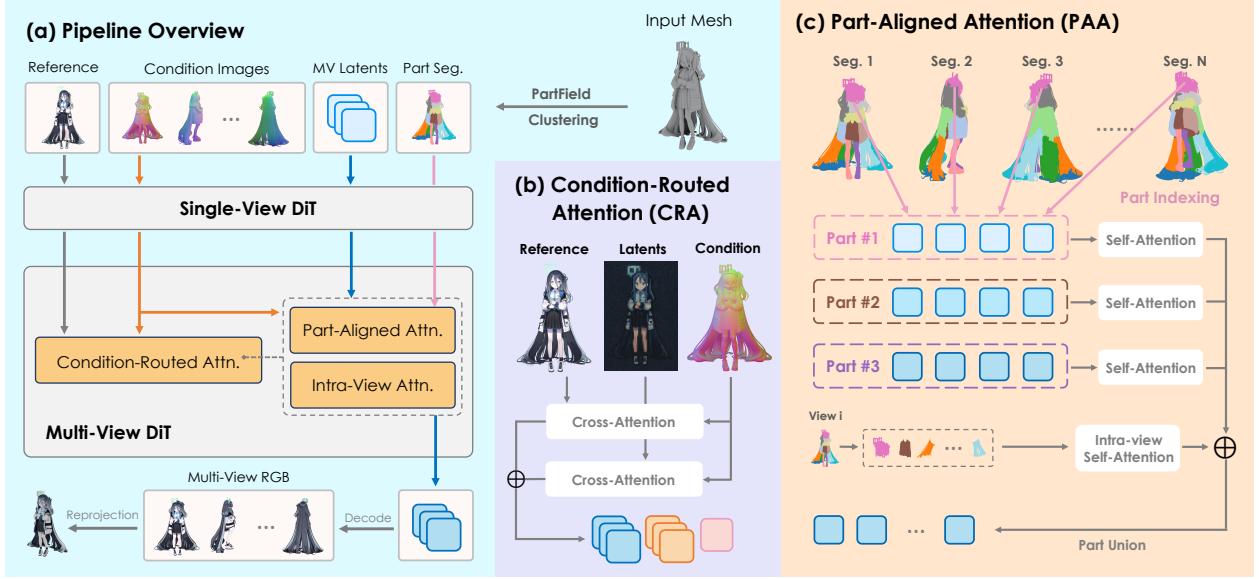


Figure 3. Overview of our method. (a) We employ a two-stage generation framework: the *Single-View DiT* captures intra-view correlations, while the *Multi-View DiT* enhances geometric alignment and cross-view consistency using (b) *Condition-Routed Attention* and (c) *Part-Aligned Attention*. The generated multi-view images are then projected back and inpainted to produce the final 3D texture.

into the geometry condition tokens, which are then used in the subsequent block to guide generation. This mechanism effectively prevents direct reference copying and promotes texture alignment with the underlying 3D surface.

3.4. Part-Aligned Attention

In this section, we describe the multi-view attention mechanism within the noise-condition branch in Section 3.3. Previous methods simply employ full attention to capture multi-view correspondences, allowing the model to learn from data that similar shapes should correspond to the same regions and thereby yield coherent textures. However, such global modeling often struggles in cases of symmetry or geometric similarity, resulting in inconsistencies across views.

To address this limitation, we introduce a Part-Aligned Attention (PAA) mechanism that leverages geometric priors to group spatially adjacent tokens across different views. Specifically, we adopt PartField [27] to decompose the 3D mesh \mathcal{M} into K semantic parts (we set $K = 20$, which is suitable for most common 3D objects):

$$\mathcal{M} = \{\mathcal{P}_1, \mathcal{P}_2, \dots, \mathcal{P}_K\}, \quad (3)$$

where each face $f \in \mathcal{P}_k$ is assigned a part index k . We then render the mesh, all faces belonging to the same part \mathcal{P}_k are assigned an identical color across six views, producing part-colored images.

In the latent preparation stage, the model encodes both the condition and reference images into the latent space using a VAE with a down-sampling factor F . The resulting feature maps are then partitioned into patches of size P ,

where each token represents an $FP \times FP$ image region. Let $\mathbf{T}^{(v)} = \{t_1^{(v)}, t_2^{(v)}, \dots, t_N^{(v)}\}$ denote the set of latent tokens extracted from view v , where each token $t_i^{(v)} \in \mathbb{R}^d$ corresponds to a local image patch on the rendered condition image or its generative result. We denote the union of all multi-view tokens as $\mathbf{T} = \bigcup_{v=1}^6 \mathbf{T}^{(v)} = \{t_1, t_2, \dots, t_{6N}\}$.

For each token $t_i \in \mathbf{T}$, we determine its semantic part labels based on the previous rendered part-colored images. Each pixel is assigned a part index $c(p) \in \{1, \dots, K\}$, where K is the total number of semantic parts. If any pixel within the image patch corresponding to t_i belongs to part k , we assign t_i to group \mathcal{G}_k .

$$\mathcal{G}_k = \{t_i \mid \exists p \in \text{Patch}(t_i), c(p) = k\}, \quad k = 1, \dots, K. \quad (4)$$

Notice that it allows a token to be associated with multiple semantic groups if it overlaps with the boundaries of different parts. Then, we perform self-attention jointly on the noise and condition latent tokens within each part group \mathcal{G}_k . For tokens belonging to the same group, their query, key, and value matrices are denoted as Q_k, K_k, V_k , respectively. The part-level self-attention is formulated as:

$$\text{Attn}_k(Q, K, V) = \text{Softmax}\left(\frac{Q_k K_k^\top}{\sqrt{d}}\right) V_k. \quad (5)$$

The outputs from all part groups are then aggregated to obtain the overall part-aligned attention:

$$\text{Attn}_{\text{PAA}}(Q, K, V) = \bigcup_{k=1}^K \text{Attn}_k(Q, K, V). \quad (6)$$

Also, to maintain global perception within each view, we retain full attention for intra-view context:

$$\text{Attn}_{\text{intra}}^{(v)}(Q, K, V) = \text{Softmax} \left(\frac{Q^{(v)} K^{(v)\top}}{\sqrt{d}} \right) V^{(v)}, \quad (7)$$

where $Q^{(v)}$, $K^{(v)}$, and $V^{(v)}$ denote the query, key, and value matrices derived from tokens of view v . Finally, we union the part-aligned and intra-view attention together, obtaining the noise-condition branch attention:

$$\text{Attn}_{\text{n-c}} = \text{Attn}_{\text{PAA}} \cup \text{Attn}_{\text{intra}}. \quad (8)$$

This design constrains cross-view attention to small, semantically coherent 3D-local regions. By aligning attention with 3D semantic parts, PAA strengthens the model’s perception of 3D spatial structure and corrects misaligned correspondences, resulting in more coherent textures.

4. Experiments

4.1. Implementation Details

We selected 80k objects from Objaverse-XL [7] and Texverse [57] for model training. For each object, we rendered six pre-defined viewpoints at a resolution of 768×768, which were used as the ground truth multi-view images. Additionally, a random view was rendered as the reference image. Our DiT backbone was initialized from FLUX.1-Kontext [20], and we integrate a LoRA adapter [14] with a rank of 16. The model was trained on 8 GPUs for approximately 600 GPU hours in total.

4.2. Comparisons

Baselines. We compare our method with several open-source state-of-the-art image-to-texture approaches, including MV-Adapter [15], UniTEX [23], Step1X-3D [21], and Hunyuan3D 2.1 [38]. In addition, we also include comparisons with high-fidelity commercial models, as discussed in Section 4.2.2.

Metrics. We use FID [13], CLIP-FID, CLIP Maximum-Mean Discrepancy (CMMD) [16], and CLIP-I to evaluate the fidelity of generated textures. We further use LPIPS [54] to measure the perceptual similarity between the generated textures and the ground truth.

4.2.1. Quantitative Comparison

Textured Mesh Evaluation. We evaluate our method on a diverse test set including objects from Objaverse [8] and high-quality game assets, ranging from simple objects to highly detailed models with complex geometry and refined textures. To compare fairly with PBR-based methods, we use illumination-free albedo renderings as reference images. Specifically, for game characters, a random animated

pose is sampled as reference while textures are generated on the canonical A-pose to simulate shape discrepancies between reference images and 3D geometries. Each mesh is rendered from 32 viewpoints and compared with the corresponding ground truth albedo images.

As shown in Tab. 1, our method achieves the lowest FID, indicating higher overall fidelity. It also attains superior semantic fidelity, achieving lower CLIP-FID and CMMD and higher CLIP-I, as well as better perceptual quality reflected by lower LPIPS.

User Study. To further assess human perceptual quality, we conduct a user study with 25 participants who rate results (1–5) from different methods along three perceptual dimensions: texture overall quality (*Qual*), geometric alignment (*GeoAlign*), and multi-view consistency (*MV-Cons*). Texture quality measures the visual appearance and fidelity to reference images, geometric alignment evaluates alignment with the underlying mesh, and multi-view consistency checks for naturalness across views, avoiding ghosting and seams. As shown in Tab. 1, our method receives the highest scores on the overall quality, showing strong agreement with human preference. Facilitated by our Condition-Routed Attention and Part-Aligned Attention modules, our method also achieves strong performance on geometric alignment and multi-view consistency.

4.2.2. Qualitative Comparison

We further present qualitative comparisons with other methods in Fig. 4, including Hunyuan3D-3.0 to compare with the latest commercial models. MV-Adapter, Hunyuan3D-2.1 and Hunyuan3D-3.0 tend to produce blurry textures on geometrically complex regions and sometimes fail to align semantically with the reference image in certain parts. UniTEX generates overly smooth textures, resulting in the loss of fine local details. In addition, its multi-view consistency is relatively low, producing visible seams in the textures. In contrast, our method preserves almost all fine details from the reference images and achieves higher visual fidelity than all other approaches. Furthermore, with our Condition-Routed Attention, we maintain strong geometric coherence, producing minimal artifacts, while our Part-Aligned Attention improves multi-view consistency, resulting in textures that are consistent across views and free of seams.

4.3. Ablation Study

4.3.1. Pixel-Level Multi-View Alignment

To quantitatively evaluate the effectiveness of our two proposed attention modules, we introduce a metric called **Multi-View Mean Squared Error (MV-MSE)**, which measures the pixel-level consistency across different rendered views. For each pair of generated views among the six pre-defined viewpoints, we compute the average MSE

Method	Quantitative Metrics Evaluation					User Study		
	FID↓	CLIP-FID↓	CMMMD↓	CLIP-I↑	LPIPS↓	Qual↑	GeoAlign↑	MV-Cons↑
Step1X-3D [21]	254.1	28.23	3.914	0.8433	0.3154	2.06	2.09	2.07
UniTEX [23]	176.2	17.19	1.156	0.8818	0.3335	3.10	3.21	3.15
MV-Adapter [15]	169.4	13.61	0.747	0.8975	0.2939	3.02	3.03	2.88
Hunyuan3D-2.1 [38]	167.4	16.21	1.067	0.8867	0.3215	2.48	2.60	2.51
Ours	157.8	12.85	0.672	0.9106	0.2508	4.53	4.47	4.52

Table 1. **Quantitative comparison with baseline methods.** We evaluate generation fidelity using FID, CLIP-FID, CMMMD, CLIP-I, and LPIPS on novel-view renderings, and assess human perceptual quality through a user study rating texture quality (*Qual*), geometric alignment (*GeoAlign*), and multi-view consistency (*MV-Cons*). Our method achieves the best performance across all metrics and human evaluations, demonstrating superior visual fidelity, geometric coherence, and consistency across views.

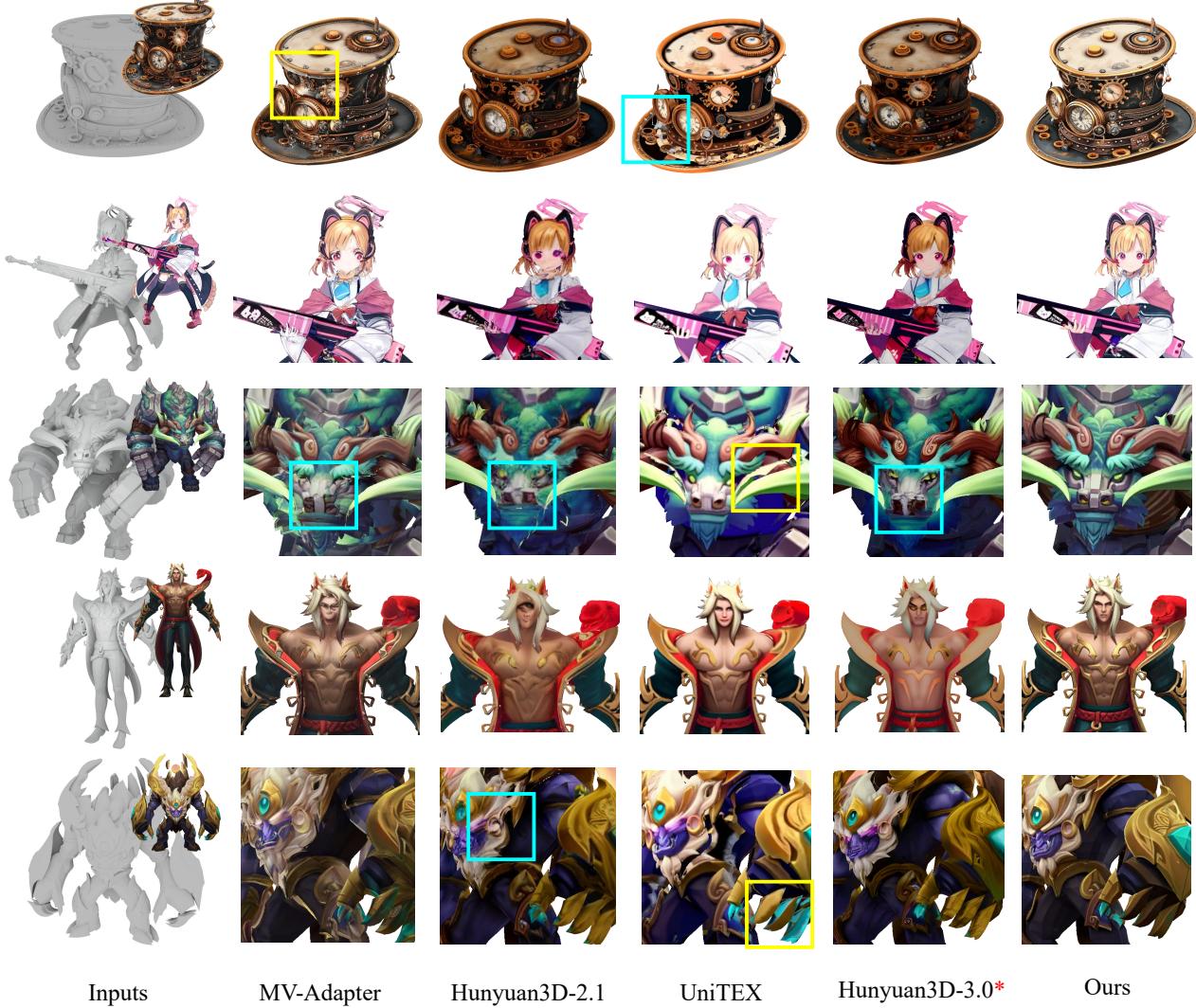


Figure 4. **Qualitative comparison with recent methods.** We compare our approach with both open-source and commercial models (marked with *) on various objects. Regions highlighted in yellow indicate seams or cross-view inconsistencies, while regions highlighted in blue denote misalignment with the underlying geometry. Please zoom in for more details. Meshes are generated by Hunyuan3D-3.0.

Method	MV-MSE ↓
Ours	0.0384
w/o Part-Aligned	0.0415
w/o Condition-Routed Attention	0.0403

Table 2. **Ablation on pixel-level multi-view alignment.**

over pixels that correspond to the same 3D locations. Formally, MV-MSE is defined as:

$$\text{MV-MSE} = \frac{2}{N(N-1)} \sum_{(i,j)} \frac{1}{|\Omega_i(j)|} \sum_{p \in \Omega_i(j)} \|I_i(p) - I_j(\pi_j(X_p))\|_2^2, \quad (9)$$

where I_i and I_j denote the generated images from views i and j , X_p is the 3D position corresponding to pixel p in view i , $\pi_j(\cdot)$ projects the 3D point into view j , and $\Omega_i(j)$ denotes the set of pixels in view i whose projected 3D points have a valid corresponding pixel in view j within a small spatial threshold.

As shown in Tab. 2, our modules reduce MV-MSE by mitigating geometry–texture mismatch and enhancing cross-view consistency.

4.3.2. Part-Aligned Attention

As shown in Fig. 5, removing the Part-Aligned Attention causes the model to ambiguously associate geometrically similar regions as the same location across views. Such ambiguous cross-view attention leads to incorrectly aligned multi-view images, producing artifacts in generated 3D textures. By constraining attention within each local part, the proposed Part-Aligned Attention effectively eliminates these erroneous correspondences, resulting in coherent textures with significantly fewer artifacts across all viewpoints.

4.3.3. Condition-Routed Attention

Fig. 6 shows textures generated with and without the Condition-Routed Attention. Removing this module results in inconsistencies between the generated views and the geometry condition. In contrast, by regulating the attention flow through the geometry condition, the module improves texture–geometry alignment and produces more consistent results, mitigating artifacts in the final textures.

5. Conclusion

In this work we present CaliTEx, a novel framework for high-fidelity 3D texture generation that explicitly addresses attention ambiguity in multi-view diffusion models. We identify two primary sources of ambiguity: (1) cross-view attention ambiguity among geometrically similar regions, which can cause inconsistent textures



Figure 5. **Ablation study of Part-Aligned Attention.** Without Part-Aligned Attention, ambiguous cross-view attention causes incorrect alignment across views, while our method yields correct results.

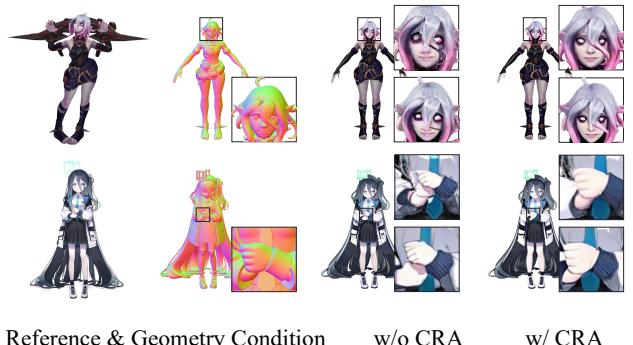


Figure 6. **Ablation study on the Condition-Routed Attention.** We compare textures generated with and without the proposed Condition-Routed Attention. The bottom-right inset shows the corresponding multi-view generation results, while the top-right inset illustrates the final textured mesh.

across views, and (2) cross-modal attention ambiguity, which can degrade geometry–texture correspondence. To address these problems, we propose two attention mechanisms: Part-Aligned Attention to constrain attention within semantic parts and Condition-Routed Attention to mediate attention flow through the geometry condition. Extensive experiments demonstrate that our method significantly improves texture quality, outperforming existing open-source and commercial baselines. Our study highlights the importance of geometry-aware attention design and takes a step toward automatic, artist-level 3D texture creation.

References

- [1] BlackForestLabs. Flux.1 model family. 2024. 2, 4
- [2] Dave Zhenyu Chen, Yawar Siddiqui, Hsin-Ying Lee, Sergey Tulyakov, and Matthias Nießner. Text2tex: Text-driven texture synthesis via diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 18558–18568, 2023. 3
- [3] Rui Chen, Yongwei Chen, Ningxin Jiao, and Kui Jia. Fantasia3d: Disentangling geometry and appearance for high-quality text-to-3d content creation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22246–22256, 2023. 3
- [4] Yiwen Chen, Zhihao Li, Yikai Wang, Hu Zhang, Qin Li, Chi Zhang, and Guosheng Lin. Ultra3d: Efficient and high-fidelity 3d generation with part attention, 2025. 3, 4
- [5] Zilong Chen, Feng Wang, Yikai Wang, and Huaping Liu. Text-to-3d using gaussian splatting, 2024. 3
- [6] Wei Cheng, Juncheng Mu, Xianfang Zeng, Xin Chen, Anqi Pang, Chi Zhang, Zhibin Wang, Bin Fu, Gang Yu, Ziwei Liu, and Liang Pan. Mvpaint: Synchronized multi-view diffusion for painting anything 3d. *arXiv preprint arxiv:2411.02336*, 2024. 3
- [7] 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. *Advances in Neural Information Processing Systems*, 36:35799–35813, 2023. 6
- [8] 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. 6
- [9] Kangle Deng, Timothy Omernick, Alexander Weiss, Deva Ramanan, Jun-Yan Zhu, Tinghui Zhou, and Maneesh Agrawala. Flashtex: Fast relightable mesh texturing with lightcontrolnet. In *Proceedings of European Conference on Computer Vision*, pages 90–107, 2024. 3
- [10] Lihe Ding, Shaocong Dong, Yaokun Li, Chenjian Gao, Xiao Chen, Rui Han, Yihao Kuang, Hong Zhang, Bo Huang, Zhanpeng Huang, Zibin Wang, Dan Xu, and Tianfan Xue. Fullpart: Generating each 3d part at full resolution, 2025. 4
- [11] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, Kyle Lacey, Alex Goodwin, Yannik Marek, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis. *arXiv preprint arXiv:2403.03206*, 2024. 2, 3
- [12] Yifei Feng, Mingxin Yang, Shuhui Yang, Sheng Zhang, Jia-ao Yu, Zibo Zhao, Yuhong Liu, Jie Jiang, and Chuncho Guo. Romantex: Decoupling 3d-aware rotary positional embedded multi-attention network for texture synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 17203–17213, 2025. 3
- [13] 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. *Advances in neural information processing systems*, 30, 2017. 6
- [14] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022. 6
- [15] 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, 6, 7
- [16] Sadeep Jayasumana, Srikumar Ramalingam, Andreas Veit, Daniel Glasner, Ayan Chakrabarti, and Sanjiv Kumar. Rethinking fid: Towards a better evaluation metric for image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9307–9315, 2024. 6
- [17] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering, 2023. 3
- [18] Chung Min Kim, Mingxuan Wu, Justin Kerr, Ken Goldberg, Matthew Tancik, and Angjoo Kanazawa. Garfield: Group anything with radiance fields, 2024. 4
- [19] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollar, and Ross Girshick. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4015–4026, 2023. 4
- [20] Black Forest Labs, Stephen Batifol, Andreas Blattmann, Frederic Boesel, Saksham Consul, Cyril Diagne, Tim Dockhorn, Jack English, Zion English, Patrick Esser, Sumith Kulal, Kyle Lacey, Yam Levi, Cheng Li, Dominik Lorenz, Jonas Müller, Dustin Podell, Robin Rombach, Harry Saini, Axel Sauer, and Luke Smith. Flux.1 kontext: Flow matching for in-context image generation and editing in latent space, 2025. 3, 4, 6
- [21] Weiyu Li, Xuanyang Zhang, Zheng Sun, Di Qi, Hao Li, Wei Cheng, Weiwei Cai, Shihao Wu, Jiarui Liu, Zihao Wang, et al. Step1x-3d: Towards high-fidelity and controllable generation of textured 3d assets. *arXiv preprint arXiv:2505.07747*, 2025. 2, 3, 6, 7
- [22] Zihao Li, Yufei Wang, Heliang Zheng, Yihao Luo, and Bi-han Wen. Sparc3d: Sparse representation and construction for high-resolution 3d shapes modeling, 2025. 3
- [23] Yixin Liang, Kunming Luo, Xiao Chen, Rui Chen, Hongyu Yan, Weiyu Li, Jiarui Liu, and Ping Tan. Unitex: Universal high fidelity generative texturing for 3d shapes. *arXiv preprint arXiv:2505.23253*, 2025. 2, 3, 4, 6, 7
- [24] Chen-Hsuan Lin, Jun Gao, Luming Tang, Towaki Takikawa, Xiaohui Zeng, Xun Huang, Karsten Kreis, Sanja Fidler, Ming-Yu Liu, and Tsung-Yi Lin. Magic3d: High-resolution text-to-3d content creation. In *CVPR*, 2023. 3
- [25] Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling, 2023. 3, 4
- [26] Jialun Liu, Chenming Wu, Xinqi Liu, Xing Liu, Jinbo Wu, Haotian Peng, Chen Zhao, Haocheng Feng, Jingtuo Liu, and

- Errui Ding. Texoct: Generating textures of 3d models with octree-based diffusion. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4284–4293, 2024. 3
- [27] Minghua Liu, Mikaela Angelina Uy, Donglai Xiang, Hao Su, Sanja Fidler, Nicholas Sharp, and Jun Gao. Partfield: Learning 3d feature fields for part segmentation and beyond, 2025. 3, 4, 5
- [28] Xiaoxiao Long, Yuan-Chen Guo, Cheng Lin, Yuan Liu, Zhiyang Dou, Lingjie Liu, Yuexin Ma, Song-Hai Zhang, Marc Habermann, Christian Theobalt, et al. Wonder3d: Single image to 3d using cross-domain diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9970–9980, 2024. 2
- [29] Changfeng Ma, Yang Li, Xinhao Yan, Jiachen Xu, Yunhan Yang, Chunshi Wang, Zibo Zhao, Yanwen Guo, Zhuo Chen, and Chunchao Guo. P3-sam: Native 3d part segmentation, 2025. 4
- [30] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis, 2020. 3
- [31] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952*, 2023. 2
- [32] Ben Poole, Ajay Jain, Jonathan T. Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv*, 2022. 2, 3
- [33] Elad Richardson, Gal Metzer, Yuval Alaluf, Raja Giryes, and Daniel Cohen-Or. Texture: Text-guided texturing of 3d shapes, 2023. 3
- [34] 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. 2
- [35] Nuri Ryu, Jiyun Won, Jooeun Son, Minsu Gong, Joo-Haeng Lee, and Sunghyun Cho. Elevating 3d models: High-quality texture and geometry refinement from a low-quality model. In *Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers*, page 1–12. ACM, 2025. 3
- [36] 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. 2
- [37] George Tang, William Zhao, Logan Ford, David Benhaim, and Paul Zhang. Segment any mesh, 2025. 4
- [38] Tencent Hunyuan3D Team. Hunyuan3d 2.1: From images to high-fidelity 3d assets with production-ready pbr material. *arXiv preprint arXiv:2506.15442*, 2025. 3, 4, 6, 7
- [39] Tencent Hunyuan3D Team. Hunyuan3d 2.5: Towards high-fidelity 3d assets generation with ultimate details. *arXiv preprint arXiv:2506.16504*, 2025. 2, 3
- [40] Zhengyi Wang, Cheng Lu, Yikai Wang, Fan Bao, Chongxuan Li, Hang Su, and Jun Zhu. Prolificdreamer: High-fidelity and diverse text-to-3d generation with variational score distillation. In *NeurIPS*, 2023. 2
- [41] 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, 2025. 3
- [42] Bojun Xiong, Jialun Liu, Jiakui Hu, Chenming Wu, Jinbo Wu, Xing Liu, Chen Zhao, Errui Ding, and Zhouhui Lian. Texgaussian: Generating high-quality pbr material via octree-based 3d gaussian splatting, 2025. 3
- [43] Yunhan Yang, Yufan Zhou, Yuan-Chen Guo, Zi-Xin Zou, Yukun Huang, Ying-Tian Liu, Hao Xu, Ding Liang, Yan-Pei Cao, and Xihui Liu. Omnipart: Part-aware 3d generation with semantic decoupling and structural cohesion, 2025. 4
- [44] Mingqiao Ye, Martin Danelljan, Fisher Yu, and Lei Ke. Gaussian grouping: Segment and edit anything in 3d scenes, 2024. 4
- [45] Haiyang Ying, Yixuan Yin, Jinzhi Zhang, Fan Wang, Tao Yu, Ruqi Huang, and Lu Fang. Omniseg3d: Omniprversal 3d segmentation via hierarchical contrastive learning, 2023. 4
- [46] Jonathan Young. xatlas, 2016. 3
- [47] Kim Youwang, Tae-Hyun Oh, and Gerard Pons-Moll. Paintit: Text-to-texture synthesis via deep convolutional texture map optimization and physically-based rendering. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 3
- [48] Xin Yu, Peng Dai, Wenbo Li, Lan Ma, Zhengze Liu, and Xiaojuan Qi. Texture generation on 3d meshes with point-uv diffusion, 2023. 3
- [49] Xin Yu, Ze Yuan, Yuan-Chen Guo, Ying-Tian Liu, Jianhui Liu, Yangguang Li, Yan-Pei Cao, Ding Liang, and Xiaojuan Qi. Texgen: a generative diffusion model for mesh textures. *ACM Transactions on Graphics*, 43(6):1–14, 2024. 3
- [50] Ze Yuan, Xin Yu, Yangtian Sun, Yuan-Chen Guo, Yan-Pei Cao, Ding Liang, and Xiaojuan Qi. Seqtex: Generate mesh textures in video sequence, 2025. 2, 3
- [51] Xianfang Zeng, Xin Chen, Zhongqi Qi, Wen Liu, Zibo Zhao, Zhibin Wang, Bin Fu, Yong Liu, and Gang Yu. Paint3d: Paint anything 3d with lighting-less texture diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4252–4262, 2024. 3
- [52] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3836–3847, 2023. 2
- [53] 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*, 43(4):1–20, 2024. 2
- [54] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018. 6
- [55] Yuqing Zhang, Yuan Liu, Zhiyu Xie, Lei Yang, Zhongyuan Liu, Mengzhou Yang, Runze Zhang, Qilong Kou, Cheng Lin,

- Wenping Wang, et al. Dreammat: High-quality pbr material generation with geometry-and light-aware diffusion models. *ACM Transactions on Graphics*, 43(4):1–18, 2024. [3](#)
- [56] Yuqing Zhang, Hao Xu, Yiqian Wu, Sirui Chen, Sirui Lin, Xiang Li, Xifeng Gao, and Xiaogang Jin. Aligntex: Pixel-precise texture generation from multi-view artwork. *ACM Transactions on Graphics (TOG)*, 44(4):1–12, 2025. [3](#)
- [57] Yibo Zhang, Li Zhang, Rui Ma, and Nan Cao. Texverse: A universe of 3d objects with high-resolution textures, 2025. [6](#)
- [58] Zibo Zhao, Zeqiang Lai, Qingxiang Lin, Yunfei Zhao, Haolin Liu, Shuhui Yang, Yifei Feng, Mingxin Yang, Sheng Zhang, Xianghui Yang, et al. Hunyuan3d 2.0: Scaling diffusion models for high resolution textured 3d assets generation. *arXiv preprint arXiv:2501.12202*, 2025. [2](#)
- [59] Lingting Zhu, Jingrui Ye, Runze Zhang, Zeyu Hu, Yingda Yin, Lanjiong Li, Jinnan Chen, Shengju Qian, Xin Wang, Qingmin Liao, and Lequan Yu. Muma: 3d pbr texturing via multi-channel multi-view generation and agentic post-processing. *arXiv preprint arXiv:2503.18461*, 2025. [2](#), [3](#)