TEX4D: ZERO-SHOT 4D SCENE TEXTURING WITH VIDEO DIFFUSION MODELS

Anonymous authors
Paper under double-blind review

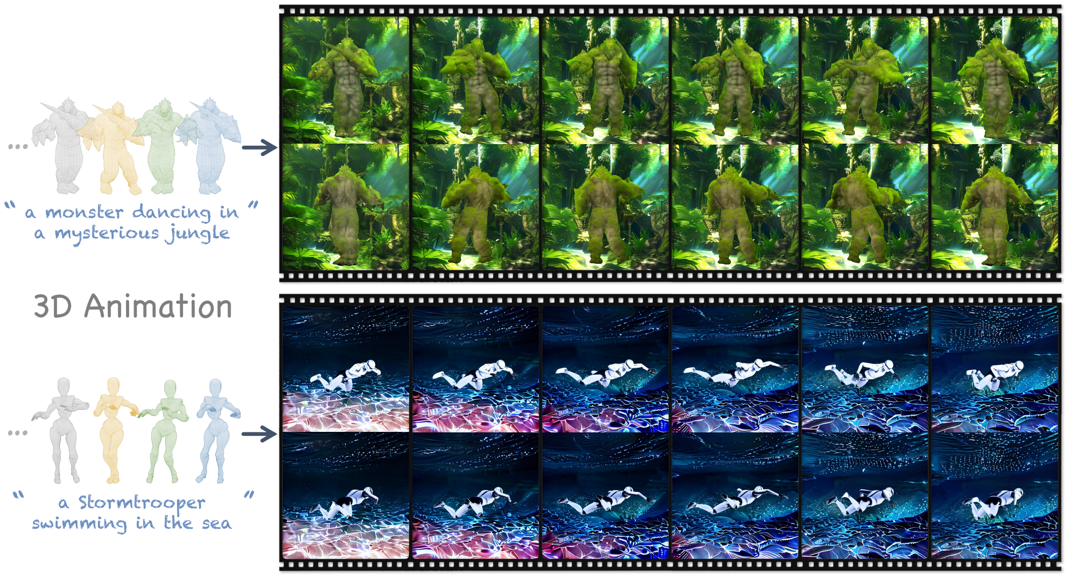


Figure 1. Given an untextured mesh sequence and a text prompt as inputs (Left), **Tex4D** generates multi-view, temporally consistent textures along with a dynamic background. On the right, we show renderings of the textured meshes from two different perspectives. Zoom in to view the texture details.

ABSTRACT

3D meshes are widely used in computer vision and graphics because of their efficiency in animation and minimal memory footprint. They are extensively employed in movies, games, AR, and VR, leading to the creation of a vast number of mesh sequences. However, creating temporally consistent and realistic textures for these mesh sequences to model the appearance transformations remains labor-intensive for professional artists. On the other hand, video diffusion models have demonstrated remarkable capabilities in text-driven video generation, enabling users to create countless video clips based solely on their imagination. Despite their strengths, these models often lack 3D geometry awareness for fine-grained video control and struggle with achieving multi-view consistent texturing for 3D mesh sequences. In this work, we present **Tex4D**, a zero-shot approach that integrates inherent 3D geometry knowledge from mesh sequences with the expressiveness of video diffusion models. Given an untextured mesh sequence and a text prompt as inputs, our method enhances multi-view consistency by synchronizing the diffusion process across different views through latent aggregation in the UV space. To ensure temporal consistency such as lighting changes, wrinkles, and appearance transformations, we leverage prior knowledge from a conditional video generation model for texture synthesis. However, straightforwardly combining the video diffusion model and the UV texture aggregation leads to blurry results. We analyze the underlying causes and propose a simple yet effective modification to the DDIM sampling process to address this issue. Additionally, we introduce a reference latent texture to strengthen the correlation between frames during the denoising process. To the best of our knowledge, Tex4D is the first method specifically designed for 4D scene texturing. Extensive experiments demonstrate its superiority in producing multi-view and multi-frame consistent videos based on untextured mesh sequences.

1 Introduction

3D meshes are widely used in modeling, computer-aided design (CAD), animation, and computer graphics due to their low memory footprint and efficiency in animation. Visual artists, game designers, and movie creators build numerous animated mesh sequences for visual applications. However, creating vivid videos involves complex post-processing steps, such as lighting controls and appearance transformations. These steps are labor-intensive and require specialized expertise by artists.

On the other hand, recent advancements in generative models have democratized content creation and demonstrated impressive performance in image and video synthesis. For instance, video generation models (Ho et al., 2022; Esser et al., 2023; Li et al., 2023; He et al., 2022; Yu et al., 2023a; Zhou et al., 2022; Hong et al., 2022; Yang et al., 2024; Zhang et al., 2023b; Xing et al., 2023; Chen et al., 2023c; 2024) trained on large-scale video datasets (Bain et al., 2021; Schuhmann et al., 2021) allow users to create realistic video clips from various inputs such as text prompts, images, or geometric conditions. However, these text-to-video generation models, which are trained solely on 2D data, often struggle with spatial consistency when applied to multi-view image generation (Tang et al., 2023; Shi et al., 2023b; Liu et al., 2023a; Weng et al., 2023; Long et al., 2023; Shi et al., 2023a; Kwak et al., 2023; Tang et al., 2024; Voleti et al., 2024) or 3D object texturing (Cao et al., 2023; Liu et al., 2023b; Richardson et al., 2023; Chen et al., 2023b; Huo et al., 2024).

To address these limitations, two main approaches have been developed. One approach (Richardson et al., 2023; Chen et al., 2023b; Cao et al., 2023) focuses on resolving multi-view inconsistency in static 3D object texturing by synchronizing multi-view image diffusion processes and enforcing UV space consistency. While these methods produce multi-view consistent textures for static 3D objects, they do not address the challenge of generating temporally consistent textures for mesh sequences. Another approach (Guo et al., 2023a; Lin et al., 2024; Peng et al., 2024) aims to generate temporally consistent video clips based on the rendering (e.g., depth, normal or UV maps) of an untextured mesh sequence. To encourage temporal consistency, these methods modify the attention mechanism in 2D diffusion models and utilize inherent correspondences in a mesh sequence to facilitate feature synchronization between frames. Although these techniques can be adapted for multi-view image generation by treating camera pose movement as temporal motion, they usually produce inconsistent 3D texturing due to insufficient exploitation of 3D geometry priors.

In this paper, we introduce a novel task: 4D scene texturing. Given an animated untextured 3D mesh sequence and a text prompt, our goal is to generate textures that are both temporally and multi-view consistent. Our objective is to texture 4D scenes while capturing temporal variations, such as lighting changes and wrinkles, to produce vivid visual results—a key requirement in downstream tasks like character generation. Different from existing works, we fully leverage 3D geometry knowledge from the mesh sequence to enforce multi-view consistency. Specifically, we develop a method that synchronizes the diffusion process from different views through latent aggregation in the UV space. To ensure temporal consistency, we employ prior knowledge from a conditional video generation model for texture synthesis and introduce a reference latent texture to enhance frame-to-frame correlations during the denoising process. However, naively integrating the UV texture aggregation into the video diffusion process causes the variance shift problem, leading to blurry results. To resolve this issue, we propose a simple yet effective modification to the DDIM (Song et al., 2020) sampling process by uniformly transforming the equation. Additionally, we propose to synthesize a dynamic background along with the textures of the given mesh sequences, which not only creates a complete 4D scene but also fully exploits the prior knowledge embedded in the video diffusion model. Our method is computationally efficient thanks to its zero-shot nature. The textured mesh sequence can be rendered from any camera view, thus supporting a wide range of applications in content creation.

We evaluate our method on various mesh sequences with key contributions as follows:

- We present **Tex4D**, a zero-shot pipeline for generating high-fidelity textures that are temporally and multi-view consistent, utilizing text-to-video diffusion models and mesh sequence controls.
- We develop a simple and effective modification to the DDIM sampling process to address the variance shift issue caused by multi-view texture aggregation.
- We introduce a reference UV blending mechanism to establish correlations during the denoising steps, addressing self-occlusions, and synchronizing the diffusion process in invisible regions.
- Our method is not only computationally efficient, but also demonstrates comparable if not superior performance to various state-of-the-art baselines.

2 Related Work

Video Stylization and Editing Text-to-video diffusion models have shown remarkable performance in the field of video generation. These models learn motions and dynamics from large-scale video datasets using 3D-UNet to create high-quality, realistic, and temporally coherent videos. Although these approaches show compelling results, the generated videos lack fine-grained control, inhibiting their application in stylization and editing. To solve this issue, inspired by ControlNet (Zhang et al., 2023a), SparseCtrl (Guo et al., 2023a) trains a sparse encoder from scratch using frame masks and sparse conditioning images as input to guide a pre-trained video diffusion model. CTRL-Adapter (Lin et al., 2024) proposes a trainable intermediate adapter to efficiently connect the features between ControlNet and video diffusion models.

Meanwhile, Tumanyan et al. (2023) observed that the spatial features of T2I models play an influential role in determining the structure and appearance, Text2Video-Zero (Khachatryan et al., 2023) uses a frame-warping method to animate the foreground object by T2I models and Wu et al. (2023); Ceylan et al. (2023); Qi et al. (2023) propose utilizing self-attention injection and cross-frame attention to generate stylized and temporally consistent video using DDIM inversion (Song et al., 2020). Subsequently, numerous works (Zhang et al., 2023c; Cai et al., 2024; Yang et al., 2023; Geyer et al., 2023; Eldesokey & Wonka, 2024) generate temporally consistent videos utilizing T2I diffusion models by spatial latent alignment without training. However, the synthesized videos usually show flickerings due to the empirical correspondences, such as feature embedding distances and UV maps, which are insufficient to express the continuous content in the latent space. Another line of work (Singer et al., 2022; Bar-Tal et al., 2022; Blattmann et al., 2023; Xu et al., 2024; Guo et al., 2023b) is to train additional modules on large-scale video datasets to construct feature mappings, for example, Text2LIVE (Bar-Tal et al., 2022) applies test-time training with the CLIP loss, and MagicAnimate (Xu et al., 2024) introduced an appearance encoder to retain intricate clothes details.

Texture Synthesis With the rapid development of foundation models, researchers have focused on applying their generation capability and adaptability to simplify the process of designing textures and reduce the expertise required. To incorporate the result 3D content with prior knowledge, earlier works (Khalid et al., 2022; Michel et al., 2021; Chen et al., 2022) jointly optimize the meshes and textures from scratch with the simple semantic loss from the pre-trained CLIP (Radford et al., 2021) to encourage the 3D alignment between the generated results and the semantic priors. However, the results show apparent artifacts and distortion because the semantic feature cannot provide fine-grained supervision during the generation of 3D content.

DreamFusion (Poole et al., 2022) and similar models (Lin et al., 2023; Wang et al., 2023; Po & Wetzstein, 2024; Metzer et al., 2022; Chen et al., 2023a) distill the learned 2D diffusion priors from the pre-trained diffusion models (Rombach et al., 2021) to synthesize the 3D content by Score Distillation Sampling (SDS). These methods render 2D projections of the 3D asset parameters and compare them against reference images, iteratively refining the 3D asset parameters to minimize the discrepancy of the target distribution of 3D shapes learned by the diffusion model. Although these approaches enable people without expertise to generate detailed 3D content by textual prompt, their results are typically over-saturated and over-smoothed, hindering their application in actual cases. Another line of optimization-based methods (Yu et al., 2023b; Zeng et al., 2024; Bensadoun et al., 2024) turned to fuse 3D shape information, such as vertex positions, depth maps and normal maps, with the pre-trained diffusion model by training separate modules on 3D datasets. Still, they require a specific UV layout process to achieve plausible results.

Recently, TexFusion (Cao et al., 2023) and numerous zero-shot methods (Liu et al., 2023b; Richardson et al., 2023; Chen et al., 2023b; Huo et al., 2024) have shown significant success in generating globally consistent textures without additional 3D datasets. Based on depth-aware diffusion models, they sequentially inpaint the latents in the UV domain to ensure the spatial consistency of latents observed across different views. Then, they decode the latents from multiple views and finally synthesize the RGB texture through backprojection.

However, these methods primarily focus on generating static 3D assets and do not account for temporal changes in the final visual presentation, such as in videos. Our work introduces a zero-shot framework that enables multi-view consistent video generation based on animated meshes, which is effective in capturing temporal variations. To the best of our knowledge, this is the first approach to synthesize multi-view and multi-frame consistent textures for mesh sequences.

3 PRELIMINARIES

Video Diffusion Prior. In this paper, we adopt CTRL-Adapter (Lin et al., 2024) as our prior model to provide dynamic information. CTRL-Adapter aims to adapt a pre-trained text-to-video diffusion model to condition various types of images such as depth or normal map sequences. The key idea behind CTRL-Adapter is to leverage a pre-trained ControlNet (Zhang et al., 2023a) and to align its latents with those of the video diffusion model through a learnable mapping module. Intuitively, the video diffusion model generates temporally consistent video frames that capture dynamic elements like character motions and environmental lighting, while the ControlNet further enhances this capability by allowing the model to condition on geometric information, such as depth and normal map sequences. This makes CTRL-Adapter particularly effective in providing a temporally consistent texture prior for our 4D scene texturing task. Specifically, we leverage the depth-conditioned CTRL-Adapter model. Given a sequence of depth images denoted as $\{D_1, ..., D_K\}$ and a text prompt $\boldsymbol{\mathcal{P}}$, CTRL-Adapter (denoted as $\boldsymbol{\mathcal{C}}$) synthesizes a frame sequence F by $F = \boldsymbol{\mathcal{C}}(\{D_1, ..., D_K\}, \boldsymbol{\mathcal{P}})$.

DDIM Sampling. DDIM (Song et al., 2020) is a widely used sampling method in diffusion models due to its superior efficiency and deterministic nature compared to DDPM (Ho et al., 2020). To enhance numerical stability and prevent temporal color shifts in Video Diffusion Models (VDMs), numerous models (Zhang et al., 2023b; Ho et al., 2022) employ a learning-based sampling technique known as v-prediction (Salimans & Ho, 2022). At each denoising step, the DDIM sampling process for the latents (denoted as $\boldsymbol{z}_t$) can be described as follows:

$$\boldsymbol{z}_{t-1} = \sqrt{\alpha_{t-1}} \cdot \hat{\boldsymbol{z}}_0(\boldsymbol{z}_t) + \sqrt{1-\alpha_{t-1}} \cdot \boldsymbol{\epsilon}_\theta(\boldsymbol{z}_t), \tag{1}$$

$$\hat{\boldsymbol{z}}_0(\boldsymbol{z}_t) = \frac{\boldsymbol{z}_t - \sqrt{1-\alpha_t} \cdot \boldsymbol{\epsilon}_\theta}{\sqrt{\alpha_t}}, \quad \boldsymbol{\epsilon}_\theta(\boldsymbol{z}_t) = \boldsymbol{\epsilon}_\theta, \tag{2}$$

where α_t represents the noise variance at time step t, $\boldsymbol{\epsilon}_\theta$ is the estimated noise from the U-Net denoising module, which is expected to follow $\mathcal{N}(0, \mathcal{I})$, and $\hat{\boldsymbol{z}}_0(\boldsymbol{z}_t)$ denotes the predicted original sample (i.e., the latents at timestep 0). After the v-parameterization, the predicted original sample $\hat{\boldsymbol{z}}_0(\boldsymbol{z}_t)$ and the predicted epsilon $\boldsymbol{\epsilon}_\theta(\boldsymbol{z}_t)$ are computed as follows:

$$\hat{\boldsymbol{z}}_0(\boldsymbol{z}_t) = \sqrt{\alpha_t} \cdot \boldsymbol{z}_t - \sqrt{1-\alpha_t} \cdot \boldsymbol{\epsilon}_\theta, \quad \boldsymbol{\epsilon}_\theta(\boldsymbol{z}_t) = \sqrt{\alpha_t} \cdot \boldsymbol{\epsilon}_\theta + \sqrt{1-\alpha_t} \cdot \boldsymbol{z}_t. \tag{3}$$

In this paper, we leverage an enhanced DDIM sampling process in video diffusion models, along with a multi-view consistent texture aggregation mechanism to synthesize 4D textures.

4 METHOD

Given an untextured mesh animation and a text prompt, our goal is to generate multi-view and multi-frame consistent texture for each mesh that aligns with both the text description and motion cues, meanwhile capturing the dynamics from video diffusion models. To optimize computational efficiency, instead of processing all video frames, we uniformly sample K key frames from the video and synthesize textures specifically for these key frames. The textures for the remaining frames are then generated by interpolating the key frame textures. Formally, given K animated meshes at the key frames ($\{M_1, ..., M_K\}$), along with a text description $\boldsymbol{\mathcal{P}}$, our method produces a sequence of temporally and spatially consistent UV maps denoted as $\{UV_1, ..., UV_K\}$, in a zero-shot manner.

Previous texture generation methods (Richardson et al., 2023; Chen et al., 2023b; Cao et al., 2023) typically inpaint and update textures sequentially using pre-defined camera views in an incremental manner. However, these approaches rely on view-dependent depth conditions and lack global spatial consistency, often resulting in visible discontinuities in the assembled texture map. This issue arises from error accumulation during the autoregressive view update process, as noted by Bensadoun et al. (2024). To resolve these issues, rather than processing each view independently, recent methods (Liu et al., 2023b; Huo et al., 2024; Zhang et al., 2024) propose to generate multi-view textures simultaneously through diffusion, and then aggregate them in the UV space at each diffusion step. In this work, we similarly leverage the UV space as an intermediate representation to ensure multi-view consistency during texture generation.

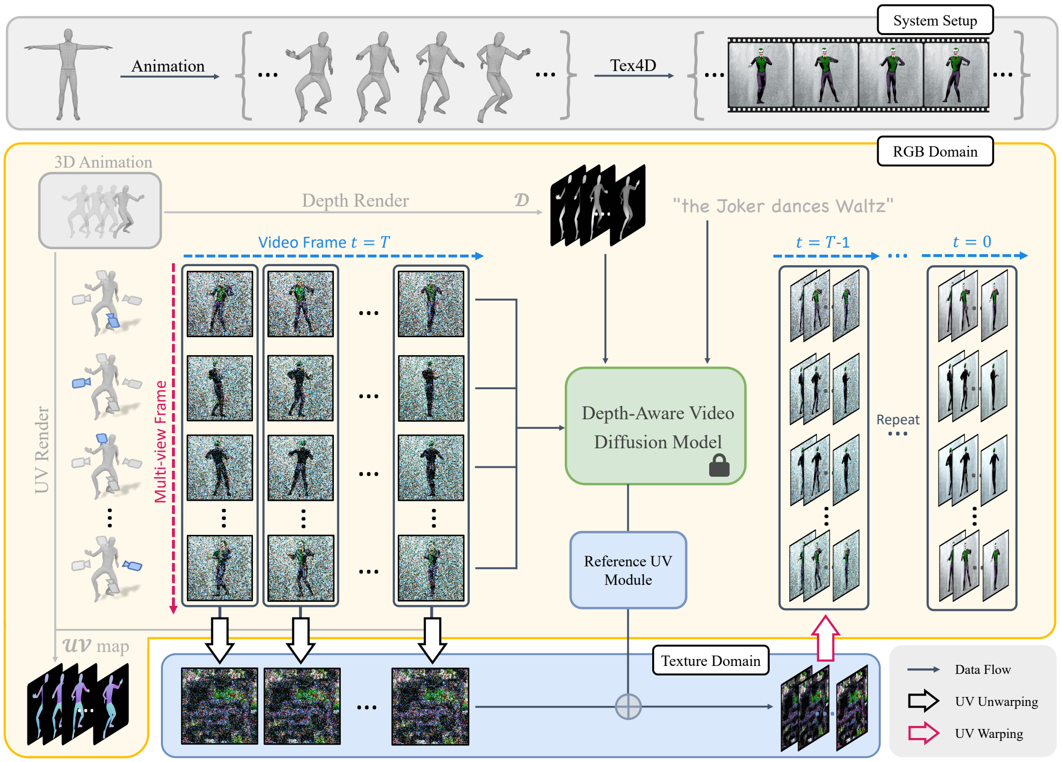
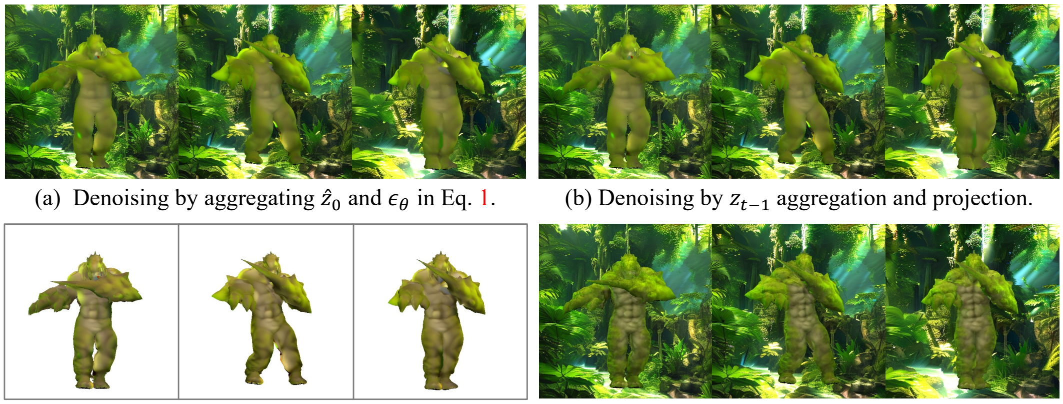


Figure 2. **Overview of our pipeline.** Given a mesh sequence and a text prompt as inputs, Tex4D generates a UV-parameterized texture sequence that is both globally and temporally consistent, aligning with the prompt and the mesh sequence. We sample multi-view video sequences using a depth-aware video diffusion model. At each diffusion step, latent views are aggregated into UV space, followed by multi-view latent texture diffusion to ensure global consistency. To maintain temporal coherence and address self-occlusions, a Reference UV Blending module is applied at the end of each step. Finally, the latent textures are back-projected and decoded to produce RGB textures for each frame.

4.1 Overview

As shown in Fig. 2, given a sequence of K meshes, we start by rendering the mesh at V predefined, uniformly sampled camera poses to obtain multi-view depth images (denoted as $\{D_{1,1}, ..., D_{1,K}, D_{2,1} ..., D_{V,K}\}$), which serve as the geometric conditions. To generate textures for each mesh, we initialize $V \times K$ noise images sampled from a Normal distribution (denoted as $\{\boldsymbol{z}^{1,1}, ..., \boldsymbol{z}^{1,K}, \boldsymbol{z}^{2,1}, ..., \boldsymbol{z}^{V,K}\}$). Additionally, we initialize an extra noise map sequence $\{\boldsymbol{z}_b^1, ..., \boldsymbol{z}_b^K\}$ for the backgrounds learning. This noise map corresponds to the texture of a plane mesh that is composited with the foreground object at each diffusion step (See Sec. 4.3). Next, for each view $v \in \{1, ..., V\}$, we apply the video diffusion model (Lin et al., 2024) discussed in Sec. 3 to simultaneously denoise all latents and obtain multi-frame consistent images as $\{I^{1,v}, ..., I^{K,v}\} = \boldsymbol{\mathcal{C}}(\{D_{1,v}, ..., D_{K,v}\}, \boldsymbol{\mathcal{P}})$, where $\boldsymbol{\mathcal{P}}$ is the provided text prompt. Finally, we un-project and aggregate all denoised multi-view images for each mesh to formulate temporally consistent UV textures.

However, applying the video diffusion model independently to each camera view often results in multi-view inconsistencies. Inspired by (Liu et al., 2023b; Huo et al., 2024; Zhang et al., 2024), we aggregate the multi-view latents of each mesh in the UV space to merge observations across different views at each denoising step. We then render latent from the latent texture to ensure multi-view consistency. To simultaneously generate a dynamic background and fully exploit prior in the video diffusion model, we composite the rendered foreground latents with the background latents at each diffusion step. This aggregation process is discussed in detail in Sec. 4.2. Nonetheless, such a simple aggregation method introduces blurriness in the final results. In Sec. 4.3, we analyze the underlying causes and propose a simple yet effective method to enhance the denoising process. Additionally, we create and leverage a reference UV to handle self-occlusions and further improve temporal consistency in Sec. 4.4.



(a) Denoising by aggregating $\hat{z}_0$ and ϵ_θ in Eq. 1. (b) Denoising by z_{t-1} aggregation and projection.

(c) Ours w/o background priors. (d) Our results.

Figure 3. **Ablation studies on the multi-view denoising algorithm and backgrounds.** (a) Aggregating $\hat{\boldsymbol{z}}_0(\boldsymbol{z}_t)$, $\boldsymbol{\epsilon}_\theta(\boldsymbol{z}_t)$ in Eq. 1 into UV space. (b) Aggregating $\boldsymbol{z}_{t-1}$ in Eq. 1 into UV space. (c) Replacing learnable background with white background. (d) Our results. See Sec. 5.3 for details.

4.2 Multi-view Latents Aggregation in the UV Space

We describe how to aggregate multi-view latent maps in the UV space. Taking frame $k \in \{1, ..., K\}$ as an example, we aggregate the multi-view latents $\{\boldsymbol{z}^{1,k}, \ldots, \boldsymbol{z}^{V,k}\}$ in the UV space by:

$$\mathcal{T}^k\left(\boldsymbol{z}^k\right) = \frac{\sum_{v=1}^{V} \mathcal{R}^{-1}(\boldsymbol{z}^{v,k}, c_v) \odot \cos\left(\theta^v\right)^\alpha}{\sum_{v=1}^{V} \cos\left(\theta^v\right)^\alpha}, \tag{4}$$

where $\mathcal{R}^{-1}$ represents the inverse rendering operator that un-projects the latents to the UV space, thus $\mathcal{R}^{-1}(\boldsymbol{z}^{v,k}, \boldsymbol{c_v})$ produces a partial latent UV texture from view v, $\cos(\theta^v)$ is the cosine map buffered by the geometry shader, recording the cosine value between the view direction and the surface normal for each pixel, α is a scaling factor, and $\boldsymbol{c_v}$ denotes one of the predefined cameras. After multi-view latents aggregation, we obtain multi-view consistent latents by rendering the aggregated UV latent map using $\tilde{\boldsymbol{z}}^{v,k} = \mathcal{R}\left(\mathcal{T}^k; \boldsymbol{c}_v\right)$, where $\mathcal{R}$ is the rendering operation.

4.3 Multi-frame Consistent Texture Generation

The aggregation process discussed above yields multi-view consistent latents $\{\tilde{\boldsymbol{z}}^{v,k}\}$ for the subsequent denoising steps. However, this simple aggregation and projection strategy leads to a blurry appearance as shown in Fig. 3(b). This issue arises primarily because the aggregation process depicted in Eq. 4 derails the DDIM denoise process. Specifically, the estimated noise $\boldsymbol{\epsilon}_\theta(\boldsymbol{z}_t)$ for each step in Eq. 1 is expected to follow $\mathcal{N}(0, \mathcal{I})$, but Eq. 4 indicates that after aggregating multi-view latents, the expected norm of variance of the noise distribution would be less than $\mathcal{I}$. We denote this as the "variance shift" issue caused by the texture aggregation.

To resolve this issue, we rewrite the estimated noise ϵ_θ for a latent as the combination of the t-step latent $\boldsymbol{z}_t$ and the estimated latent $\hat{z}_0(\boldsymbol{z}_t)$ at step 0, thus the v-paramaterized predicted epsilon $\boldsymbol{\epsilon}_\theta(\boldsymbol{z}_t)$ in Eq. 3 can be equally expressed as follow:

$$\begin{aligned}
\boldsymbol{\epsilon_\theta} &= \left(\sqrt{\alpha_t} \cdot \boldsymbol{z}_t - \hat{\boldsymbol{z}}_0(\boldsymbol{z}_t)\right) / \sqrt{1-\alpha_t} \\
\boldsymbol{\epsilon}_\theta(\boldsymbol{z}_t) &= \sqrt{\alpha_t} \cdot \boldsymbol{\epsilon_\theta} + \sqrt{1-\alpha_t} \cdot \boldsymbol{z}_t \\
&= \sqrt{\frac{\alpha_t}{1-\alpha_t}} \cdot \left(\sqrt{\alpha_t}\boldsymbol{z}_t - \hat{\boldsymbol{z}}_0(\boldsymbol{z}_t)\right) + \sqrt{1-\alpha_t} \cdot \boldsymbol{z}_t.
\end{aligned} \tag{5}$$

In practice, we carry out this denoising technique in the UV space. Specifically, we first compute the original texture map (i.e., texture map at step 0, denoted as $\hat{\mathcal{T}}_0$) by aggregating the predicted original multi-view image latents through Eq. 4. The noisy latent texture map at time step t (denoted as $\mathcal{T}_t$) can be similarly computed. We then run one desnoising step by:

$$\mathcal{T}_{t-1} = \sqrt{\alpha_{t-1}} \cdot \hat{\mathcal{T}}_0 + \sqrt{1-\alpha_{t-1}} \left(\sqrt{\frac{\alpha_t}{1-\alpha_t}} \cdot (\sqrt{\alpha_t}\mathcal{T}_t - \hat{\mathcal{T}}_0) + \sqrt{1-\alpha_t} \cdot \mathcal{T}_t \right). \tag{6}$$

Through experimentation, we observe that background optimization plays a crucial role in fully exploiting the prior within the video diffusion model. As shown in Fig. 3(c), using a simple white background leads to blurry results. This may be attributed to a mismatch between the white-background images and the training dataset, which likely contains fewer such examples, affecting the denoising process. To resolve this issue, we compute the final latents as the combination of the foreground latent $\tilde{\boldsymbol{z}}_{t-1}$ projected from the aggregated UV latents and the residual background latent $\boldsymbol{z}_{b,t-1}$ denoised by diffusion models. Specifically, we composite the estimated latents in the $t-1$ step as follows:

$$\boldsymbol{z}_{t-1} = \tilde{\boldsymbol{z}}_{t-1} \odot \boldsymbol{\mathcal{M}}_{\text{fg}} + \boldsymbol{z}_{b,t-1} \odot (1 - \boldsymbol{\mathcal{M}}_{\text{fg}}), \quad \tilde{\boldsymbol{z}}_{t-1}, \boldsymbol{\mathcal{M}}_{\text{fg}} = \mathcal{R}\left(\mathcal{T}_{t-1}; \boldsymbol{c}_v\right), \tag{7}$$

where $\boldsymbol{\mathcal{M}}_{\text{fg}}$ represents the foreground mask of the mesh.

To summarize, our diffusion process starts with $K \times (V+1)$ randomly initialized noise maps sampled (i.e., $\{\boldsymbol{z}_T^{1,k}, \ldots, \boldsymbol{z}_T^{V,k}\}$, for foreground, $\{\boldsymbol{z}_b^1, ..., \boldsymbol{z}_b^K\}$ for background) and denoise them into images simultaneously. At each denoising step t with the key frame k, we derive the estimated noises $\{\boldsymbol{\epsilon}_{t-1}^{1,k}, \ldots, \boldsymbol{\epsilon}_{t-1}^{V,k}\}$ using the video diffusion model and calculate the estimated original latent $\{\hat{\boldsymbol{z}}_0^{1,k}, \ldots, \hat{\boldsymbol{z}}_0^{V,k}\}$ by Eq. 2. Then, we use Eq. 4 to aggregate the latents onto UV space. Next, we utilize Eq. 6 to take the diffusion step in the UV space, and render the synchronized latents $\{\tilde{\boldsymbol{z}}_{t-1}^{1,k}, \ldots, \tilde{\boldsymbol{z}}_{t-1}^{V,k}\}$ from latent UVs $\{\mathcal{T}_{t-1}^1, \ldots, \mathcal{T}_{t-1}^K\}$ to ensure multi-view consistency. Finally, we composite the denoised latent with the latents at step $t-1$ according to foreground masks by Eq. 7.

4.4 Reference UV Blending

While the video diffusion model ensures temporal consistency for latents from each view, consistency can sometimes diminish after aggregation in the texture domain. This issue primarily stems from the view-dependent nature of the depth conditions and the limited resolution of latents, which can lead to distortions when features from different camera angles are combined onto the UV texture. Additionally, self-occlusion during mesh animation often results in a loss of information in invisible regions.

To address these challenges, we introduce a reference UV map to provide additional correlations between latent textures across frames. Specifically, the reference UV map is constructed by sequentially combining latent textures over time, with each new texture filling only the empty texels of the reference UV map. Then, each texture is blended using the reference UV $\mathcal{T}_{\mathcal{UV}}$ with a mask $\boldsymbol{\mathcal{M}}_{\mathcal{UV}}$ that labels the visible region:

$$\mathcal{T}_t^k = \left((1-\lambda) \cdot \mathcal{T}_t^k + \lambda \cdot \mathcal{T}_{\mathcal{UV}}\right) \odot \boldsymbol{\mathcal{M}}_{\mathcal{UV}}^k + \mathcal{T}_{\mathcal{UV}} \odot \left(1 - \boldsymbol{\mathcal{M}}_{\mathcal{UV}}^k\right), \tag{8}$$

where λ is the blending weight for the reference UV in the visible region, while the invisible region is simply replaced with the reference texture. We empirically set the blending weight to 0.2 during our experiments.

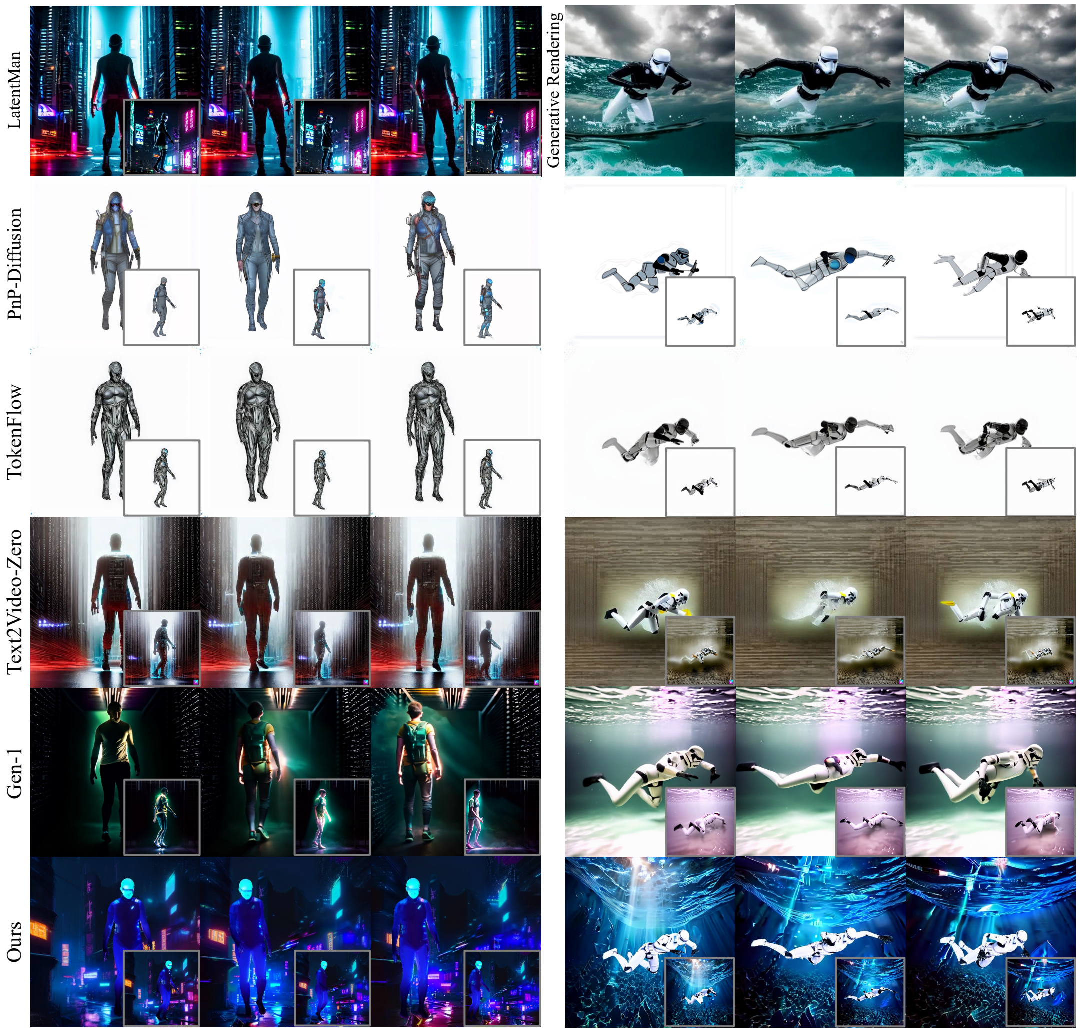
5 Experiments

Datasets. We sourced our datasets from two primary repositories: human motion diffusion outputs and the Mixamo[1] and Sketchfab[2] websites. We employed the text-to-motion diffusion model (HDM) (Tevet et al., 2023) to compare our approach with LatentMan (Eldesokey & Wonka, 2024), as LatentMan requires the SMPL model (Loper et al., 2015) to get corresponding features. For comparison with Generative Rendering (Cai et al., 2024), we obtained animated characters from the Mixamo platform and rendered them with different motions. Specifically, we first used Blender Community (2024) to extract meshes, joints, skinning weights, and animation data from the FBX files. Then, we applied linear blend skinning to animate the meshes. For meshes without UV maps, we utilized XATLAS to parameterize the mesh and unwrap the UVs.

Baselines. To the best of our knowledge, no existing studies directly address the task of multi-view consistent video generation guided by untextured mesh sequences, as our method does. Consequently, we adapted six recent methods and rendered the input (untextured mesh renders and

[1]https://www.mixamo.com/

[2]https://sketchfab.com/



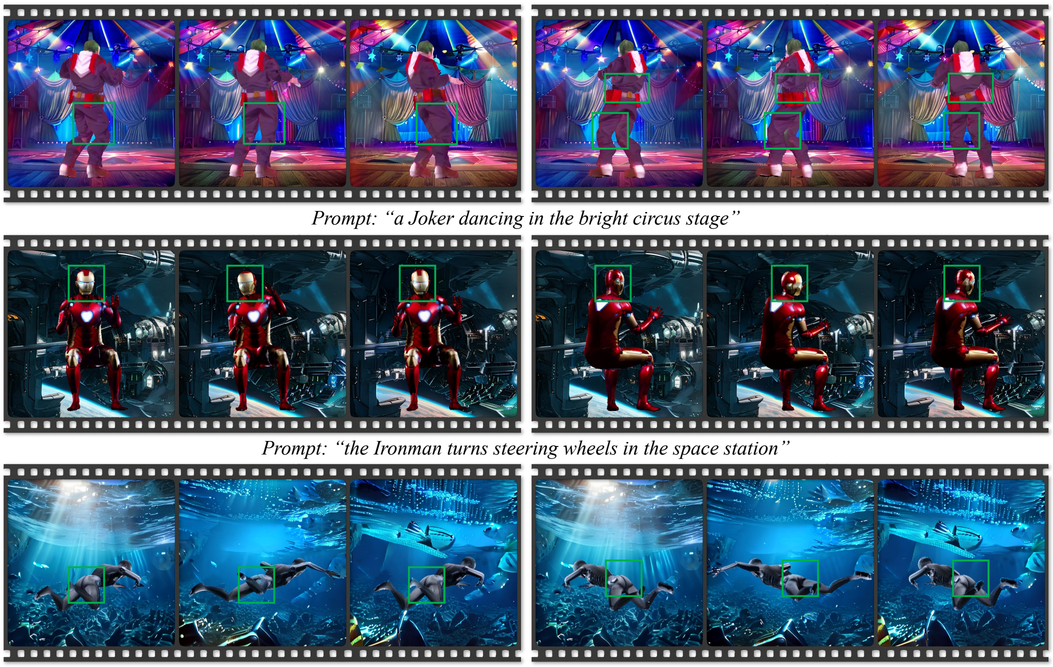
Prompt: "a cyberpunk walks" *Prompt:* "a Stormtrooper swimming"

Figure 4. **Qualitative comparisons of multi-view video generation.** We compare our method against Text2Video-Zero (Khachatryan et al., 2023), PnP-diffusion (Tumanyan et al., 2023), TokenFlow (Geyer et al., 2023), Gen-1 (Esser et al., 2023), LatentMan (Eldesokey & Wonka, 2024), and Generative Rendering (Cai et al., 2024). We generate videos in the front view and the side view (gray box) on SMPL model data (left column) and Mixamo dataset (right column). Our method manages to generate vivid videos that align with the textual prompts while preserving spatial and temporal consistency.

depth maps) according to their configurations to serve as baselines. These baselines include DDIM inversion-based video stylization methods and video generation methods with different control mechanisms, including PnP-Diffusion (Tumanyan et al., 2023), Text2Video-Zero (Khachatryan et al., 2023), TokenFlow (Geyer et al., 2023), Generative Rendering (Cai et al., 2024), LatentMan (Eldesokey & Wonka, 2024), and Gen-1 (Esser et al., 2023). PnP-Diffusion is an image style transfer method that is conditioned on the attention feature of the input image by DDIM inversion. We extended the method to stylize videos on a frame-by-frame basis for comparison, aligning with previous work (Geyer et al., 2023). Built upon cross-frame attention, Text2Video-Zero guides the video by warping latents to implicitly enhance video dynamics, and we utilized their official extension, which supports depth control. TokenFlow, Generative Rendering, and LatentMan study frame relations in latent space and establish feature correspondences through nearest neighbor matching, UV maps, and DensePose features, respectively. Gen-1 is a video-to-video model that learns the structure of input videos and transforms the input content (untextured mesh renders) into stylized outputs. Given the availability of the source code for Generative Rendering, we utilize the experimental results presented in their video demos for qualitative comparison.

Table 1. **Quantitative evaluation**. We present FVD values and a comparison highlighting the percentage of user preference for our approach against other methods. Our method shows the best spatio-temporal consistency as measured by the FVD metric (Unterthiner et al., 2018). Users consistently favored Tex4D over all baselines.

Method	FVD (↓)	Appearance Quality	Spatio-temporal Consistency	Consistency with Prompt
Text2Video-Zero	3078.94	89.33%	91.78%	91.55%
PnP-Diffusion	1390.04	86.42%	87.18%	89.74%
TokenFlow	1330.43	92.31%	86.84%	93.42%
Gen-1	3114.26	70.27%	75.00%	77.78%
LatentMan	2811.23	86.57%	86.57%	81.82%
Ours	**1303.14**	-	-	-



Prompt: "a Joker dancing in the bright circus stage"

Prompt: "the Ironman turns steering wheels in the space station"

Prompt: "a machinery swimming in the sea"

Figure 5. **Qualitative results.** Our method generates multi-view consistent foreground objects with a diverse set of styles and prompts. We highlight the temporal changes in the green boxes.

Evaluation Metric. Quantitatively assessing multi-view consistency and temporal coherence is still an unresolved problem. We perform a user study to assess overall performance, including the appearance, temporal coherence and spatial consistency, and the fidelity to prompt based on human preference. In addition, we compute the multi-view coherence via Fréchet Video Distance (FVD) (Unterthiner et al., 2018), a video-level metric that assesses temporal coherence, as utilized in previous approaches (Li et al., 2024; Xie et al., 2024).

5.1 Qualitative Results

We present qualitative evaluation in Fig. 4. LatentMan (Eldesokey & Wonka, 2024), Generative Rendering (Cai et al., 2024), TokenFlow (Geyer et al., 2023), and Text2Video-Zero (Khachatryan et al., 2023), which are based on T2I diffusion models with cross-frame attention mechanisms, exhibit significant flickering compared to other methods. This issue arises in part from the empirical and implicit correspondence mapping used to encourage the interframe latent consistency, and the correspondences in the latent space may not exactly match the RGB space. In contrast, our approach interpolates the frames between key frame textures in RGB space, effectively eliminating artifacts caused by latent manipulation. PnP-Diffusion (Tumanyan et al., 2023), which edits frames independently, generates detailed and sophisticated appearances but suffers from poor spatio-temporal consistency due to the loss of temporal correlations in the latent space. While Gen-1 (Esser et al., 2023) (fifth row) produces high-quality videos, it exhibits a jitter effect on the foreground and lacks spatio-temporal consistency.

Furthermore, we present additional multi-view results showcasing a variety of styles and prompts in Fig. 5. Our denoising algorithm, driven by video diffusion models, effectively captures temporal

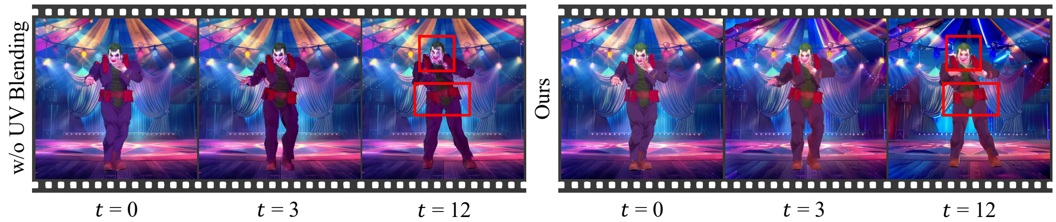


Figure 6. Ablation study on the reference UV blending module. Without this module, the generated textures may lose consistency over time as highlighted in the red boxes.

variations over time. For instance, as highlighted in the green boxes in Fig. 5, our method accurately represents cloth wrinkles (Row 1) and changes in lighting (Rows 2 and 3).

5.2 Quantitative Evaluation

To quantitatively assess the effectiveness of our proposed method, we follow prior research (Eldesokey & Wonka, 2024; Geyer et al., 2023; Esser et al., 2023) and conduct a comprehensive A/B user study. Our study involved 67 participants who provided a total of 1104 valid responses based on six different scenes drawn from six previous works, with each scene producing videos from two different views. During each evaluation, participants were presented with rendered meshes and depth conditions viewed from two angles, serving as motion references. They were shown a pair of videos: one generated by our approach and the other from a baseline method. Participants were asked to select the method that exhibited superior performance in three criteria: 1) appearance quality, 2) spatial and temporal consistency, and 3) fidelity to the prompts. Table 1 summarizes the preference percentage of our method over the baseline methods. Our method significantly surpasses state-of-the-art methods by a large margin. In addition, our method achieves lower FVD that demonstrates better multi-view consistency in generated video clips.

5.3 Ablation Study

Ablation for texture aggregation. In Fig. 3 (a) and (b), we present two alternative texture aggregation methods. In Fig. 3 (a), we un-project $\hat{z}_0(z_t)$ and $\epsilon_\theta(z_t)$ into UV space for aggregation. In Fig. 3 (b), we map z_{t-1} to the UV space. Both these two approaches show inferior results compared to our method, which verifies the effectiveness of the proposed texture aggregation algorithm.

Ablation for UV blending module. In Sec. 4.4, we propose a reference UV blending schema to resolve the temporal inconsistency caused by latent aggregation. To validate the effectiveness of this mechanism (See Sec. 4.4), we conduct an ablation study by disabling the reference UV blending module (setting λ to 0). As shown in Fig. 6, without the UV blending module, our method generates textures with noticeable distortions on the Joker's face over time.

Ablation for background priors. Sec. 4.3 discusses the importance of including a plausible background and proposes to learn a dynamic background through diffusion. To verify the effectiveness of this design, we replace the learnable background latents with an all-white background while keeping all other parts unchanged. Fig. 3 (c) illustrates that this ablation experiment produces significantly blurrier textures compared to our full method, highlighting the importance of background learning.

6 Conclusions

In this paper, we present a zero-shot approach that generates multi-view, multi-frame consistent textures for untextured, animated mesh sequences based on a text prompt. By incorporating texture aggregation in the UV space within the diffusion process of a conditional video diffusion model, we ensure both temporal and spatial coherence in the generated textures. To address the variance shift introduced by texture aggregation, we propose a simple yet effective modification to the DDIM sampling algorithm. Additionally, we enhance temporal consistency by introducing a reference UV map and develop a dynamic background learning framework to produce fully textured 4D scenes. Extensive experiments show that our method can synthesize realistic and consistent 4D textures, demonstrating its superiority against state-of-the-art baselines.

References

Max Bain, Arsha Nagrani, Gül Varol, and Andrew Zisserman. Frozen in time: A joint video and image encoder for end-to-end retrieval. In *ICCV*, 2021. 2

Omer Bar-Tal, Dolev Ofri-Amar, Rafail Fridman, Yoni Kasten, and Tali Dekel. Text2live: Text-driven layered image and video editing. In *ECCV*, pp. 707–723, 2022. 3

Raphael Bensadoun, Yanir Kleiman, Idan Azuri, Omri Harosh, Andrea Vedaldi, Natalia Neverova, and Oran Gafni. Meta 3d texturegen: Fast and consistent texture generation for 3d objects. *arXiv preprint arXiv:2407.02430*, 2024. 3, 4

Andreas Blattmann, Robin Rombach, Huan Ling, Tim Dockhorn, Seung Wook Kim, Sanja Fidler, and Karsten Kreis. Align your latents: High-resolution video synthesis with latent diffusion models. In *CVPR*, 2023. 3

Shengqu Cai, Duygu Ceylan, Matheus Gadelha, Chun-Hao Huang, Tuanfeng Wang, and Gordon. Wetzstein. Generative rendering: Controllable 4d-guided video generation with 2d diffusion models. In *CVPR*, 2024. 3, 7, 8, 9

Tianshi Cao, Karsten Kreis, Sanja Fidler, Nicholas Sharp, and KangXue Yin. Texfusion: Synthesizing 3d textures with text-guided image diffusion models. In *ICCV*, 2023. 2, 3, 4, 15

Duygu Ceylan, Chun-Hao Huang, and Niloy J. Mitra. Pix2video: Video editing using image diffusion. In *ICCV*, 2023. 3

Dave Zhenyu Chen, Haoxuan Li, Hsin-Ying Lee, Sergey Tulyakov, and Matthias Nießner. Scenetex: High-quality texture synthesis for indoor scenes via diffusion priors, 2023a. 3

Dave Zhenyu Chen, Yawar Siddiqui, Hsin-Ying Lee, Sergey Tulyakov, and Matthias Nießner. Text2tex: Text-driven texture synthesis via diffusion models. *arXiv preprint arXiv:2303.11396*, 2023b. 2, 3, 4, 17

Haoxin Chen, Menghan Xia, Yingqing He, Yong Zhang, Xiaodong Cun, Shaoshu Yang, Jinbo Xing, Yaofang Liu, Qifeng Chen, Xintao Wang, Chao Weng, and Ying Shan. Videocrafter1: Open diffusion models for high-quality video generation, 2023c. 2

Haoxin Chen, Yong Zhang, Xiaodong Cun, Menghan Xia, Xintao Wang, Chao Weng, and Ying Shan. Videocrafter2: Overcoming data limitations for high-quality video diffusion models, 2024. 2

Yongwei Chen, Rui Chen, Jiabao Lei, Yabin Zhang, and Kui Jia. Tango: Text-driven photorealistic and robust 3d stylization via lighting decomposition. *NeurIPS*, 35:30923–30936, 2022. 3

Blender Online Community. *Blender - a 3D modelling and rendering package*. Blender Foundation, Stichting Blender Foundation, Amsterdam, 2024. URL `http://www.blender.org`. 7

Abdelrahman Eldesokey and Peter Wonka. Latentman: Generating consistent animated characters using image diffusion models. In *CVPR*, pp. 7510–7519, 2024. 3, 7, 8, 9, 10

Patrick Esser, Johnathan Chiu, Parmida Atighehchian, Jonathan Granskog, and Anastasis Germanidis. Structure and content-guided video synthesis with diffusion models. In *ICCV*, pp. 7346–7356, 2023. 2, 8, 9, 10

Michal Geyer, Omer Bar-Tal, Shai Bagon, and Tali Dekel. Tokenflow: Consistent diffusion features for consistent video editing. *arXiv preprint arXiv:2307.10373*, 2023. 3, 8, 9, 10

Yuwei Guo, Ceyuan Yang, Anyi Rao, Maneesh Agrawala, Dahua Lin, and Bo Dai. Sparsectrl: Adding sparse controls to text-to-video diffusion models. *arXiv preprint arXiv:2311.16933*, 2023a. 2, 3

Yuwei Guo, Ceyuan Yang, Anyi Rao, Zhengyang Liang, Yaohui Wang, Yu Qiao, Maneesh Agrawala, Dahua Lin, and Bo Dai. Animatediff: Animate your personalized text-to-image diffusion models without specific tuning, 2023b. 3

Yingqing He, Tianyu Yang, Yong Zhang, Ying Shan, and Qifeng Chen. Latent video diffusion models for high-fidelity long video generation. *arXiv preprint arXiv:2211.13221*, 2022. 2

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *NeurIPS*, 33: 6840–6851, 2020. 4

Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey Gritsenko, Diederik P Kingma, Ben Poole, Mohammad Norouzi, David J Fleet, et al. Imagen video: High definition video generation with diffusion models. *arXiv preprint arXiv:2210.02303*, 2022. 2, 4

Wenyi Hong, Ming Ding, Wendi Zheng, Xinghan Liu, and Jie Tang. Cogvideo: Large-scale pre-training for text-to-video generation via transformers. *arXiv preprint arXiv:2205.15868*, 2022. 2

Dong Huo, Zixin Guo, Xinxin Zuo, Zhihao Shi, Juwei Lu, Peng Dai, Songcen Xu, Li Cheng, and Yee-Hong Yang. Texgen: Text-guided 3d texture generation with multi-view sampling and re-sampling. *ECCV*, 2024. 2, 3, 4, 5, 15

Levon Khachatryan, Andranik Movsisyan, Vahram Tadevosyan, Roberto Henschel, Zhangyang Wang, Shant Navasardyan, and Humphrey Shi. Text2video-zero: Text-to-image diffusion models are zero-shot video generators. *arXiv preprint arXiv:2303.13439*, 2023. 3, 8, 9

Nasir Mohammad Khalid, Tianhao Xie, Eugene Belilovsky, and Popa Tiberiu. Clip-mesh: Generating textured meshes from text using pretrained image-text models. *SIGGRAPH Aisa*, December 2022. 3

Jeong-gi Kwak, Erqun Dong, Yuhe Jin, Hanseok Ko, Shweta Mahajan, and Kwang Moo Yi. Vivid-1-to-3: Novel view synthesis with video diffusion models. *arXiv preprint arXiv:2312.01305*, 2023. 2

Bing Li, Cheng Zheng, Wenxuan Zhu, Jinjie Mai, Biao Zhang, Peter Wonka, and Bernard Ghanem. Vivid-zoo: Multi-view video generation with diffusion model, 2024. 9

Xin Li, Wenqing Chu, Ye Wu, Weihang Yuan, Fanglong Liu, Qi Zhang, Fu Li, Haocheng Feng, Errui Ding, and Jingdong Wang. Videogen: A reference-guided latent diffusion approach for high definition text-to-video generation. *arXiv preprint arXiv:2309.00398*, 2023. 2

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

Han Lin, Jaemin Cho, Abhay Zala, and Mohit Bansal. Ctrl-adapter: An efficient and versatile framework for adapting diverse controls to any diffusion model, 2024. 2, 3, 4, 5, 15

Yuan Liu, Cheng Lin, Zijiao Zeng, Xiaoxiao Long, Lingjie Liu, Taku Komura, and Wenping Wang. Syncdreamer: Generating multiview-consistent images from a single-view image. *arXiv preprint arXiv:2309.03453*, 2023a. 2, 16

Yuxin Liu, Minshan Xie, Hanyuan Liu, and Tien-Tsin Wong. Text-guided texturing by synchronized multi-view diffusion. *arXiv preprint arXiv:2311.12891*, 2023b. 2, 3, 4, 5, 15, 16

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. *arXiv preprint arXiv:2310.15008*, 2023. 2

Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. SMPL: A skinned multi-person linear model. *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)*, 34(6):248:1–248:16, October 2015. 7

Gal Metzer, Elad Richardson, Or Patashnik, Raja Giryes, and Daniel Cohen-Or. Latent-nerf for shape-guided generation of 3d shapes and textures. *arXiv preprint arXiv:2211.07600*, 2022. 3

Oscar Michel, Roi Bar-On, Richard Liu, Sagie Benaim, and Rana Hanocka. Text2mesh: Text-driven neural stylization for meshes. *arXiv preprint arXiv:2112.03221*, 2021. 3

Bohao Peng, Jian Wang, Yuechen Zhang, Wenbo Li, Ming-Chang Yang, and Jiaya Jia. Controlnext: Powerful and efficient control for image and video generation. *arXiv preprint arXiv:2408.06070*, 2024. 2

Ryan Po and Gordon Wetzstein. Compositional 3d scene generation using locally conditioned diffusion. In *2024 International Conference on 3D Vision (3DV)*, pp. 651–663. IEEE, 2024. 3

Ben Poole, Ajay Jain, Jonathan T. Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv*, 2022. 3

Chenyang Qi, Xiaodong Cun, Yong Zhang, Chenyang Lei, Xintao Wang, Ying Shan, and Qifeng Chen. Fatezero: Fusing attentions for zero-shot text-based video editing. *arXiv:2303.09535*, 2023. 3

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*, pp. 8748–8763, 2021. 3

Elad Richardson, Gal Metzer, Yuval Alaluf, Raja Giryes, and Daniel Cohen-Or. Texture: Text-guided texturing of 3d shapes. In *SIGGRAPH*, pp. 1–11, 2023. 2, 3, 4

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2021. 3

Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. *arXiv preprint arXiv:2202.00512*, 2022. 4

Christoph Schuhmann, Richard Vencu, Romain Beaumont, Robert Kaczmarczyk, Clayton Mullis, Aarush Katta, Theo Coombes, Jenia Jitsev, and Aran Komatsuzaki. Laion-400m: Open dataset of clip-filtered 400 million image-text pairs. *arXiv preprint arXiv:2111.02114*, 2021. 2

Ruoxi Shi, Hansheng Chen, Zhuoyang Zhang, Minghua Liu, Chao Xu, Xinyue Wei, Linghao Chen, Chong Zeng, and Hao Su. Zero123++: a single image to consistent multi-view diffusion base model. *arXiv preprint arXiv:2310.15110*, 2023a. 2

Yichun Shi, Peng Wang, Jianglong Ye, Long Mai, Kejie Li, and Xiao Yang. Mvdream: Multi-view diffusion for 3d generation. *arXiv:2308.16512*, 2023b. 2

Uriel Singer, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, et al. Make-a-video: Text-to-video generation without text-video data. *arXiv preprint arXiv:2209.14792*, 2022. 3

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020. 2, 3, 4

Shitao Tang, Fuyang Zhang, Jiacheng Chen, Peng Wang, and Yasutaka Furukawa. Mvdiffusion: Enabling holistic multi-view image generation with correspondence-aware diffusion. *arXiv*, 2023. 2

Shitao Tang, Jiacheng Chen, Dilin Wang, Chengzhou Tang, Fuyang Zhang, Yuchen Fan, Vikas Chandra, Yasutaka Furukawa, and Rakesh Ranjan. Mvdiffusion++: A dense high-resolution multi-view diffusion model for single or sparse-view 3d object reconstruction. *arXiv preprint arXiv:2402.12712*, 2024. 2

Guy Tevet, Sigal Raab, Brian Gordon, Yoni Shafir, Daniel Cohen-or, and Amit Haim Bermano. Human motion diffusion model. In *The Eleventh International Conference on Learning Representations*, 2023. URL `https://openreview.net/forum?id=SJ1kSyO2jwu`. 7

Narek Tumanyan, Michal Geyer, Shai Bagon, and Tali Dekel. Plug-and-play diffusion features for text-driven image-to-image translation. In *CVPR*, pp. 1921–1930, June 2023. 3, 8, 9

Thomas Unterthiner, Sjoerd Van Steenkiste, Karol Kurach, Raphael Marinier, Marcin Michalski, and Sylvain Gelly. Towards accurate generative models of video: A new metric & challenges. *arXiv preprint arXiv:1812.01717*, 2018. 9

Vikram Voleti, Chun-Han Yao, Mark Boss, Adam Letts, David Pankratz, Dmitrii Tochilkin, Christian Laforte, Robin Rombach, and Varun Jampani. SV3D: Novel multi-view synthesis and 3D generation from a single image using latent video diffusion. In *ECCV*, 2024. 2

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. 3

Haohan Weng, Tianyu Yang, Jianan Wang, Yu Li, Tong Zhang, CL Chen, and Lei Zhang. Consistent123: Improve consistency for one image to 3d object synthesis. *arXiv preprint arXiv:2310.08092*, 2023. 2

Jay Zhangjie Wu, Yixiao Ge, Xintao Wang, Stan Weixian Lei, Yuchao Gu, Yufei Shi, Wynne Hsu, Ying Shan, Xiaohu Qie, and Mike Zheng Shou. Tune-a-video: One-shot tuning of image diffusion models for text-to-video generation. In *ICCV*, pp. 7623–7633, 2023. 3

Yiming Xie, Chun-Han Yao, Vikram Voleti, Huaizu Jiang, and Varun Jampani. SV4D: Dynamic 3d content generation with multi-frame and multi-view consistency. *arXiv preprint arXiv:2407.17470*, 2024. 9

Jinbo Xing, Menghan Xia, Yong Zhang, Haoxin Chen, Xintao Wang, Tien-Tsin Wong, and Ying Shan. Dynamicrafter: Animating open-domain images with video diffusion priors. *arXiv preprint arXiv:2310.12190*, 2023. 2

Zhongcong Xu, Jianfeng Zhang, Jun Hao Liew, Hanshu Yan, Jia-Wei Liu, Chenxu Zhang, Jiashi Feng, and Mike Zheng Shou. Magicanimate: Temporally consistent human image animation using diffusion model. In *CVPR*, 2024. 3

Shuai Yang, Yifan Zhou, Ziwei Liu, , and Chen Change Loy. Rerender a video: Zero-shot text-guided video-to-video translation. In *SIGGRAPH Aisa*, 2023. 3

Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, et al. Cogvideox: Text-to-video diffusion models with an expert transformer. *arXiv preprint arXiv:2408.06072*, 2024. 2

Sihyun Yu, Kihyuk Sohn, Subin Kim, and Jinwoo Shin. Video probabilistic diffusion models in projected latent space. In *CVPR*, pp. 18456–18466, 2023a. 2

Xin Yu, Peng Dai, Wenbo Li, Lan Ma, Zhengzhe Liu, and Xiaojuan Qi. Texture generation on 3d meshes with point-uv diffusion. In *ICCV*, pp. 4206–4216, 2023b. 3

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 *CVPR*, pp. 4252–4262, 2024. 3

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)*, 2024. 4, 5

Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *CVPR*, pp. 3836–3847, 2023a. 3, 4

Shiwei Zhang, Jiayu Wang, Yingya Zhang, Kang Zhao, Hangjie Yuan, Zhiwu Qing, Xiang Wang, Deli Zhao, and Jingren Zhou. I2vgen-xl: High-quality image-to-video synthesis via cascaded diffusion models. *arXiv preprint arXiv:2311.04145*, 2023b. 2, 4, 15

Yabo Zhang, Yuxiang Wei, Dongsheng Jiang, Xiaopeng Zhang, Wangmeng Zuo, and Qi Tian. Controlvideo: Training-free controllable text-to-video generation. *arXiv preprint arXiv:2305.13077*, 2023c. 3

Daquan Zhou, Weimin Wang, Hanshu Yan, Weiwei Lv, Yizhe Zhu, and Jiashi Feng. Magicvideo: Efficient video generation with latent diffusion models. *arXiv preprint arXiv:2211.11018*, 2022. 2

A More Implementation Details

A.1 Implementation Details

We utilize the CTRL-Adapter (Lin et al., 2024), trained on the video diffusion model I2VGen-XL (Zhang et al., 2023b), as the backbone for generation, with the denoising steps set to $T = 50$. Initially, we center the untextured mesh sequence and pre-define six different viewpoints around the Y-axis in the XZ-plane, uniformly sampled in spherical coordinates, along with an additional top view with an elevation angle of zero and an azimuth angle of $30°$. For latent initialization, we first sample Gaussian noise on the latent textures and then render 2D latent samples for each view to improve the coherence of the generated outputs. During denoising, we upscale the latent resolution to 96×96 to reduce aliasing. We empirically set the blending coefficient to 0.2. It takes approximately 30 minutes to generate a video with 24 key frames taken on a RTX A6000 Ada GPU. We decode the denoised latents in key frames to RGB images, and then un-project and aggregate these images to transform the latent UV maps to RGB textures as previous works (Liu et al., 2023b; Cao et al., 2023; Huo et al., 2024). Finally, we interpolate the textures of the key frames at intervals of 3 to synthesize the final video clips.

A.2 Denoising Algorithm of Our Method

We present the complete workflow of our method in Algorithm 1. The reference UV map $\mathcal{T}_{\mathcal{UV}}$ is constructed by progressively combining latent textures over time, with each new texture filling only the unoccupied texels in the reference UV map.

Algorithm 1 Tex4D

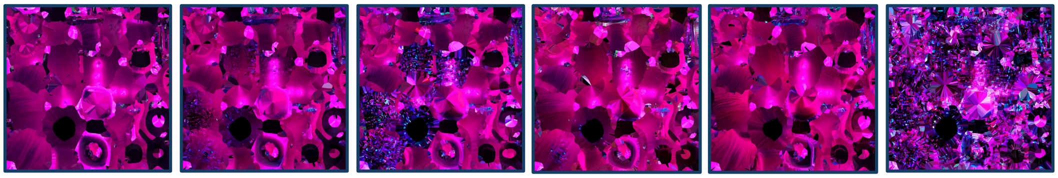
Input: UV maps $\mathcal{UV} = \{UV_1, ..., UV_K\}$; depth maps $\boldsymbol{\mathcal{D}} = \{D_{1,1}, ..., D_{1,V}, D_{2,1}, ..., D_{K,V}\}$; text prompt $\boldsymbol{\mathcal{P}}$; CTRL-Adapter model $\boldsymbol{\mathcal{C}}$; rendering operation $\mathcal{R}$; unproject operation $\mathcal{R}^{-1}$; cameras $\boldsymbol{c}$; T diffusion steps; $\mathcal{T}$ latent textures (including foreground and background); λ blending weight; k key frames

$\mathcal{T}_T \sim \mathcal{N}(\mathbf{0}, \mathcal{I})$ // Sample noise in UV space
$\tilde{\boldsymbol{z}}_T, \boldsymbol{\mathcal{M}}_{\text{fg}} = \mathcal{R}(\mathcal{T}_T; \boldsymbol{c})$
$\boldsymbol{z}_{b,T} \sim \mathcal{N}(\mathbf{0}, \mathcal{I})$
$\boldsymbol{z} = \boldsymbol{z}_T = \tilde{\boldsymbol{z}}_T \odot \boldsymbol{\mathcal{M}}_{\text{fg}} + \boldsymbol{z}_{b,T} \odot (1 - \boldsymbol{\mathcal{M}}_{\text{fg}})$ // Composite latents
For $t = T, \dots, 1$ **do**
  $\boldsymbol{z}_{b,t-1} \leftarrow \boldsymbol{\mathcal{C}}(\boldsymbol{z}_{b,t}; \boldsymbol{\mathcal{D}}, \boldsymbol{\mathcal{P}})$
  $\boldsymbol{\epsilon}_\theta \leftarrow \boldsymbol{\mathcal{C}}(\boldsymbol{z}_t; \boldsymbol{\mathcal{D}}, \boldsymbol{\mathcal{P}})$ // Estimate noise from $\boldsymbol{\mathcal{C}}$
  $\hat{\boldsymbol{z}}_0(\boldsymbol{z}_t) = \sqrt{\alpha_t} \cdot \boldsymbol{z}_t - \sqrt{1 - \alpha_t} \cdot \boldsymbol{\epsilon}_\theta$
  $\hat{\mathcal{T}}_0, \boldsymbol{\mathcal{M}}_{\mathcal{UV}} \leftarrow \mathcal{R}^{-1}(\hat{\boldsymbol{z}}_0; \boldsymbol{c}, \mathcal{UV})$ // Bake textures by Eq. 4
  $\mathcal{T}_{\mathcal{UV}} = \text{Combine}(\hat{\mathcal{T}}_0; \boldsymbol{\mathcal{M}}_{\mathcal{UV}})$
  For k in $1, ..., K$ **do**
    $\mathcal{T}^k_{t-1} = \sqrt{\alpha_{t-1}} \cdot \hat{\mathcal{T}}^k_0 + \sqrt{1 - \alpha_{t-1}} \left(\sqrt{\frac{\alpha_t}{1-\alpha_t}} \cdot (\sqrt{\alpha_t} \mathcal{T}^k_t - \hat{\mathcal{T}}^k_0) + \sqrt{1 - \alpha_t} \cdot \mathcal{T}^k_t \right)$ // Denoise Eq. 6
    $\mathcal{T}^k_{t-1} = \left((1 - \lambda) \cdot \mathcal{T}^k_{t-1} + \lambda \cdot \mathcal{T}_{\mathcal{UV}}\right) \odot \boldsymbol{\mathcal{M}}^k_{\mathcal{UV}} + \mathcal{T}_{\mathcal{UV}} \odot \left(1 - \boldsymbol{\mathcal{M}}^k_{\mathcal{UV}}\right)$ // Blend textures by Eq. 8
  $\tilde{\boldsymbol{z}}_{t-1}, \boldsymbol{\mathcal{M}}_{\text{fg}} = \mathcal{R}\left(\mathcal{T}_{t-1}; \boldsymbol{c}, \mathcal{UV}\right)$
  $\boldsymbol{z}_{t-1} = \tilde{\boldsymbol{z}}_{t-1} \odot \boldsymbol{\mathcal{M}}_{\text{fg}} + \boldsymbol{z}_{b,t-1} \odot (1 - \boldsymbol{\mathcal{M}}_{\text{fg}})$ // Composite latents by Eq. 7
  $\boldsymbol{z} = \boldsymbol{z}_{t-1}$
Output: $\boldsymbol{z}$

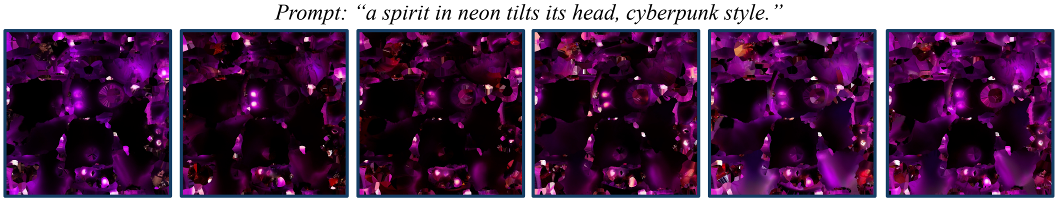
B More Qualitative Results

B.1 Multi-view Results

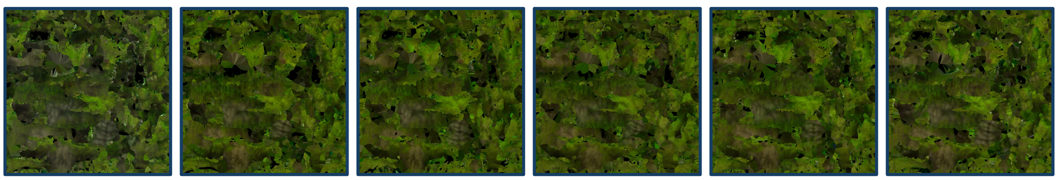
In Fig. 12, we present additional characters generated by Tex4D, showcasing the method's effectiveness and its ability to generalize across a diverse array of styles and prompts. We also evaluate Tex4D on non-human character animations in Fig. 13, demonstrating its robust generalization capa-



Prompt: "a spirit in neon tilts its head, cyberpunk style."



Prompt: "a ghost flashed a magical light, causing dramatic shifts in lighting."



Prompt: "a mossy stone monster dances in a mysterious forest."

Figure 7. **Visualization of generated textures for mesh sequences.** Our method effectively incorporates temporal changes, such as lighting variations, wrinkles, and appearance transformations, directly into the textures, eliminating the need for post-production by artists.

bilities across various types of mesh sequences. In each case, we provide two different view to show that our method can ensure the multi-view consistency.

To emphasize the temporal changes in the generated textures, we also design some prompts, for example, 'flashed a magical light', 'dramatic shifts in lighting', 'cyberpunk style' in our experiments. As shown in Fig. 13, the results of 'ghost', 'King Boo' and 'Snowman' validate the effectiveness of our method in generating different level of temporal changes by a variety of textual prompts, while maintaining the consistency both spatially and temporally. Additionally, we provide a supplementary video that includes baseline comparisons and multi-view results for all examples.

B.2 Texture Results

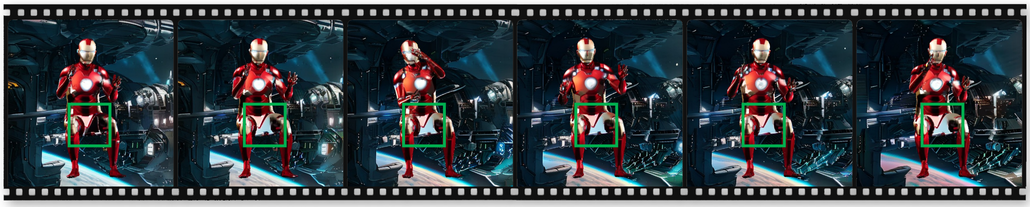
In this section, we present the texture sequences which are the intermediate results of our pipeline. Our method utilizes XATLAS to unwrap the UV maps from meshes without human labors. XATLAS is a widely used library for mesh parameterization and UV unwrapping, commonly integrated into popular tools and engines, facilitating efficient texture mapping in 3D graphics applications. As shown in Fig. 7, our method seamlessly bakes temporal changes, including lighting variations, wrinkles, and appearance transformations, directly into the textures, removing the need for manual post-production by artists.

C More Ablation Results

Ablation on Background To show the effects of various background latent initialization strategies, we provide additional examples, including the approach used in the texture synthesis method (Liu et al., 2023b) and a background that contrasts sharply with the foreground object, as shown in Fig. 8. In detail, (Liu et al., 2023a) encodes the backgrounds with alternative random solid color images. For the high contrast background experiment, we use the latents obtained from the DDIM inversion of highly contrast foreground and background to initialize our latents.

Ablation on Reference UV Blending We present an additional ablation study to illustrate how our UV blending module enhances temporal consistency across frames. As shown in Fig. 9, the absence of UV blending results in noticeable distortions, underscoring the importance of this module in maintaining visual coherence.

(a) Background w/ alternative random noise (b) Background highly contrast w/ foreground



(c) Ours w/o background priors (foreground composited with our background)

Figure 8. **More ablation study on the background priors.**

Ours w/o UV Blending Ours

Figure 9. **More ablation study on the reference UV blending module.**

D More Method Comparisons

D.1 Comparison with Depth-Conditioned Video Diffusion Models

While depth-conditioned video diffusion models are effective at generating visually compelling results from a single viewpoint, they often struggle to maintain consistent multi-view representations of a single object, such as a character, across different perspectives. To highlight this limitation, we present multi-view results from the depth-conditioned video diffusion model in Fig. 11. The primary cause of this issue is that depth conditions are inherently view-dependent, in contrast to UV maps, which provide global information about the 3D space, enabling a unique mapping for each 3D point across all views.

D.2 Comparison with Textured Mesh Animations

In this section, we highlight the differences between our method and traditional approaches, demonstrating the effectiveness of 4D texturing in capturing temporal variations (e.g., lighting and wrinkles) within mesh sequences to produce vivid visual results. Traditional methods typically involve texturing a base mesh (often referred to as a clay mesh) and animating it using skinning techniques. This animated sequence is then refined by technical artists who control scene lighting or simulate cloth dynamics to achieve the final visual presentation. This process is labor-intensive and demands specialized expertise in cinematic production and technical engines.

In contrast, our method offers a streamlined alternative by directly integrating complex temporal changes into mesh sequences. As shown in Fig. 5, 12 and 13, our approach effectively captures intricate temporal effects such as cloth wrinkles, dynamic lighting, and evolving appearances using textual prompts, significantly simplifying the workflow while maintaining high-quality results.

We demonstrate the limitations of traditional textured mesh animation in handling complex temporal changes in Fig. 10. Specifically, we employ the Text2Tex (Chen et al., 2023b) to generate the texture for the input mesh in T-pose and render it from multiple viewpoints. To ensure a fair comparison, we composite the rendered results with the background generated by our method. Notably, the 'ghost'

Prompt: "a ghost flashed a magical light, causing dramatic shifts in lighting."

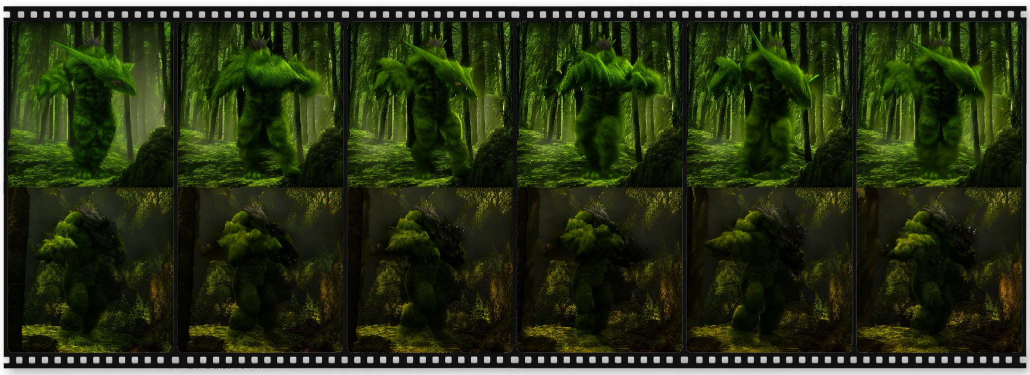
Prompt: "a spirit in neon tilts its head, cyberpunk style."

Figure 10. **Results of textured mesh animation (Text2Tex).**

and 'snowman' examples exhibit visible seams during animation due to self-occlusions are common appeared in dynamic poses but cannot be accurately predicted during T-pose texture generation. This results in empty texels and disrupts the visual continuity of the animation.

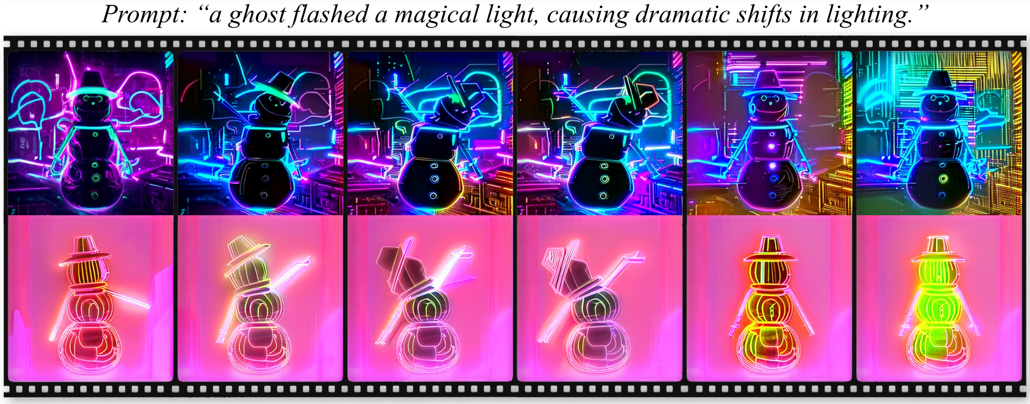
E LIMITATIONS AND DISCUSSION

One limitation of our method is the lack of seamless integration between the generated textures and the background, resulting in a disjointed appearance where the foreground and background elements may seem artificially stitched together. This issue arises due to the absence of a comprehensive scene-level dataset. Alternatively, our approach relies on a shared background mesh across different views, which disrupts overall consistency. Addressing scene-level 4D texturing remains an open challenge that we aim to explore in future work. In addition, we notice that our method is relatively computational intensive compared with other texture synthesis methods. The computation time of our method primarily depends on the foundation model (CTRL-Adapter), which takes approximately 5 minutes to generate a 24-frame video. We anticipate significant efficiency improvements with advancements in conditioned video diffusion models, further enhancing the practicality of our approach.



Prompt: "a mossy stone monster dances in a mysterious forest."

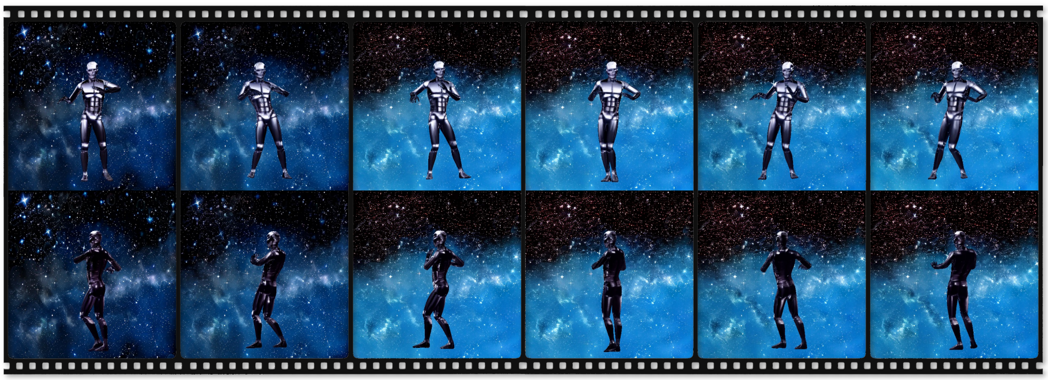
Prompt: "a ghost flashed a magical light, causing dramatic shifts in lighting."



Prompt: "a spirit in neon tilts its head, cyberpunk style."

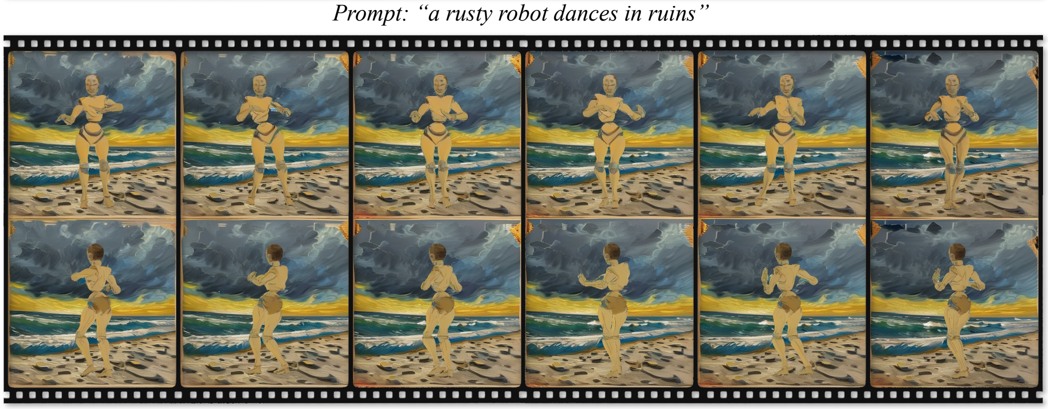
Figure 11. **Multi-view results from video diffusion model.**

Prompt: "the Joker dances, comic style"



Prompt: "the terminator dancing in the milky way"

Prompt: "a rusty robot dances in ruins"



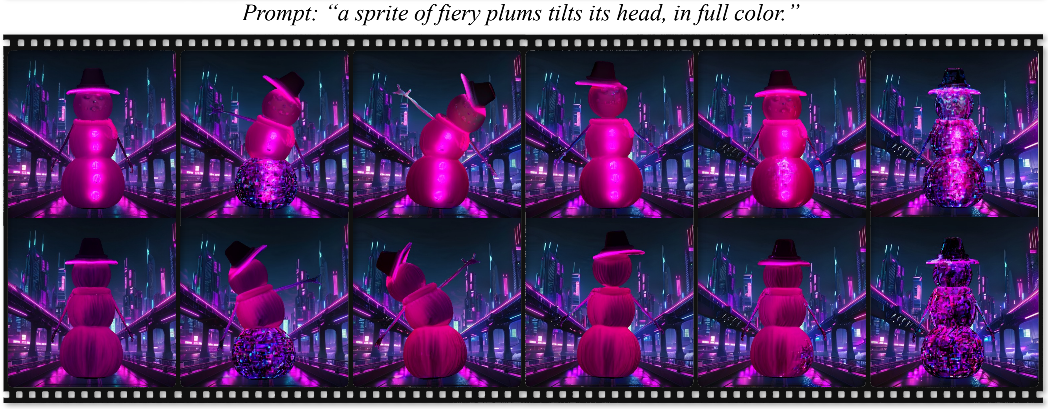
Prompt: "a sketch of bot dancing in a sandy beach, Van-Gogh style."

Figure 12. **More qualitative results.**

Prompt: "a ghost flashed a magical light, causing dramatic shifts in lighting."

Prompt: "a dingy, magic King Boo, flashing a weird light, static background."

Prompt: "a sprite of fiery plums tilts its head, in full color."



Prompt: "a spirit in neon tilts its head, cyberpunk style."

Figure 13. **More qualitative results on non-human character animations.**