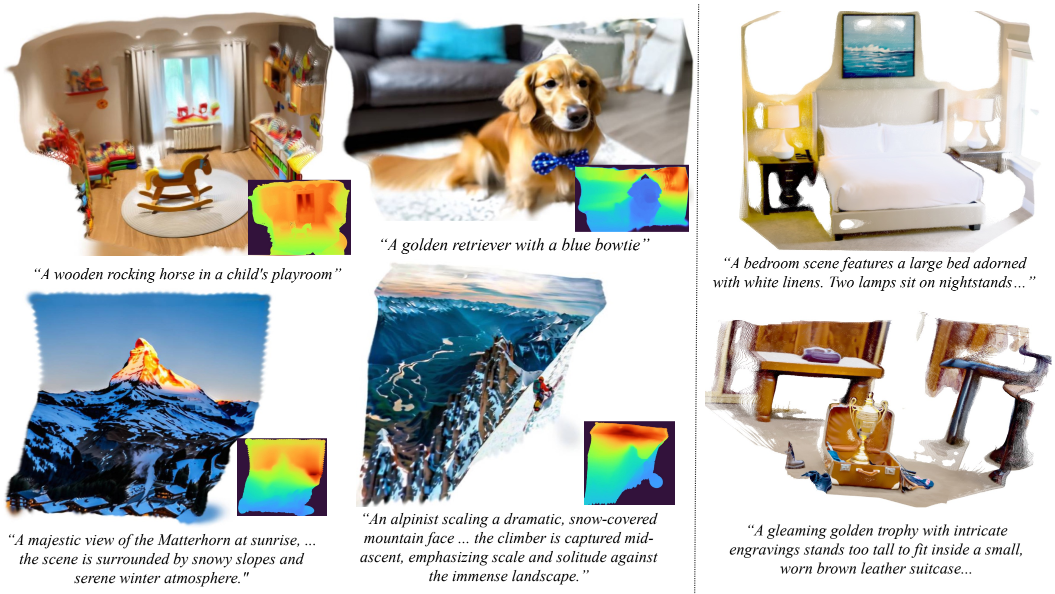
Text-to-3D by Stitching a Multi-view Reconstruction Network to a Video Generator

Anonymous authors
Paper under double-blind review


"A wooden rocking horse in a child's playroom"
"A golden retriever with a blue bowtie"
"A bedroom scene features a large bed adorned with white linens. Two lamps sit on nightstands..."
"A majestic view of the Matterhorn at sunrise, ... the scene is surrounded by snowy slopes and serene winter atmosphere."
"An alpinist scaling a dramatic, snow-covered mountain face ... the climber is captured mid-ascent, emphasizing scale and solitude against the immense landscape."
"A gleaming golden trophy with intricate engravings stands too tall to fit inside a small, worn brown leather suitcase..."

(a) Text-to-3DGS **(b) Text-to-Pointmap**

Figure 1: **Text-to-3D generation with *VIST3A*.** Video models excel at generating latent visual content from text prompts, whereas 3D foundation models shine when it comes to decoding such a latent representation into consistent scene geometry. By stitching a video generator and a 3D reconstruction network together and aligning their latents, we obtain an end-to-end model that produces high-quality Gaussian splats (a) or point maps (b) from text prompts.

Abstract

The rapid progress of large, pretrained models for both visual content generation and 3D reconstruction opens up new possibilities for text-to-3D generation. Intuitively, one could obtain a formidable 3D scene generator if one were able to combine the power of a modern latent text-to-video model as "generator" with the geometric abilities of a recent (feedforward) 3D reconstruction system as "decoder". We introduce VIST3A, a general framework that does just that, addressing two main challenges. First, the two components must be joined in a way that preserves the rich knowledge encoded in their weights. We revisit *model stitching*, i.e., we identify the layer in the 3D decoder that best matches the latent representation produced by the text-to-video generator and stitch the two parts together. That operation requires only a small dataset and no labels. Second, the text-to-video generator must be aligned with the stitched 3D decoder, to ensure that the generated latents are decodable into consistent, perceptually convincing 3D scene geometry. To that end, we adapt *direct reward finetuning*, a popular technique for human preference alignment. We evaluate the proposed VIST3A approach with different video generators and 3D reconstruction models. All tested pairings markedly improve over prior text-to-3D models that output Gaussian splats. Moreover, by choosing a suitable 3D base model, VIST3A also enables high-quality text-to-pointmap generation.

Generated results can be found in: https://unknownresearch15.github.io/anon13/

1 Introduction

With image and video generators now a commodity, text-to-3D models that produce 3D scenes from text prompts have become a new research frontier, with applications in AR/VR, gaming, robotics, and simulation. Early methods for 3D generation adopt Score Distillation Sampling (SDS) (Poole et al., 2023; Tang et al., 2024b; Wang et al., 2023b; Chen et al., 2024b) to optimize a 3D representation, e.g. a NeRF (Mildenhall et al., 2021; Müller et al., 2022) or 3D Gaussian Splats (3DGS, Kerbl et al., 2023) under a pretrained 2D diffusion prior (Rombach et al., 2022). A drawback these methods have in common is the need for slow per-scene optimization. Another line of work uses multi-stage pipelines that first synthesize images and then lift them to 3D with a separate model (Tang et al., 2024a; Xu et al., 2024b; Zhang et al., 2024b) or with per-scene optimization (Gao et al., 2024; Wu et al., 2024a; Yu et al., 2024b); employ progressive warping and refinement (Shriram et al., 2025; Yu et al., 2025; 2024a); or sequentially chain multiple generative modules (Yang et al., 2025b; Engstler et al., 2025). The multi-stage design not only increases model complexity and engineering effort, but also makes such models prone to error accumulation (Lin et al., 2025; Meng et al., 2025).

A recent trend is to directly generate the 3D representation with end-to-end latent diffusion models (LDMs, Schwarz et al., 2025; Lan et al., 2024; Li et al., 2025b;a). A prominent line of work starts from pretrained 2D image (Esser et al., 2024; Rombach et al., 2022) or video models (Team, 2024; Yang et al., 2024b) and finetunes them to output multi-view 2D latents, reusing the pretrained priors (Szymanowicz et al., 2025; Liang et al., 2025; Schwarz et al., 2025; Lin et al., 2025; Yang et al., 2025c; Go et al., 2025a;b). Subsequently, a VAE-style decoder is trained to decode those latents into the desired 3D representation, see Fig. 2. The LDM-like design unifies 2D generation and multi-view reconstruction within the latent space and enables efficient 3D scene generation with a compact, well-amortized decoder.

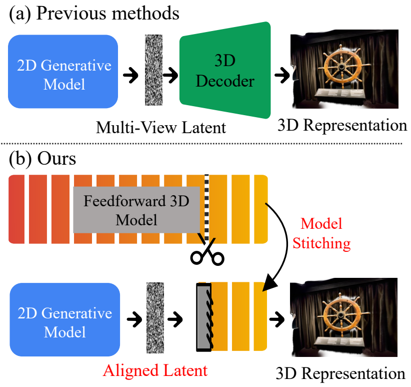


Figure 2: **Comparison with existing, LDM-based 3D generators.** Instead of training a custom decoder from multi-view 2D latents to 3D outputs, we stitch and align an existing, pretrained 3D reconstruction model.

Still, two key limitations remain. First, we argue that the Achilles heel of existing 2D-to-3D diffusion models is the decoder. By simply repurposing the 2D VAE to produce 3D outputs, the network must learn 3D reconstruction more or less from scratch, which requires extensive training and large datasets that are hard to obtain (Yang et al., 2025c; Szymanowicz et al., 2025; Go et al., 2025b). This practice becomes increasingly problematic as new, better 3D foundation models emerge (Wang et al., 2025d;a; 2024b; Zhang et al., 2025) and the ad-hoc trained decoders of text-to-3D models fall further behind the state of the art in 3D vision.

Second, the prevalent training scheme tends to suffer from weak alignment between the generative model and the VAE decoder. Typically, the former is finetuned on multi-view datasets with a generative objective like a diffusion loss (Song et al., 2020; Sohl-Dickstein et al., 2015; Ho et al., 2020) or flow matching (Liu et al., 2023; Lipman et al., 2023; Albergo & Vanden-Eijnden, 2023), which only indirectly promotes 3D-consistent latents. Moreover, the separate training may cause the latents, even if 3D-consistent, to be out of domain from the perspective of the decoder. To mitigate that misalignment, it has been proposed to add rendering losses that promote decodable latents (Lin et al., 2025). However, the resulting objective is based on single-step sampling and does not sufficiently take into account the denoising trajectory, leading to weak alignment at inference.

We introduce **VIST3A**: **VI**deo VAE **ST**itching and **3**D **A**lignment. The proposed method consists of two complementary components that address the above-mentioned limitations, see Fig. 2. First, we resort to the concept of *model stitching* (Pan et al., 2023; Lenc & Vedaldi, 2015; Bansal et al., 2021; Csiszárik et al., 2021; Yang et al., 2022) to leverage powerful, pretrained feedforward 3D models for decoding, rather than start from scratch. The idea is to attach the relevant part of a 3D reconstruction network as a “decoder” to the latent space of a video VAE. For this to work, there needs to be one or more layers in the 3D model whose activations are similar (up to a linear transformation) to those in the VAE’s latent space, despite their independent pretraining. Perhaps surprisingly, this turns out to be the case. For the 3D model, we identify the layer with the most linear relation to the LDM latents, slice the network before that layer, and retain the downstream portion as 3D decoder. After fitting a single, linear stitching layer (in closed form), the VAE latent space already matches the expected input of the 3D decoder well, such that subsequent fine-tuning will be minor and not degrade the respective generative and 3D reasoning capabilities of the two base models.

Second, we further improve alignment between the generative model and the stitched decoder through *direct reward finetuning* (Clark et al., 2023; Xu et al., 2023; Prabhudesai et al., 2024; Wu et al., 2024c; Shen et al., 2025). In that technique, commonly used to align diffusion models with human preferences, reward signals are defined based on the "goodness" of the VAE output – in our setting, the visual quality and 3D consistency of the decoded 3D representations. Maximizing these rewards encourages the LDM to produce latents that are 3D-consistent and lie within the decoder's input domain, ensuring high-quality outputs. Importantly, our alignment compares video model outputs and images rendered from the generated 3D scenes, hence it does not require labels.

In our experiments, we show that the proposed stitching scheme is applicable across a range of video generative models and also across several different feedforward 3D models. VIST3A's direct 3D decoding consistently outperforms prior text-to-3DGS methods, and additionally offers high-quality pointmap generation from text prompts.

2 Related Works

3D generation. Recent works have explored various 3D representations for generative modelling, including point clouds (Mo et al., 2023; Nichol et al., 2022; Vahdat et al., 2022), meshes (Xu et al., 2024a), voxel grids (Sanghi et al., 2023), NeRFs (Chen et al., 2023; Müller et al., 2022; Mildenhall et al., 2021), and 3DGS (Henderson et al., 2024; Zhang et al., 2024a; Kerbl et al., 2023). Score distillation using 2D diffusion models is time-consuming, as it requires per-scene test time optimization (Wang et al., 2023a; Shi et al., 2023; Wang et al., 2023b), while multi-stage pipelines (Yu et al., 2024b; Liu et al., 2024; Zheng et al., 2025) lack robustness and create significant engineering overhead. For further details on multi-stage pipelines, please refer to Appendix A.

More recently, the field has shifted towards end-to-end latent diffusion models, where the generator operates in the latent space of a VAE, and the latter directly decodes the resulting latents to 3D outputs. Many of these works focus on object-centric asset generation (Wu et al., 2024b; Zhao et al., 2023; Lin et al., 2025) and train the LDM on curated datasets such as Objaverse (Deitke et al., 2023), with single objects or bounded scenes, and controlled camera paths. Consequently, they are unable to handle real-world challenges like strongly varying scene scale, variable lighting, etc.

To tackle such situations, recent methods (Szymanowicz et al., 2025; Liang et al., 2025; Schwarz et al., 2025; Lin et al., 2025; Yang et al., 2025c; Go et al., 2025a;b) repurpose the comprehensive knowledge of the visual world that is implicit in 2D image generators. The general strategy is to finetune a pretrained 2D model on multi-view data, by using generative losses to enforce cross-view consistency. In many cases training is further supported by additional 3D cues like camera poses (Li et al., 2024; Go et al., 2025b), depthmaps (Go et al., 2025a; Yang et al., 2025c), or pointmaps (Szymanowicz et al., 2025). The resulting multi-view latents are decoded to 3D scenes with a dedicated VAE-style decoder, meaning that 3D reasoning capabilities must be rebuilt from scratch, and that they are only weakly aligned with the generator output – limitations which we address with VIST3A.

Learned 3D reconstruction. A notable trend in 3D computer vision is the trend to move away from multi-stage pipelines and iterative optimization towards end-to-end, feedforward 3D modelling. Classical reconstruction pipelines based on SfM (Hartley & Zisserman, 2003; Schönberger & Frahm, 2016) and MVS (Furukawa et al., 2015; Schönberger et al., 2016) require incremental, iterative optimization, whereas recent advances like DUSt3R (Wang et al., 2024b) and MASt3R (Leroy et al., 2024) directly predict 3D point maps in one forward pass. Several follow-up works have further reduced test-time optimization (Tang et al., 2025; Wang et al., 2025b; Yang et al., 2025a). Likewise, 3D Gaussian splatting has evolved from per-scene optimization to feedforward prediction (Charatan et al., 2024; Chen et al., 2024a; Ye et al., 2024). Once more, data scaling has been a critical factor (Wang et al., 2025a;d). Consequently, replicating the 3D capabilities of recent feedforward models as part of VAE training would be difficult and costly. VIST3A offers a solution by reusing, rather than rebuilding, models like AnySplat (Jiang et al., 2025), VGGT (Wang et al., 2025a), or MVDUSt3R (Tang et al., 2025).

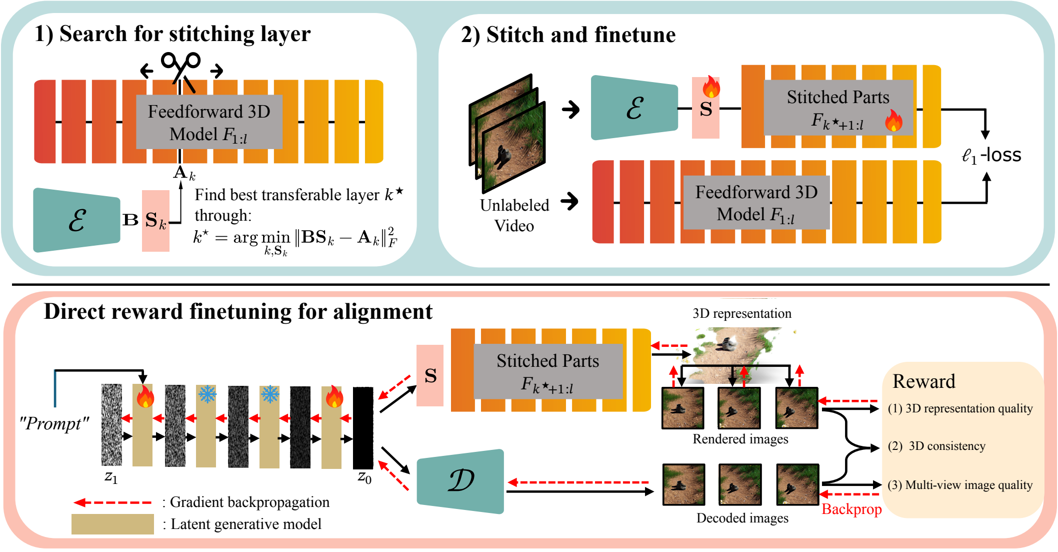

1) Search for stitching layer
Feedforward 3D Model $F_{1:l}$
$\mathbf{A}_k$
$\mathcal{E}$
$\mathbf{B}$ $\mathbf{S}_k$
Find best transferable layer $k^\star$ through:
$k^\star = \arg\min_{k,\mathbf{S}_k} \|\mathbf{B}\mathbf{S}_k - \mathbf{A}_k\|_F^2$
2) Stitch and finetune
Unlabeled Video
$\mathcal{E}$
$\mathbf{S}$
Stitched Parts $F_{k^\star+1:l}$
ℓ_1-loss
Feedforward 3D Model $F_{1:l}$
Direct reward finetuning for alignment
"Prompt"
z_1
z_0
$\mathbf{S}$
Stitched Parts $F_{k^\star+1:l}$
3D representation
Rendered images
$\mathcal{D}$
Decoded images
Backprop
Reward
(1) 3D representation quality
(2) 3D consistency
(3) Multi-view image quality
: Gradient backpropagation
: Latent generative model

Figure 3: **VIST3A constructs a 3D VAE through model stitching (top), then aligns it with a generative model via direct reward finetuning (bottom).** Stitching repurposes a part of a pretrained 3D vision model as decoder to obtain a 3D VAE. Direct reward finetuning simulates full-trajectory denoising, forcing the generative model to produce 3D-consistent, decodable latents.

Model stitching. Recomposing the heads and tails of two different networks was initially studied as a way to assess the equivariance of neural representations (Lenc & Vedaldi, 2015), and as an experimental tool to compare two different representations (Csiszárik et al., 2021; Bansal et al., 2021). To ensure invariance against trivial affine transformations, the head of some trained network A is normally attached to the tail of another network B via a linear, trainable *stitching layer*. Besides revealing similarities between networks that common metrics like CKA (Kornblith et al., 2019) would miss, it was also found that different architectures that were trained on the same data can often be stitched into a new, hybrid model with minimal degradation (Bansal et al., 2021). This has opened the door for practical uses of stitching, e.g. DeRy (Yang et al., 2022) for resource-constrained reassembly of pretrained models and SN-Net (Pan et al., 2023) to build networks with varying scales. Going one step further, we demonstrate that strong 3D VAEs[1] can be obtained by stitching a foundational 3D model to the latent space of a video VAE as its decoder, even if they were trained independently on different data.

3 Methodology

VIST3A consists of two key components, see Fig. 3: (1) model stitching to optimally attach (part of) a foundational 3D model as the decoder for the latent, and (2) direct reward finetuning to optimize the alignment of the (latent) generative model with that new decoder.

3.1 Model Stitching for 3D VAE Construction

Our objective is to build a 3D VAE by seamlessly combining the encoder of a video LDM and a feedforward 3D reconstruction model. Note that, for stitching purposes, one can skip the denoising loop, since feeding images into the encoder already gives clean latents. Let $\mathcal{E}$ denote the encoder and $\mathcal{D}$ the decoder of the VAE, and let $F_{1:l}(\boldsymbol{x}) = f_l \circ \cdots \circ f_1(\boldsymbol{x}) = \boldsymbol{y}$ be the feedforward 3D network that maps a set of views $\boldsymbol{x}$ to a 3D output $\boldsymbol{y}$, with l the total number of layers in that feedforward model. As shown in Fig. 3, we cut the feedforward model at layer k^* and stitch the downstream part $F_{k^\star+1:l} = f_l \circ \cdots \circ f_{k^\star+1}$ to the output layer of the encoder $\mathcal{E}$, with the help of a linear *stitching layer* $\mathbf{S}$. In doing so, we obtain a new 3D VAE $\mathcal{M}_{\text{stitched}}$ that outputs the same representation $\hat{\boldsymbol{y}}$ as the original 3D model:

$$\mathcal{M}_{\text{stitched}} = F_{k^\star+1:l} \circ \mathbf{S} \circ \mathcal{E}(\boldsymbol{x}) = \hat{\boldsymbol{y}}, \quad \mathcal{D}_{\text{stitched}} = F_{k^\star+1:l} \circ \mathbf{S} \tag{1}$$

[1] To be consistent with existing literature (Lan et al., 2024; Yang et al., 2025c), we also use the term "3D VAE", although the mapping from 2D images to 3D scene is, technically, not a variational auto-encoder.

The front portion $F_{1:k^\star}$ of the 3D model is discarded – but if the clean encoder latents, after the affine warping $\mathbf{S}$, are (almost) the same as the activations $f_{k^\star}$, then the back portion will still produce the same output, $\hat{\boldsymbol{y}} \approx \boldsymbol{y}$. In other words, the stitched VAE $\mathcal{M}_{\text{stitched}}$ is an approximation of the original 3D model F. It retains much of the ability to map multi-view images to a 3D reconstruction and only requires a little fine-tuning to restore that ability.

Step 1: Finding the stitching index and initialization. To identify the layer k^* in the 3D model whose representation is most compatible with the VAE latent, we first push a set of N samples through the encoder $\mathcal{E}$ to obtain their latents $\mathbf{B} \in \mathbb{R}^{N \times D_\mathcal{E}}$. Here, $D_\mathcal{E}$ denotes the dimensionality of the encoder latent space, and D_F^k denotes the dimensionality of the feature (activation) at layer k. Then, we scan over candidate layers $k \in \{1, ..., l-1\}$ of the 3D model and, for each layer in turn, extract the activations $\mathbf{A}_k \in \mathbb{R}^{N \times D_F^k}$ and fit the linear stitching layer $\mathbf{S}^*{}_k \in \mathbb{R}^{D_\mathcal{E} \times D_F^k}$ that best recovers the activations of the 3D model at layer k, by solving a least-squares problem:

$$\mathbf{S}^*{}_k = \arg\min_{\mathbf{S}_k} \|\mathbf{B}\mathbf{S}_k - \mathbf{A}_k\|_F^2 = \left(\mathbf{B}^\top\mathbf{B}\right)^{-1}\mathbf{B}^\top\mathbf{A}_k. \tag{2}$$

Finally, we select the stitching layer $k^\star$ that leads to the smallest (mean squared) error, $k^\star = \arg\min_k \|\mathbf{B}\mathbf{S}_k^* - \mathbf{A}_k\|_F^2$, and assemble the 3D VAE by concatenating $\mathcal{E}$, $\mathbf{S}_{k^\star}^*$ and $F_{k^\star+1:l}$. Empirically, we find that most combinations of foundational VAEs and 3D feedforward models can be stitched in this manner, with minimal performance loss.

Step 2: Stitched decoder finetuning. To further reduce the remaining discrepancies between the newly assembled 3D VAE and the original 3D model, we finetune $\mathbf{S}$ and $F_{k^\star+1:l}$ to reproduce the predictions of the original 3D model $\boldsymbol{y}$, using them as pseudo-targets. Practical feedforward models produce multiple outputs (e.g., point maps, depth, poses), so we optimize a weighted sum of ℓ_1 losses for all of them. Note that the fine-tuning step is self-supervised and does not require labels. In our implementation, we restrict the stitching layer to a 3D convolution and employ LoRA (Hu et al., 2022) for updating $F_{k^\star+1:l}$, to prevent large deviations from the pretrained weights. For further details, see Appendix B.1.

3.2 Alignment via Direct Reward Finetuning

So far, we have assembled a 3D VAE with a strong, pretrained 3D decoder. However, during text-to-3D inference, the latents are not obtained from the encoder but generated from noise by the denoising loop conditioned on the text prompt. Therefore, we must also align the generative model itself with the 3D decoder, such that it produces decodable latents.

Previous work finetunes the generative network by minimizing generative losses over some multi-view dataset. Unfortunately, that strategy does not ensure 3D-consistent latents. Even if it did, the finetuning bypasses the decoder, hence there is no guarantee that the generated latents fall within the distribution expected by the 3D VAE and can be decoded to meaningful outputs.

To address the disconnect between the denosing loop and the 3D VAE, we adopt direct reward finetuning to align the two. In other words, we extend conventional, generative multi-view finetuning with reward maximization. The conventional generative loss L_{gen} uses paired data, i.e., multi-view images and corresponding prompts. In contrast, the proposed reward term $r(\cdot, c)$ relies only on the text prompt and requires no ground-truth images. Our total loss is defined as

$$L_{\text{total}} = L_{\text{gen}} - r\big(z_0(\theta, c, z_T), c\big), \tag{3}$$

where θ are the parameters of the video generative model, c represents the text prompt, z_T is the initial noise, and $z_0(\theta, c, z_T)$ is the final latent produced by the denoising loop.

Reward. The proposed reward function consists of three components that ensure high-quality and 3D-consistent generation. *(1) Multi-view Image Quality:* As we keep the encoder frozen, the generated latents can be decoded by the original video decoder $\mathcal{D}$ to obtain multi-view images. We evaluate these images against the input prompt using CLIP-based (Fang et al., 2024) and HPSv2 human preference scores (Wu et al., 2023) to promote prompt adherence and visual quality, similar to DanceGRPO (Xue et al., 2025). *(2) 3D Representation Quality:* To encourage high-quality 3D outputs after decoding with $\mathcal{D}_{\text{stitched}}$, we render the generated 3D scenes (pointmaps and/or 3DGS) back into 2D views and apply the same (CLIP + HPSv2) metrics to them as above. *(3) 3D Consistency:* To enforce 3D consistency, we render the 3D representation from the same viewpoints as the multi-view images reconstructed by the video decoder $\mathcal{D}$, using the camera poses predicted by the feedforward 3D model. We then compute a combination of ℓ_1-loss and LPIPS (Zhang et al., 2018) for each pair of decoded and rendered images belonging to the same viewpoint. The final (negative) reward is a weighted sum of these three losses. For further details, see Appendix B.2.

Alignment algorithm. To optimize the generative model according to the reward function above, we employ direct reward finetuning (Clark et al., 2023; Xu et al., 2023; Prabhudesai et al., 2024; Wu et al., 2024c; Shen et al., 2025). I.e., the model generates samples by unfolding the full denoising path, and the rewards computed from these samples are then backpropagated through the denoising chain. While the algorithm benefits from gradient-based feedback, it can also suffer from exploding gradient norms. To stabilize the optimization, we generalize the idea of DRTune (Wu et al., 2024c): gradients are detached from the inputs to the generative model, but retained during the update step to the next denoising state. In this way, reward propagation remains stable even at early denoising steps. Furthermore, we modify the optimizer for better computational efficiency by (i) randomized sampling, using fewer timesteps than during inference, and (ii) randomizing the subset of denoising steps where gradients are backpropagated, such that the model learns from diverse denoising trajectories. For further details, see Appendix B.2.

In summary, we perform joint, end-to-end alignment of the VAE and the generative model, unlike conventional multi-view fine-tuning that keeps them separate. Reward tuning ensures that, throughout the iterative denoising process, the generative model remains aligned with our 3D VAE and generates latents that suit the stitched decoder.

4 Experimental Results

In what follows, we demonstrate **VIST3A**'s text-to-3D generation performance. The main findings are that **VIST3A** clearly outperforms existing feedforward text-to-3DGS approaches and also offers high-quality text-to-pointmap generation. Moreover, we experimentally analyze our two core components, self-supervised *model stitching* and *alignment finetuning*.

4.1 Experimental Setups

We provide a high-level overview of the experimental setup. A complete description of evaluation protocols and training details can be found in Appendix C.

Target 3D models. We target last-generation foundational 3D vision models that have been trained on large-scale datasets, have demonstrated generality and reliable performance across diverse domains, and require only images as input. For our experiments, we select three representative state-of-the-art models: *(1) MVDUSt3R* (Tang et al., 2025) predicts pointmaps and Gaussian splats, *(2) VGGT* (Wang et al., 2025a) predicts pointmaps, depth maps and camera poses, and *(3) AnySplat* (Jiang et al., 2025) predicts Gaussian splats and camera poses.

Target video generators. Our primary video model is Wan 2.1 T2V large (Wan et al., 2025), a state-of-the-art text-to-video generator. To demonstrate the generality of VIST3A across different architectures, we additionally use several other latent video models, including CogVideoX (Yang et al., 2024b), SVD (Blattmann et al., 2023), and HunyuanVideo (Kong et al., 2024).

Training data. We finetune stitched VAEs on DL3DV-10K (Ling et al., 2024) and ScanNet (Dai et al., 2017), without 3D labels. To align the video generator in latent space, we utilize DL3DV-10K to compute the generative loss, with prompts from the HPSv2 training set (Wu et al., 2023).

4.2 Main Results: 3D Generation

Stitching Wan to the 3D models listed in Section 4.1 yields two types of generative models: (i) Text-to-3DGS when using AnySplat or MVDUSt3R as decoder; and (ii) Text-to-Pointmap when using VGGT or MVDUSt3R. Both variants are evaluated in the following.

Baselines. Important baselines for text-to-3DGS are SplatFlow (Go et al., 2025a), Director3D (Li et al., 2024), Prometheus3D (Yang et al., 2025c), and VideoRFSplat (Go et al., 2025b). Additionally, we include Matrix3D-omni (Yang et al., 2025d), to our knowledge, the only other model that unifies generation and reconstruction in latent space.

Table 1: **Quantitative results on T3Bench and SceneBench.**

Method	T3Bench (Object-centric)						SceneBench (Scene-level)					
	Imaging↑	Aesthetic↑	CLIP↑	Unified Reward			Imaging↑	Aesthetic↑	CLIP↑	Unified Reward		
				Align.↑	Coher.↑	Style↑				Align.↑	Coher.↑	Style↑
Matrix3D-omni	43.05	37.66	25.06	2.44	3.10	2.69	46.65	37.62	24.04	2.66	3.29	2.80
Director3D	54.32	53.33	30.94	3.25	3.43	3.05	47.79	52.81	29.31	3.36	3.67	3.20
Prometheus3D	47.46	44.32	29.15	2.84	3.12	2.66	44.73	45.85	28.57	3.20	3.36	2.98
SplatFlow	46.09	53.24	29.48	3.29	3.25	2.93	48.85	53.71	29.43	3.47	3.65	3.26
VideoRFSplat	46.52	39.50	30.13	3.12	3.24	3.09	58.19	51.71	29.76	3.58	3.63	3.30
VIST3A: Wan + MVDUSt3R	**58.83**	**56.55**	**32.75**	**3.56**	**3.89**	**3.56**	62.08	55.67	**30.26**	**3.72**	**3.97**	**3.47**
VIST3A: Wan + AnySplat	57.03	54.11	31.38	3.36	3.68	3.17	**64.87**	**56.96**	30.18	3.67	3.86	3.40

Table 2: **Quantitative results on DPG-Bench.**

Method	DPG-Bench				
	Global↑	Entity↑	Attribute↑	Relation↑	Other↑
Matrix3D-omni	53.32	42.44	56.23	37.12	10.32
Director3D	66.67	64.96	60.85	45.15	22.73
Prometheus3D	45.45	48.35	55.03	33.50	9.10
SplatFlow	69.70	68.43	65.55	50.49	40.91
VideoRFSplat	36.36	56.93	66.89	48.53	31.82
VIST3A: Wan + MVDUSt3R	**81.82**	84.31	**86.13**	68.93	**54.55**
VIST3A: Wan + AnySplat	78.79	**85.58**	84.12	**76.70**	45.45

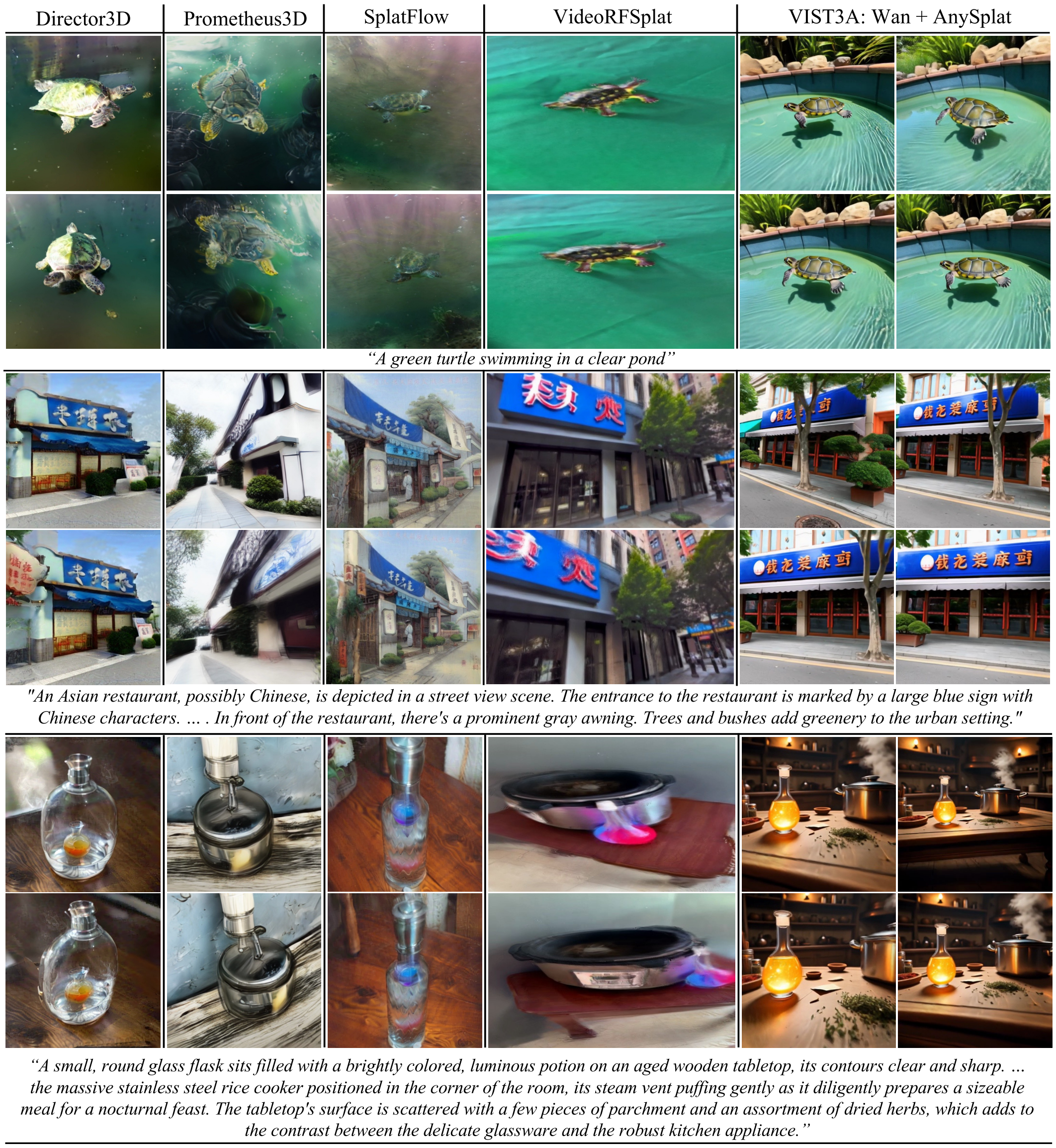
Table 3: **Stitching enhances NVS.**

Method	PSNR↑	SSIM↑	LPIPS↓
SplatFlow	19.10	0.671	0.278
VideoRFSplat	19.05	0.674	0.281
Prometheus3D	19.56	0.683	0.277
AnySplat	20.85	0.695	0.238
Hunyuan + AnySplat	21.17	0.710	0.242
SVD + AnySplat	**21.48**	**0.720**	**0.218**
CogVid + AnySplat	21.32	0.716	0.222
Wan + AnySplat	21.29	0.718	0.232

Evaluation protocol. We evaluate text-to-3DGS models on three benchmarks: T3bench (He et al., 2023) for object-centric generation, SceneBench (Yang et al., 2025c) for scene-level synthesis, and DPG-bench (Hu et al., 2024) to assess adherence to long, detailed prompts. On T3bench and SceneBench, we render images and compute Imaging Quality and Aesthetic Quality scores as defined by VBench (Huang et al., 2024) to assess visual fidelity, CLIP score (Hessel et al., 2021) for text-prompt alignment, and Alignment, Coherence, and Style scores according to Wang et al. (2025c) as comprehensive quality metrics. We prefer to avoid traditional no-reference metrics like NIQE (Mittal et al., 2012b) and BRISQUE (Mittal et al., 2012a) that have sometimes been used in the context of 3D generation, but lack a meaningful connection to the conditional generation task (e.g., they can be gambled by always returning the same sharp and colorful, high-scoring image, independent of the prompt). For DPG-bench, we follow the suggested protocol (Hu et al., 2024), but upgrade from the originally proposed language models to the more capable, UnifiedReward LLM (based on Qwen 7B). Text-to-pointmap models are evaluated qualitatively, as no established benchmarks or baselines exist.

Quantitative Results. Tables 1 and 2 show the results for the three text-to-3DGS benchmarks. Notably, both tested VIST3A variants exhibit superior performance across all datasets and evaluation metrics. On T3bench, both Wan+AnySplat and Wan+MVDUSt3R consistently outperform all baselines, with particularly large margins in Imaging Quality and Coherence score. For the more complex scene-level synthesis of SceneBench, our models reach Imaging Quality scores >60 and Coherence scores >3.8, again a marked improvement over prior art. On DPG-bench, our models greatly outperform the baselines, mostly scoring >75 (often even ≈ 85), values that previously seemed out of reach. The consistent gains on T3bench, SceneBench, and DPG-bench demonstrate the effectiveness and versatility of our stitching approach for text-based 3D scene generation. We attribute these results to the power of foundational contemporary video and 3D models, which our stitching and fine-tuning scheme unlocks for the purpose of 3D generative modeling.

Qualitative Results. Figure 4 qualitatively compares VIST3A (Wan+AnySplat) to several baselines. In line with the quantitative results, VIST3A produces superior, visually compelling, and geometrically coherent renderings that closely follow the input prompts; whereas previous methods tend to exhibit artifacts, structural distortions, and poor text alignment. Further qualitative results, including Wan+MVDUst3R and Wan+AnySplat variants of VIST3A, as well as text-to-pointmap examples, can be found in Appendix D.2. Interestingly, we find that, even without specific training on very long image sequences, VIST3A can generate coherent large-scale scenes by extending the number of frames generated by the LDM. This demonstrates that our framework preserves the ability of video generator and the 3D decoder to handle long sequences. Examples are depicted in Fig. 16.



“A green turtle swimming in a clear pond”

"An Asian restaurant, possibly Chinese, is depicted in a street view scene. The entrance to the restaurant is marked by a large blue sign with Chinese characters. ... . In front of the restaurant, there's a prominent gray awning. Trees and bushes add greenery to the urban setting."

“A small, round glass flask sits filled with a brightly colored, luminous potion on an aged wooden tabletop, its contours clear and sharp. ... the massive stainless steel rice cooker positioned in the corner of the room, its steam vent puffing gently as it diligently prepares a sizeable meal for a nocturnal feast. The tabletop's surface is scattered with a few pieces of parchment and an assortment of dried herbs, which adds to the contrast between the delicate glassware and the robust kitchen appliance.”

Figure 4: **Qualitative results for 3DGS generation.** We show samples from T3Bench (top), SceneBench (middle), and DPG-bench (bottom). VIST3A generates realistic and crisp 3D scenes and adheres to intricate details in the prompt.

4.3 Main Results: Model Stitching

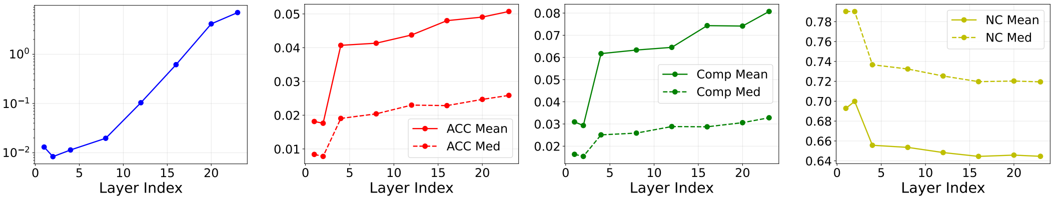
Stitching the 3D foundation models from Section 4.1 with a video VAE yields two variants: a VAE for Gaussian splats (AnySplat + video VAE) or a VAE capable of reconstructing pointmaps and camera poses (MVDust3R or VGGT + video VAE). In the following, we evaluate both variants.

Evaluation protocol. For 3DGS models, we evaluate novel-view synthesis on RealEstate10K (Zhou et al., 2018), with 8 source and 4 target images. For 3D reconstruction models, we follow Pi3 (Wang et al., 2025d) and assess pointmap quality on 7Scenes (Shotton et al., 2013) and ETH3D (Schöps et al., 2017), and camera pose estimation on RealEstate10K and ScanNet (Dai et al., 2017). Specifically, Accuracy (Acc.), Completion (Comp.), and Normal Consistency (N.C.) are used for pointmap estimation, while camera pose estimation is evaluated with Relative Rotation Accuracy (RRA) and Relative Translation Accuracy (RTA) at 5° and their AUC up to 30°.

Novel view synthesis. Table 3 reports results on RealEstate10K. Stitching AnySplat onto any video model always improves over using AnySplat alone. We attribute the gains to the richer appearance representation of video VAE latents. The experiment is consistent with the results of Wonder-

Table 4: **Results of point map reconstruction with stitched models.**

Method	Pointmap Estimation												Camera Pose Estimation					
	7-Scenes						ETH3D						RealEstate10K			ScanNet		
	Acc.↓		Comp.↓		NC.↑		Acc.↓		Comp.↓		NC.↑		RRA@5↑	RTA@5↑	AUC@30↑	ATE↓	RPE-T↓	RPE-R↓
	Mean	Med.	Mean	Med.	Mean	Med.	Mean	Med.	Mean	Med.	Mean	Med.						
MVDUSt3R	0.026	0.011	0.030	0.013	0.730	0.838	0.400	0.291	0.376	0.159	0.805	0.905	98.66	12.91	42.34	0.015	0.019	0.691
VGGT	0.020	0.008	0.029	0.016	0.694	0.790	0.263	0.188	0.197	0.120	0.855	0.961	99.51	15.75	50.06	0.015	0.015	0.500
Hunyuan+MVDUSt3R	0.027	0.012	0.032	0.012	0.699	0.806	0.405	0.288	0.399	0.166	0.802	0.887	98.36	12.40	41.97	0.016	0.019	0.668
SVD+MVDUSt3R	0.026	0.011	0.030	0.013	0.727	0.834	0.410	0.310	0.387	0.168	0.804	0.899	98.12	12.67	41.69	0.016	0.020	0.690
CogVid+MVDUSt3R	0.028	0.012	0.033	0.014	0.699	0.808	0.412	0.281	0.387	0.157	0.781	0.888	98.29	12.36	41.96	0.016	0.019	0.680
Wan+MVDUSt3R	0.027	0.011	0.032	0.012	0.712	0.825	0.401	0.297	0.386	0.164	0.797	0.910	98.28	12.30	42.12	0.016	0.019	0.680
Wan+VGGT	0.018	0.008	0.032	0.015	0.693	0.790	0.265	0.166	0.193	0.121	0.837	0.960	99.65	15.98	50.86	0.014	0.015	0.520



(a) log-MSE value in Eq. 2 (b) Acc.↓ (c) Comp.↓ (d) NC.↑

Figure 5: **MSE and pointmap quality on 7-Scenes vs. to stitching layer.** Lower MSE in the stitching layer correlates with better 3D reconstruction.

land (Liang et al., 2025), where operating in latent space rather than RGB space also benefits 3DGS. Moreover, our stitched VAEs outperform the earlier VAE-based approaches. Remarkably, we surpass Prometheus3D and VideoRFSplat despite their use of camera poses and large-scale training data, showing that stitching high-performance 3D models is indeed an effective strategy to obtain powerful 3D VAEs.

Pointmap reconstruction results. Table 4 shows that stitching preserves the accuracy and completeness of the original 3D foundation models: both pointmap quality and camera pose accuracy barely change when using video encoder latents as input. The results confirm that stitching achieves its goal, to take advantage of the pretrained models' 3D reconstruction capabilities and repurpose them for generative modeling, without relying on large training datasets or labels.

4.4 Ablations

Effectiveness of MSE for finding stitching layer (Sec 3.1). We pick the best layer for stitching according to a fairly simple criterion, namely the one that best supports a linear transfer of the encoder latents. To analyze the impact of this design, we train stitched decoders for the combination (Wan+VGGT) while varying the stitching index. In Fig. 5, we see that layers with lower stitching residual indeed yield better pointmaps, supporting the MSE of the linear stitching layer as our selection criterion.

This empirical trend is also consistent with existing theory: Theorem 1 in Insulla et al. (2025) shows that the stitching risk of the hybrid network, obtained by connecting the source model's early layers f_1 with the target model's latter layers g_2 via a linear map $S_{1,2}$, is upper-bounded by the MSE at the stitching layer,

$$\mathbb{E}\big[\|g_2(S_{1,2}f_1)(x) - g_2(f_2)(x)\|^2\big] \leq \kappa_2^2\, \mathbb{E}\big[\|S_{1,2}f_1(x) - f_2(x)\|^2\big], \tag{4}$$

where κ_2 is the Lipschitz constant of g_2. Thus, an MSE is related to the upper bound on the stitching error, supporting our use of MSE.

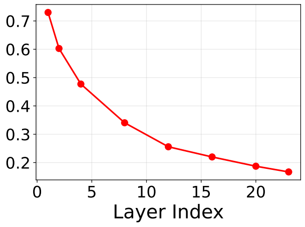


Figure 6: **CKA visualization.**

Furthermore, motivated by the observation of Insulla et al. (2025) that the right-hand side of Eq. 4 takes a similar form of kernel alignment, we investigate whether CKA (Kornblith et al., 2019) can track the trend of final performance. Figure 6 reports the CKA between latent representation of Wan VAE and the representations at different layer indices of VGGT (larger values indicate higher similarity). As shown, CKA captures the overall degradation in performance as the layer index increases; however, it is less precise than MSE in identifying the best layer, failing to capture that the best performance is achieved at layer 2. These results suggest that, in our setting, MSE is a more reliable indicator of transferability than CKA.

Impact of direct reward finetuning (Sec 3.2). As shown in Appendix D.1, direct reward finetuning is more effective than a pretrained video model on its own, as well as that same model finetuned on multi-view data, with each reward component contributing to the overall performance.

Benefits of integrated vs. sequential 3D generation. In Appendix D.2, we observe that an integrated approach is more robust to noise in the latent space, which suggests it may lead to more consistent 3D reconstruction from noise in the generation process.

5 Conclusion

We have presented VIST3A, a framework for training latent diffusion models that generate 3D content from text prompts. Our key idea is to employ model stitching as a way to integrate the generative abilities of modern video models with the 3D understanding of recent feedforward 3D models. We found that this strategy indeed leads to high-quality 3D VAEs, while not requiring labeled data or massive training runs. To then align a latent-space video generator with the stitched 3D decoder it feeds into, we design a reward-based finetuning strategy. Together, these two measures yield a family of text-to-3D models with high-quality, geometrically consistent 3D outputs. In passing, they extend 3D generation to other outputs of foundational 3D models, such as pointmaps and depthmaps. More broadly, we see great potential for model stitching as a general tool to combine two or more foundational neural networks, including latent generative models, into powerful end-to-end solutions.

References

Michael Samuel Albergo and Eric Vanden-Eijnden. Building normalizing flows with stochastic interpolants. In *International Conference on Learning Representations*, 2023.

Yamini Bansal, Preetum Nakkiran, and Boaz Barak. Revisiting model stitching to compare neural representations. *Advances in Neural Information Processing Systems*, 34:225–236, 2021.

Kevin Black, Michael Janner, Yilun Du, Ilya Kostrikov, and Sergey Levine. Training diffusion models with reinforcement learning. In *International Conference on Learning Representations*, 2024.

Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, et al. Stable video diffusion: Scaling latent video diffusion models to large datasets. *preprint arXiv:2311.15127*, 2023.

David Charatan, Sizhe Lester Li, Andrea Tagliasacchi, and Vincent Sitzmann. pixelSplat: 3d gaussian splats from image pairs for scalable generalizable 3d reconstruction. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19457–19467, 2024.

Hansheng Chen, Jiatao Gu, Anpei Chen, Wei Tian, Zhuowen Tu, Lingjie Liu, and Hao Su. Single-stage diffusion NeRF: A unified approach to 3d generation and reconstruction. In *IEEE/CVF International Conference on Computer Vision*, pp. 2416–2425, 2023.

Luxi Chen, Zihan Zhou, Min Zhao, Yikai Wang, Ge Zhang, Wenhao Huang, Hao Sun, Ji-Rong Wen, and Chongxuan Li. FlexWorld: Progressively expanding 3d scenes for flexible-view synthesis. *preprint arXiv:2503.13265*, 2025.

Yuedong Chen, Haofei Xu, Chuanxia Zheng, Bohan Zhuang, Marc Pollefeys, Andreas Geiger, Tat-Jen Cham, and Jianfei Cai. MVSplat: Efficient 3d gaussian splatting from sparse multi-view images. In *European Conference on Computer Vision*, pp. 370–386, 2024a.

Zilong Chen, Feng Wang, Yikai Wang, and Huaping Liu. Text-to-3d using gaussian splatting. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21401–21412, 2024b.

Kevin Clark, Paul Vicol, Kevin Swersky, and David J Fleet. Directly fine-tuning diffusion models on differentiable rewards. *preprint arXiv:2309.17400*, 2023.

Adrián Csiszárik, Péter Kőrösi-Szabó, Akos Matszangosz, Gergely Papp, and Dániel Varga. Similarity and matching of neural network representations. *Advances in Neural Information Processing Systems*, 34:5656–5668, 2021.

Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. ScanNet: Richly-annotated 3d reconstructions of indoor scenes. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5828–5839, 2017.

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 *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13142–13153, 2023.

Paul Engstler, Aleksandar Shtedritski, Iro Laina, Christian Rupprecht, and Andrea Vedaldi. SynCity: Training-free generation of 3d worlds. *preprint arXiv:2503.16420*, 2025.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *International Conference on Machine Learning*, 2024.

Ying Fan, Olivia Watkins, Yuqing Du, Hao Liu, Moonkyung Ryu, Craig Boutilier, Pieter Abbeel, Mohammad Ghavamzadeh, Kangwook Lee, and Kimin Lee. Reinforcement learning for fine-tuning text-to-image diffusion models. In *Advances in Neural Information Processing Systems*, 2023.

Alex Fang, Albin Madappally Jose, Amit Jain, Ludwig Schmidt, Alexander T Toshev, and Vaishaal Shankar. Data filtering networks. In *International Conference on Learning Representations*, 2024.

Hao Feng, Zhi Zuo, Jia-hui Pan, Ka-hei Hui, Yihua Shao, Qi Dou, Wei Xie, and Zhengzhe Liu. WonderVerse: Extendable 3d scene generation with video generative models. *preprint arXiv:2503.09160*, 2025.

Rafail Fridman, Amit Abecasis, Yoni Kasten, and Tali Dekel. SceneScape: Text-driven consistent scene generation. *Advances in Neural Information Processing Systems*, 36:39897–39914, 2023.

Yasutaka Furukawa, Carlos Hernández, et al. Multi-view stereo: A tutorial. *Foundations and trends® in Computer Graphics and Vision*, 9(1-2):1–148, 2015.

Ruiqi Gao, Aleksander Holynski, Philipp Henzler, Arthur Brussee, Ricardo Martin Brualla, Pratul Srinivasan, Jonathan Barron, and Ben Poole. CAT3D: Create anything in 3d with multi-view diffusion models. *Advances in Neural Information Processing Systems*, 37:75468–75494, 2024.

Hyojun Go, Byeongjun Park, Jiho Jang, Jin-Young Kim, Soonwoo Kwon, and Changick Kim. SplatFlow: Multi-view rectified flow model for 3d gaussian splatting synthesis. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21524–21536, 2025a.

Hyojun Go, Byeongjun Park, Hyelin Nam, Byung-Hoon Kim, Hyungjin Chung, and Changick Kim. VideoRFSplat: Direct scene-level text-to-3d gaussian splatting generation with flexible pose and multi-view joint modeling. *preprint arXiv:2503.15855*, 2025b.

Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge University Press, 2003.

Yuze He, Yushi Bai, Matthieu Lin, Wang Zhao, Yubin Hu, Jenny Sheng, Ran Yi, Juanzi Li, and Yong-Jin Liu. t3Bench: Benchmarking current progress in text-to-3d generation. *preprint arXiv:2310.02977*, 2023.

Paul Henderson, Melonie de Almeida, Daniela Ivanova, and Titas Anciukevičius. Sampling 3d gaussian scenes in seconds with latent diffusion models. *preprint arXiv:2406.13099*, 2024.

Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. CLIPScore: A reference-free evaluation metric for image captioning. *preprint arXiv:2104.08718*, 2021.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

Yicong Hong, Kai Zhang, Jiuxiang Gu, Sai Bi, Yang Zhou, Difan Liu, Feng Liu, Kalyan Sunkavalli, Trung Bui, and Hao Tan. LRM: Large reconstruction model for single image to 3d. *preprint arXiv:2311.04400*, 2023.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

Xiwei Hu, Rui Wang, Yixiao Fang, Bin Fu, Pei Cheng, and Gang Yu. ELLA: Equip diffusion models with LLM for enhanced semantic alignment. *preprint arXiv:2403.05135*, 2024.

Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, et al. VBench: Comprehensive benchmark suite for video generative models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21807–21818, 2024.

Francesco Insulla, Shuo Huang, and Lorenzo Rosasco. Towards a learning theory of representation alignment. In *The Thirteenth International Conference on Learning Representations*, 2025. URL `https://openreview.net/forum?id=DShqJA1Z64`.

Lihan Jiang, Yucheng Mao, Linning Xu, Tao Lu, Kerui Ren, Yichen Jin, Xudong Xu, Mulin Yu, Jiangmiao Pang, Feng Zhao, et al. AnySplat: Feed-forward 3d gaussian splatting from unconstrained views. *preprint arXiv:2505.23716*, 2025.

Junjie Ke, Qifei Wang, Yilin Wang, Peyman Milanfar, and Feng Yang. Musiq: Multi-scale image quality transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 5148–5157, 2021.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d Gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4):139–1, 2023.

Weijie Kong, Qi Tian, Zijian Zhang, Rox Min, Zuozhuo Dai, Jin Zhou, Jiangfeng Xiong, Xin Li, Bo Wu, Jianwei Zhang, et al. HunyuanVideo: A systematic framework for large video generative models. *preprint arXiv:2412.03603*, 2024.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *International Conference on Machine Learning*, pp. 3519–3529, 2019.

Yushi Lan, Shangchen Zhou, Zhaoyang Lyu, Fangzhou Hong, Shuai Yang, Bo Dai, Xingang Pan, and Chen Change Loy. GaussianAnything: Interactive point cloud flow matching for 3d object generation. *preprint arXiv:2411.08033*, 2024.

Kimin Lee, Hao Liu, Moonkyung Ryu, Olivia Watkins, Yuqing Du, Craig Boutilier, Pieter Abbeel, Mohammad Ghavamzadeh, and Shixiang Shane Gu. Aligning text-to-image models using human feedback. *preprint arXiv:2302.12192*, 2023.

Karel Lenc and Andrea Vedaldi. Understanding image representations by measuring their equivariance and equivalence. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 991–999, 2015.

Vincent Leroy, Yohann Cabon, and Jérôme Revaud. Grounding image matching in 3d with MASt3R. In *European Conference on Computer Vision*, pp. 71–91, 2024.

Jiahao Li, Hao Tan, Kai Zhang, Zexiang Xu, Fujun Luan, Yinghao Xu, Yicong Hong, Kalyan Sunkavalli, Greg Shakhnarovich, and Sai Bi. Instant3d: Fast text-to-3d with sparse-view generation and large reconstruction model. *preprint arXiv:2311.06214*, 2023.

Weiyu Li, Jiarui Liu, Hongyu Yan, Rui Chen, Yixun Liang, Xuelin Chen, Ping Tan, and Xiaoxiao Long. CraftsMan3D: High-fidelity mesh generation with 3d native diffusion and interactive geometry refiner. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5307–5317, 2025a.

Xinyang Li, Zhangyu Lai, Linning Xu, Yansong Qu, Liujuan Cao, Shengchuan Zhang, Bo Dai, and Rongrong Ji. Director3D: Real-world camera trajectory and 3d scene generation from text. *Advances in Neural Information Processing Systems*, pp. 75125–75151, 2024.

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

Hanwen Liang, Junli Cao, Vidit Goel, Guocheng Qian, Sergei Korolev, Demetri Terzopoulos, Konstantinos N Plataniotis, Sergey Tulyakov, and Jian Ren. Wonderland: Navigating 3d scenes from a single image. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 798–810, 2025.

Chenguo Lin, Panwang Pan, Bangbang Yang, Zeming Li, and Yadong MU. DiffSplat: Repurposing image diffusion models for scalable gaussian splat generation. In *International Conference on Learning Representations*, 2025.

Lu Ling, Yichen Sheng, Zhi Tu, Wentian Zhao, Cheng Xin, Kun Wan, Lantao Yu, Qianyu Guo, Zixun Yu, Yawen Lu, et al. DL3DV-10K: A large-scale scene dataset for deep learning-based 3d vision. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22160–22169, 2024.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *International Conference on Learning Representations*, 2023.

Fangfu Liu, Wenqiang Sun, Hanyang Wang, Yikai Wang, Haowen Sun, Junliang Ye, Jun Zhang, and Yueqi Duan. ReconX: Reconstruct any scene from sparse views with video diffusion model. *preprint arXiv:2408.16767*, 2024.

Jie Liu, Gongye Liu, Jiajun Liang, Yangguang Li, Jiaheng Liu, Xintao Wang, Pengfei Wan, Di Zhang, and Wanli Ouyang. Flow-GRPO: Training flow matching models via online RL. *preprint arXiv:2505.05470*, 2025.

Xingchao Liu, Chengyue Gong, and qiang liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *International Conference on Learning Representations*, 2023.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Xuyi Meng, Chen Wang, Jiahui Lei, Kostas Daniilidis, Jiatao Gu, and Lingjie Liu. Zero-1-to-G: Taming pretrained 2d diffusion model for direct 3d generation. *arXiv preprint arXiv:2501.05427*, 2025.

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. *Communications of the ACM*, 65(1):99–106, 2021.

Anish Mittal, Anush Krishna Moorthy, and Alan Conrad Bovik. No-reference image quality assessment in the spatial domain. *IEEE Transactions on Image Processing*, 21(12):4695–4708, 2012a.

Anish Mittal, Rajiv Soundararajan, and Alan C Bovik. Making a “completely blind” image quality analyzer. *IEEE Signal Processing Letters*, 20(3):209–212, 2012b.

Shentong Mo, Enze Xie, Ruihang Chu, Lanqing Hong, Matthias Niessner, and Zhenguo Li. DiT-3D: Exploring plain diffusion transformers for 3d shape generation. *Advances in Neural Information Processing Systems*, 36:67960–67971, 2023.

Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics*, 41(4):1–15, 2022.

Chaojun Ni, Xiaofeng Wang, Zheng Zhu, Weijie Wang, Haoyun Li, Guosheng Zhao, Jie Li, Wenkang Qin, Guan Huang, and Wenjun Mei. WonderTurbo: Generating interactive 3d world in 0.72 seconds. *preprint arXiv:2504.02261*, 2025.

Alex Nichol, Heewoo Jun, Prafulla Dhariwal, Pamela Mishkin, and Mark Chen. Point-E: A system for generating 3d point clouds from complex prompts. *preprint arXiv:2212.08751*, 2022.

Julian Ost, Andrea Ramazzina, Amogh Joshi, Maximilian Bömer, Mario Bijelic, and Felix Heide. LSD-3D: Large-scale 3d driving scene generation with geometry grounding. *preprint arXiv:2508.19204*, 2025.

Zizheng Pan, Jianfei Cai, and Bohan Zhuang. Stitchable neural networks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16102–16112, 2023.

Byeongjun Park, Hyojun Go, Hyelin Nam, Byung-Hoon Kim, Hyungjin Chung, and Changick Kim. Steerx: Creating any camera-free 3d and 4d scenes with geometric steering. *arXiv preprint arXiv:2503.12024*, 2025.

Xue Bin Peng, Aviral Kumar, Grace Zhang, and Sergey Levine. Advantage-weighted regression: Simple and scalable off-policy reinforcement learning. *preprint arXiv:1910.00177*, 2019.

Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. DreamFusion: Text-to-3d using 2d diffusion. In *International Conference on Learning Representations*, 2023.

Mihir Prabhudesai, Russell Mendonca, Zheyang Qin, Katerina Fragkiadaki, and Deepak Pathak. Video diffusion alignment via reward gradients. *preprint arXiv:2407.08737*, 2024.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36:53728–53741, 2023.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10684–10695, 2022.

Aditya Sanghi, Rao Fu, Vivian Liu, Karl D.D. Willis, Hooman Shayani, Amir H. Khasahmadi, Srinath Sridhar, and Daniel Ritchie. CLIP-Sculptor: Zero-shot generation of high-fidelity and diverse shapes from natural language. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18339–18348, 2023.

Johannes L Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4104–4113, 2016.

Johannes L Schönberger, Enliang Zheng, Jan-Michael Frahm, and Marc Pollefeys. Pixelwise view selection for unstructured multi-view stereo. In *European Conference on Computer Vision*, pp. 501–518, 2016.

Thomas Schöps, Johannes L Schonberger, Silvano Galliani, Torsten Sattler, Konrad Schindler, Marc Pollefeys, and Andreas Geiger. A multi-view stereo benchmark with high-resolution images and multi-camera videos. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3260–3269, 2017.

Katja Schwarz, Norman Mueller, and Peter Kontschieder. Generative gaussian splatting: Generating 3d scenes with video diffusion priors. *preprint arXiv:2503.13272*, 2025.

Xiangwei Shen, Zhimin Li, Zhantao Yang, Shiyi Zhang, Yingfang Zhang, Donghao Li, Chunyu Wang, Qinglin Lu, and Yansong Tang. Directly aligning the full diffusion trajectory with fine-grained human preference. *preprint arXiv:2509.06942*, 2025.

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

Jamie Shotton, Ben Glocker, Christopher Zach, Shahram Izadi, Antonio Criminisi, and Andrew Fitzgibbon. Scene coordinate regression forests for camera relocalization in RGB-D images. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2930–2937, 2013.

Jaidev Shriram, Alex Trevithick, Lingjie Liu, and Ravi Ramamoorthi. RealmDreamer: Text-driven 3d scene generation with inpainting and depth diffusion. In *International Conference on 3D Vision*, 2025.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, pp. 2256–2265, 2015.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *preprint arXiv:2011.13456*, 2020.

Wenqiang Sun, Shuo Chen, Fangfu Liu, Zilong Chen, Yueqi Duan, Jun Zhang, and Yikai Wang. DimensionX: Create any 3d and 4d scenes from a single image with controllable video diffusion. *preprint arXiv:2411.04928*, 2024.

Stanislaw Szymanowicz, Jason Y Zhang, Pratul Srinivasan, Ruiqi Gao, Arthur Brussee, Aleksander Holynski, Ricardo Martin-Brualla, Jonathan T Barron, and Philipp Henzler. Bolt3D: Generating 3d scenes in seconds. *preprint arXiv:2503.14445*, 2025.

Jiaxiang Tang, Zhaoxi Chen, Xiaokang Chen, Tengfei Wang, Gang Zeng, and Ziwei Liu. LGM: Large multi-view gaussian model for high-resolution 3d content creation. In *European Conference on Computer Vision*, pp. 1–18, 2024a.

Jiaxiang Tang, Jiawei Ren, Hang Zhou, Ziwei Liu, and Gang Zeng. DreamGaussian: Generative Gaussian splatting for efficient 3d content creation. In *International Conference on Learning Representations*, 2024b.

Zhenggang Tang, Yuchen Fan, Dilin Wang, Hongyu Xu, Rakesh Ranjan, Alexander Schwing, and Zhicheng Yan. MV-DUSt3R+: Single-stage scene reconstruction from sparse views in 2 seconds. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5283–5293, 2025.

Genmo Team. Mochi 1. `https://github.com/genmoai/models`, 2024.

Qwen Team. Qwen2.5-vl, January 2025. URL `https://qwenlm.github.io/blog/qwen2.5-vl/`.

Arash Vahdat, Francis Williams, Zan Gojcic, Or Litany, Sanja Fidler, Karsten Kreis, et al. LION: Latent point diffusion models for 3d shape generation. *Advances in Neural Information Processing Systems*, 35:10021–10039, 2022.

Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, et al. Wan: Open and advanced large-scale video generative models. *preprint arXiv:2503.20314*, 2025.

Haiping Wang, Yuan Liu, Ziwei Liu, Wenping Wang, Zhen Dong, and Bisheng Yang. VistaDream: Sampling multiview consistent images for single-view scene reconstruction. *preprint arXiv:2410.16892*, 2024a.

Haochen Wang, Xiaodan Du, Jiahao Li, Raymond A Yeh, and Greg Shakhnarovich. Score Jacobian chaining: Lifting pretrained 2d diffusion models for 3d generation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12619–12629, 2023a.

Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. VGGT: Visual geometry grounded transformer. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5294–5306, 2025a.

Qianqian Wang, Yifei Zhang, Aleksander Holynski, Alexei A Efros, and Angjoo Kanazawa. Continuous 3d perception model with persistent state. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10510–10522, 2025b.

Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. DUSt3R: Geometric 3d vision made easy. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 20697–20709, 2024b.

Yibin Wang, Yuhang Zang, Hao Li, Cheng Jin, and Jiaqi Wang. Unified reward model for multimodal understanding and generation. *arXiv preprint arXiv:2503.05236*, 2025c.

Yifan Wang, Jianjun Zhou, Haoyi Zhu, Wenzheng Chang, Yang Zhou, Zizun Li, Junyi Chen, Jiangmiao Pang, Chunhua Shen, and Tong He. $\pi 3$: Scalable permutation-equivariant visual geometry learning. *preprint arXiv:2507.13347*, 2025d.

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 *Advances in Neural Information Processing Systems*, pp. 8406–8441, 2023b.

Rundi Wu, Ben Mildenhall, Philipp Henzler, Keunhong Park, Ruiqi Gao, Daniel Watson, Pratul P Srinivasan, Dor Verbin, Jonathan T Barron, Ben Poole, et al. ReconFusion: 3d reconstruction with diffusion priors. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21551–21561, 2024a.

Shuang Wu, Youtian Lin, Feihu Zhang, Yifei Zeng, Jingxi Xu, Philip Torr, Xun Cao, and Yao Yao. Direct3D: Scalable image-to-3d generation via 3d latent diffusion transformer. *Advances in Neural Information Processing Systems*, 37:121859–121881, 2024b.

Xiaoshi Wu, Yiming Hao, Keqiang Sun, Yixiong Chen, Feng Zhu, Rui Zhao, and Hongsheng Li. Human preference score v2: A solid benchmark for evaluating human preferences of text-to-image synthesis. *preprint arXiv:2306.09341*, 2023.

Xiaoshi Wu, Yiming Hao, Manyuan Zhang, Keqiang Sun, Zhaoyang Huang, Guanglu Song, Yu Liu, and Hongsheng Li. Deep reward supervisions for tuning text-to-image diffusion models. In *European Conference on Computer Vision*, pp. 108–124, 2024c.

Jiale Xu, Weihao Cheng, Yiming Gao, Xintao Wang, Shenghua Gao, and Ying Shan. InstantMesh: Efficient 3d mesh generation from a single image with sparse-view large reconstruction models. *preprint arXiv:2404.07191*, 2024a.

Jiazheng Xu, Xiao Liu, Yuchen Wu, Yuxuan Tong, Qinkai Li, Ming Ding, Jie Tang, and Yuxiao Dong. ImageReward: Learning and evaluating human preferences for text-to-image generation. *Advances in Neural Information Processing Systems*, 36:15903–15935, 2023.

Yinghao Xu, Zifan Shi, Wang Yifan, Hansheng Chen, Ceyuan Yang, Sida Peng, Yujun Shen, and Gordon Wetzstein. GRM: Large Gaussian reconstruction model for efficient 3d reconstruction and generation. In *European Conference on Computer Vision*, pp. 1–20, 2024b.

Zeyue Xue, Jie Wu, Yu Gao, Fangyuan Kong, Lingting Zhu, Mengzhao Chen, Zhiheng Liu, Wei Liu, Qiushan Guo, Weilin Huang, et al. DanceGRPO: Unleashing GRPO on visual generation. *preprint arXiv:2505.07818*, 2025.

Jianing Yang, Alexander Sax, Kevin J Liang, Mikael Henaff, Hao Tang, Ang Cao, Joyce Chai, Franziska Meier, and Matt Feiszli. Fast3R: Towards 3d reconstruction of 1000+ images in one forward pass. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21924–21935, 2025a.

Kai Yang, Jian Tao, Jiafei Lyu, Chunjiang Ge, Jiaxin Chen, Weihan Shen, Xiaolong Zhu, and Xiu Li. Using human feedback to fine-tune diffusion models without any reward model. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8941–8951, 2024a.

Shuai Yang, Jing Tan, Mengchen Zhang, Tong Wu, Gordon Wetzstein, Ziwei Liu, and Dahua Lin. LayerPano3D: Layered 3d panorama for hyper-immersive scene generation. In *ACM SIGGRAPH*, 2025b.

Xingyi Yang, Daquan Zhou, Songhua Liu, Jingwen Ye, and Xinchao Wang. Deep model reassembly. *Advances in Neural Information Processing Systems*, 35:25739–25753, 2022.

Yuanbo Yang, Jiahao Shao, Xinyang Li, Yujun Shen, Andreas Geiger, and Yiyi Liao. Prometheus: 3d-aware latent diffusion models for feed-forward text-to-3d scene generation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2857–2869, 2025c.

Zhongqi Yang, Wenhang Ge, Yuqi Li, Jiaqi Chen, Haoyuan Li, Mengyin An, Fei Kang, Hua Xue, Baixin Xu, Yuyang Yin, et al. Matrix-3d: Omnidirectional explorable 3d world generation. *preprint arXiv:2508.08086*, 2025d.

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. *preprint arXiv:2408.06072*, 2024b.

Botao Ye, Sifei Liu, Haofei Xu, Xueting Li, Marc Pollefeys, Ming-Hsuan Yang, and Songyou Peng. No pose, no problem: Surprisingly simple 3d gaussian splats from sparse unposed images. *preprint arXiv:2410.24207*, 2024.

Hong-Xing Yu, Haoyi Duan, Junhwa Hur, Kyle Sargent, Michael Rubinstein, William T Freeman, Forrester Cole, Deqing Sun, Noah Snavely, Jiajun Wu, et al. WonderJourney: Going from anywhere to everywhere. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6658–6667, 2024a.

Hong-Xing Yu, Haoyi Duan, Charles Herrmann, William T Freeman, and Jiajun Wu. WonderWorld: Interactive 3d scene generation from a single image. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5916–5926, 2025.

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

Bowen Zhang, Yiji Cheng, Jiaolong Yang, Chunyu Wang, Feng Zhao, Yansong Tang, Dong Chen, and Baining Guo. GaussianCube: Structuring gaussian splatting using optimal transport for 3d generative modeling. *preprint arXiv:2403.19655*, 2024a.

Kai Zhang, Sai Bi, Hao Tan, Yuanbo Xiangli, Nanxuan Zhao, Kalyan Sunkavalli, and Zexiang Xu. GS-LRM: Large reconstruction model for 3d gaussian splatting. In *European Conference on Computer Vision*, pp. 1–19, 2024b.

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018.

Shangzhan Zhang, Jianyuan Wang, Yinghao Xu, Nan Xue, Christian Rupprecht, Xiaowei Zhou, Yujun Shen, and Gordon Wetzstein. FLARE: Feed-forward geometry, appearance and camera estimation from uncalibrated sparse views. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21936–21947, 2025.

Zibo Zhao, Wen Liu, Xin Chen, Xianfang Zeng, Rui Wang, Pei Cheng, Bin Fu, Tao Chen, Gang Yu, and Shenghua Gao. Michelangelo: Conditional 3d shape generation based on shape-image-text aligned latent representation. *Advances in Neural Information Processing Systems*, pp. 73969–73982, 2023.

Kaizhi Zheng, Ruijian Zhang, Jing Gu, Jie Yang, and Xin Eric Wang. Constructing a 3d town from a single image. *preprint arXiv:2505.15765*, 2025.

Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: Learning view synthesis using multiplane images. *preprint arXiv:1805.09817*, 2018.

A Extended related works

Pipeline-based 3D generation. A line of recent works follows a pipeline design, chaining together multiple modules and models. Typically, the first stage generates multi-view images from text or a single input image, followed by a separate reconstruction model that lifts these views into a 3D representation (Tang et al., 2024a; Xu et al., 2024b; Zhang et al., 2024b; Li et al., 2023; Park et al., 2025), with large models such as LRM (Hong et al., 2023) often used for this step. However, since the generative and reconstruction stages are trained and executed independently, errors accumulate across these parts (e.g., view inconsistency, texture flicker). Moreover, such pipeline schemes are less robust to latent-space perturbations than approaches where generation and reconstruction are performed jointly in the same latent space (see Section D.2).

A second category of methods (Liu et al., 2024; Yu et al., 2024b; Gao et al., 2024; Sun et al., 2024; Wang et al., 2024a) also generates multi-view images before lifting them into 3D, but replaces large pretrained reconstruction models with per-scene optimization of NeRFs (Mildenhall et al., 2021) or 3D Gaussian Splatting (3DGS) (Kerbl et al., 2023). While this strategy avoids reliance on pretrained decoders, it remains prone to error accumulation and requires costly per-scene optimization, making inference slow and computationally expensive.

A third line of works introduces progressive expansion and refinement pipelines (Yu et al., 2024a; Ni et al., 2025; Chen et al., 2025; Fridman et al., 2023; Feng et al., 2025; Yu et al., 2025). Some adopt iterative warping and inpainting strategies (Yu et al., 2024a; Ni et al., 2025; Fridman et al., 2023; Yu et al., 2025), while others leverage video generative models to unfold 3D scenes in a progressive manner (Chen et al., 2025; Feng et al., 2025). Beyond these, additional works propose elaborate multi-stage pipelines that further increase complexity (Yang et al., 2025b; Ost et al., 2025). However, such designs are overly complex and suffer from slow inference.

Alignment for text-to-2D models. Recent studies have explored several strategies for aligning pretrained text-to-2D models with human preferences: (1) direct fine-tuning with scalar rewards (Clark et al., 2023; Xu et al., 2023; Prabhudesai et al., 2024; Wu et al., 2024c; Shen et al., 2025), (2) Reward Weighted Regression (RWR) (Peng et al., 2019; Lee et al., 2023), (3) Direct Preference Optimization (DPO) (Rafailov et al., 2023; Yang et al., 2024a), and (4) PPO-based policy gradients (Black et al., 2024; Fan et al., 2023; Liu et al., 2025). In this work, we adopt *direct fine-tuning*, which uses gradient-based feedback to align the generative model with the stitched decoder, ensuring that the resulting latents yield high-quality, 3D-consistent outputs.

B Methodology Details and Its Implementation

In this section, we provide additional details about the methodology behind VIST3A, extending the description given in Section 3. We first elaborate on the architectural and training aspects of our stitching method in Section B.1, including the stitching layers and loss functions used for MV-DUSt3R (Tang et al., 2025), VGGT (Wang et al., 2025a), and AnySplat (Jiang et al., 2025). Section B.2 then details the direct reward finetuning methodology, outlining the reward formulations and their implementation for each 3D model (VGGT, AnySplat, and MVDUSt3R).

B.1 Model Stitching

Stitching layer. We implement the stitching layer $\mathbf{S}$ as a single Conv3D layer. Relying only on Conv3D parameters to align the spatial and temporal dimensions between the latent and the features from $F_{k+1:l}$ can result in unnatural configurations, such as excessively large padding size. To address this, we first interpolate the latent representation to the target dimensions and then apply Conv3D, which provides a cleaner alignment of spatial and temporal dimensions. This design still admits a closed-form expression of the stitching layer, as shown in Eq. 2.

Loss function for each 3D model. We train the stitched VAE using an ℓ_1 loss between its outputs and those of the original 3D model. Since 3D model outputs often consist of multiple components, we compute the ℓ_1 loss for each component separately and then aggregate them with a weighted sum. Assigning equal weights can destabilize training and even cause divergence, since some components (e.g., confidence terms) have much larger scales than others. To mitigate this, we reweight the component losses to approximately balance their scales. The specific weighting strategy is adapted to each 3D model as follows:

- **MVDUSt3R.** The outputs consist of pointmaps, confidence scores for the pointmaps, and 3D Gaussian primitives We assign a weight of 10^{-2} to the confidence term, while pointmap and Gaussian primitive losses are left unscaled.
- **VGGT.** Outputs include pointmaps, depth maps, camera poses, and confidence for both pointmaps and depth. In addition, following VGGT's practice, we add gradient-based regularization losses on pointmaps and depth maps. We weight the pose loss by 5, all confidence terms by 5×10^{-3}, and gradient regularization losses by 5×10^{-3}. Other losses remain unscaled.
- **AnySplat.** Outputs include depth maps, Gaussian primitives, confidence for both depth and Gaussian primitives, camera poses, and anchor features. Additionally, we introduce gradient-based regularization losses on the depth maps. We weight all confidence terms by 10^{-2}, gradient regularization losses by 5×10^{-3}, Gaussian scale parameters by 10, and anchor features by 0.1. Depth and other Gaussian parameters are left unscaled.

Hyperparameters and implementation details. For the stitching layer $\mathbf{S}$, we adopt a single 3D convolution with kernel size, stride, and padding chosen to align the latent features from the video VAE with the representation space of each 3D model:

- **MVDUSt3R:** a 3D convolution with kernel size $5 \times 7 \times 7$, output channels 1024, stride $1 \times 3 \times 3$, and padding $2 \times 0 \times 0$.
- **VGGT:** a 3D convolution with kernel size $5 \times 3 \times 3$, output channels 1024, stride $1 \times 2 \times 2$, and padding $2 \times 1 \times 1$.
- **AnySplat:** a 3D convolution with kernel size $5 \times 3 \times 3$, output channels 1024, stride $1 \times 2 \times 2$, and padding $2 \times 1 \times 1$.

Before applying the convolution, the interpolation layer recovers the temporal dimension compressed by the video VAE and adjusts the spatial size so that it matches the resolution expected by the feedforward 3D model. The input resolution of the video VAE is set to 384×384 for MVDUSt3R and 512×512 for both AnySplat and VGGT, as these configurations empirically yield stable training for the respective generative backbones. We employ LoRA with rank $r = 64$ and scaling factor $a = 32$ to Conv2D and linear layers across all cases.

B.2 Direct Reward Finetuning

Reward details. We combine CLIP-based scores and HPSv2.1 human preference scores to construct rewards for both multi-view image quality and 3D representation quality. Specifically, we use DFN (Fang et al., 2024) as the CLIP model and HPSv2.1 (Wu et al., 2023). Given an image I and its associated prompt c, we denote the HPSv2.1 score as s_{hps} and the DFN CLIP score as s_{clip}. The quality reward is then defined as

$$R_{\text{quality}}(I, c) = s_{\text{clip}}(I, c) + s_{\text{hps}}(I, c) - 2, \tag{5}$$

which implies that maximizing the reward is equivalent to maximizing the underlying score.

For the **multi-view image quality reward**, we compute the scores using the multi-view images decoded from the video decoder and their corresponding prompts. For the **3D representation quality reward**, we compute the scores using the rendered images obtained from the 3D representation reconstructed by the stitched decoder, together with the same prompts.

The 3D consistency reward is computed as a combination of the pixel-level ℓ_1 loss and the LPIPS between a decoded multi-view image and its corresponding rendering from the reconstructed 3D representation. Formally, given a decoded image I_{decode} and the estimated camera pose $\hat{\pi}$ predicted by the stitched decoder, we obtain the rendered image $I_{\text{rendered}}(\hat{\pi})$ from the 3D representation. The consistency reward is then defined as

$$R_{\text{consistency}}(I_{\text{decode}}, I_{\text{rendered}}(\hat{\pi})) = -|I_{\text{decode}} - I_{\text{rendered}}(\hat{\pi})|_1 - 0.25 \times \text{LPIPS}\left(I_{\text{decode}}, I_{\text{rendered}}(\hat{\pi})\right). \tag{6}$$

Here, the negative sign ensures that maximizing the reward corresponds to minimizing both the ℓ_1 distance and the perceptual discrepancy between the decoded and rendered images.

However, applying these rewards to all decoded multi-view images and their rendered counterparts is computationally expensive. To reduce computational cost, we compute all rewards only on two sampled decoded views and their corresponding rendered images. The final reward is then obtained by a weighted combination of the three components: the multi-view image quality reward and the 3D representation quality reward are each scaled by $1/16$, while the 3D consistency reward is scaled by 0.05. These scaled terms are summed to form the overall training reward.

Alignment Algorithm. For alignment, we adopt DRTune (Wu et al., 2024c)-style direct reward finetuning, which enables stable reward optimization through selective gradient computation. We outline one training iteration of our finetuning in Algorithm 1. First, we calculate the generative loss using multi-view datasets, then simulate the denoising process. Since matching the full number of inference-time denoising steps during training is costly, we instead sample t steps from a reduced range $[T_1, T_2]$ to lower the computational burden. Additionally, to reduce time and memory costs, we only enable gradient computation at K selected training steps t_{train} out of the total t steps. Following DRTune, the input z_τ to the generative model is detached at each step to stabilize optimization. Finally, we calculate the reward from the sampled latent and combine it with the generative loss by subtraction (for maximization) before backpropagation and parameter updates.

Algorithm 1 One Training Iteration of Alignment Training

```
Input: generative model $\theta$, reward $r$, sampling step range $[T_1, T_2]$, # of gradient enabled steps $K$, prompt $c$, data $D$.
$L_{\text{gen}} \leftarrow$ calculate generative loss with $D$
$t \sim \text{Uniform}(T_1, T_2)$   ▷ Sample number of denoising steps
$z_T \sim \mathcal{N}(0, I)$   ▷ Initialize starting noise
Define $t$-step schedule $\{\tau_j\}_{j=0}^{t}$ with $\tau_0 = T, \tau_t = 0$
$t_{\text{train}} \leftarrow$ randomly select $K$ indices from $\{1, \ldots, t\}$
for $j = 1$ to $t$ do   ▷ Denoising from $T$ to 0
    $\hat{z}_{\tau_j} \leftarrow \texttt{stop\_grad}(z_{\tau_j})$
    if $j \in t_{\text{train}}$ then
        prediction $\leftarrow$ model$(\theta, \hat{z}_{\tau_j}, \tau_j)$
    else
        no_grad: prediction $\leftarrow$ model$(\theta, \hat{z}_{\tau_j}, \tau_j)$
    $z_{\tau_{j+1}} \leftarrow$ update$(z_{\tau_{j-1}}$, prediction$)$
$r(z_0, c) \leftarrow$ Calculate reward of generated latent.
$L_{\text{total}} \leftarrow L_{\text{gen}} - r(z_0, c)$
Backpropagate $\nabla_\theta L_{\text{total}}$, then optimize $\theta$
```

Hyperparameter in sampling process. For generating samples required in the $[T_1, T_2]$ direct reward tuning stage, we set $T_1 = 10$ and $T_2 = 50$ in Algorithm 1, ensuring that the number of diffusion steps is smaller than the typical steps in inference. The number of gradient-enabled steps is set to $K = 2$ to reduce memory consumption during training. For scheduling, we adopt the default scheduler from Wan 2.1 (Wan et al., 2025).

C Details on Experimental Setups

C.1 Training Setup

Setup for stitching layer search. To identify the stitching layer, we rely on representations from the feedforward 3D model and the corresponding latents computed on the same dataset. Specifically, we utilize a subset of the DL3DV dataset, comprising 200 scenes for VGGT, 800 scenes for AnySplat, and 3,200 scenes for MVDUSt3R, with only 13 views per scene used for the search. We limit our search to the encoder layers of each model, as we observe that MSE values consistently increase within deeper layer indices.

Setup for stitched VAE finetuning. We train on a combination of the DL3DV and ScanNet datasets, defining one epoch as a full pass over DL3DV and two passes over ScanNet. For each training iteration, a number of scenes are sampled according to the batch size. From each selected scene, we randomly sample 9 or 13 views to serve as input samples for training. The models are trained for 50 epochs in total. The batch sizes are set to 12 for VGGT, 24 for MVDUSt3R, and 12 for AnySplat. The learning rate is fixed at 2×10^{-4} for all models with cosine decay scheduling and 500-step warmup. For training, we use AdamW (Loshchilov & Hutter, 2017), apply gradient clipping with a norm threshold of 1.0, and use gradient checkpointing on each stitched VAE block to reduce memory consumption. In addition to LoRA parameters, for AnySplat and VGGT, we also finetune register tokens and class tokens. This is necessary because we remove the earlier layers that originally process these tokens into intermediate representations, requiring adaptation of the token handling mechanism. We further utilize gradient checkpointing for every stitched VAE block.

Setup for generative model finetuning. We finetune the generative models using only the DL3DV dataset. For generative loss computation, we use a batch size of 12 with 13 views per scene. Reward calculation uses a prompt batch size of 4, with 13 views for AnySplat and MV-DUSt3R, and 9 views for VGGT. We again adopt AdamW with a learning rate of 1×10^{-4}, apply gradient clipping at a 0.1 norm, and train LoRA parameters with rank 8 and alpha 16. Gradient checkpointing is enabled for all model blocks to reduce memory usage.

C.2 Detailed Evaluation Protocol

Details for 3D generation evaluation. For T3Bench, we evaluate on all 300 prompts, in contrast to prior works that considered only the 100 single-object-with-surroundings subset. SceneBench is evaluated on 80 prompts from the Prometheus3D (Yang et al., 2025c) prompt set, targeting scene-level generation. For DPG-Bench, we sample 100 prompts from the original 1K-prompt dataset.

For Matrix3D-omni, we used their official code for text-to-generation and employed Panorama LRM for reconstruction during inference. For SDS-based methods like SplatFlow and Director3D that perform refinement, we evaluated the final results after SDS optimization. We generate 13 frames for all models using 80 denoising steps, and apply classifier-free guidance (Ho & Salimans, 2022) with a scale of 7.5. We observed that the Gaussian splatting produced by the MVDUSt3R model does not generalize well across diverse domains, often failing to estimate the scale of primitives. To address this issue, we refined the Gaussian primitives using the source view for 100 optimization steps, minimizing a reconstruction loss defined as MSE + 0.05 × LPIPS. For this refinement, we used the Adam optimizer with separate learning rates for each parameter group: 2e-4 for means, 5e-4 for opacity, 5e-4 for scale, 1e-4 for rotation, and 0 for rgbs. This lightweight refinement effectively corrected the scale estimation errors. For our text-to-3DGS evaluation, we render 8 random viewpoints from the generated Gaussian Splatting representations for assessment.

We evaluate our method and baselines across a range of metrics. To measure the semantic similarity between the input prompt and the rendered images of the generated 3DGS, we compute the CLIP score using the clip-vit-base-patch16 model. Additionally, we adopt the VBench (Huang et al., 2024) framework to assess key image properties. For Imaging Quality, which targets low-level distortions, we employ the same MUSIQ model (Ke et al., 2021) in VBench. For Aesthetic Quality, we use the LAION aesthetic predictor to evaluate the color richness and artistic merits, again following VBench. The predictor's native 0-10 rating is linearly normalized to a 0-1 scale for our analysis.

For a more comprehensive assessment of generative quality, we utilize the Unified Reward model (Wang et al., 2025c), which is based on the powerful Qwen 2.5-7B Vision Language Model (Team, 2025)[2]. This provides fine-grained, pointwise scores on complex attributes equipped with a powerful understanding capability. By feeding the input prompt and rendered images into a format adapted from the official implementation script[3], we obtain scores for three key aspects:

- *Alignment*: How well the image content matches the text prompt.
- *Coherence*: The logical and visual consistency of the image, free of distortions.
- *Style*: The aesthetic appeal of the image, independent of prompt accuracy.

This suite of metrics enables a robust and multifaceted evaluation of our model's performance.

Details for model stitching evaluation. For novel-view synthesis, we follow prior works (Go et al., 2025a;b) and adopt an 8-frame input setup to evaluate performance on 4 target views. To accommodate the fixed-length input requirements of video VAE architectures due to temporal compression, we pad shorter sequences by duplicating the final frame. For estimating the camera poses of the target views, we adopt the strategy from AnySplat (Jiang et al., 2025), which jointly predicts

[2] `https://huggingface.co/CodeGoat24/UnifiedReward-qwen-7b`

[3] `https://github.com/CodeGoat24/UnifiedReward/blob/main/inference_qwen/image_generation/qwen_point_score_ACS_image_generation.py`

Table 5: **Ablation study on direct reward finetuning on SceneBench.** We compare (1) no finetuning; (2) multi-view-only finetuning (generative loss only); (3) reward tuning with 3D-consistency reward only; (4) reward tuning with quality reward only; and (5) reward tuning with both rewards (full).

Method	Imaging	Aesthetic	CLIP	Unified Reward		
				Alignment	Coherent	Style
Finetuning-free	50.56	53.70	28.14	3.101	3.354	3.393
Multi-view only	54.56	52.08	29.71	3.622	3.834	3.351
Multi-view + 3D Consistency	38.67	50.59	29.77	3.581	3.767	3.275
Multi-view + Quality	62.27	**58.23**	**30.34**	3.643	3.842	3.358
Ours	**64.87**	56.96	30.18	**3.667**	**3.862**	**3.400**

the poses and renders the corresponding images. This contrasts with previous VAE-based methods that presume access to ground-truth camera poses for rendering.

For pointmap and camera pose estimation evaluation, we use a 13-frame input setup. Since our stitched VAE's encoder is a video VAE, we arrange the multi-view images (typically provided unordered by previous works) into sequences with smooth view transitions to resemble video input. We adopt Pi3 (Wang et al., 2025d) official evaluation code and follow their preprocessing pipeline.

C.3 Selection Criterion for 3D models

In this section, we elaborate on the criteria used for selecting 3D foundation models in VIST3A.

Our primary criterion for selecting a 3D backbone is the scale of the pretraining dataset, which mainly determines the generalizability of the model. This is because the coupled 3D model should cover highly diverse domains of video generative models. AnySplat (Jiang et al., 2025) is selected as it represents one of the few Gaussian Splatting models trained on such a scale. For pointmap-based models, we utilize MVDUSt3R (Wang et al., 2024b) for its balance of dataset scale and efficiency (facilitating feasibility checks), and VGGT (Wang et al., 2025a) for its superior performance and pretraining scale.

D Further Ablation Studies

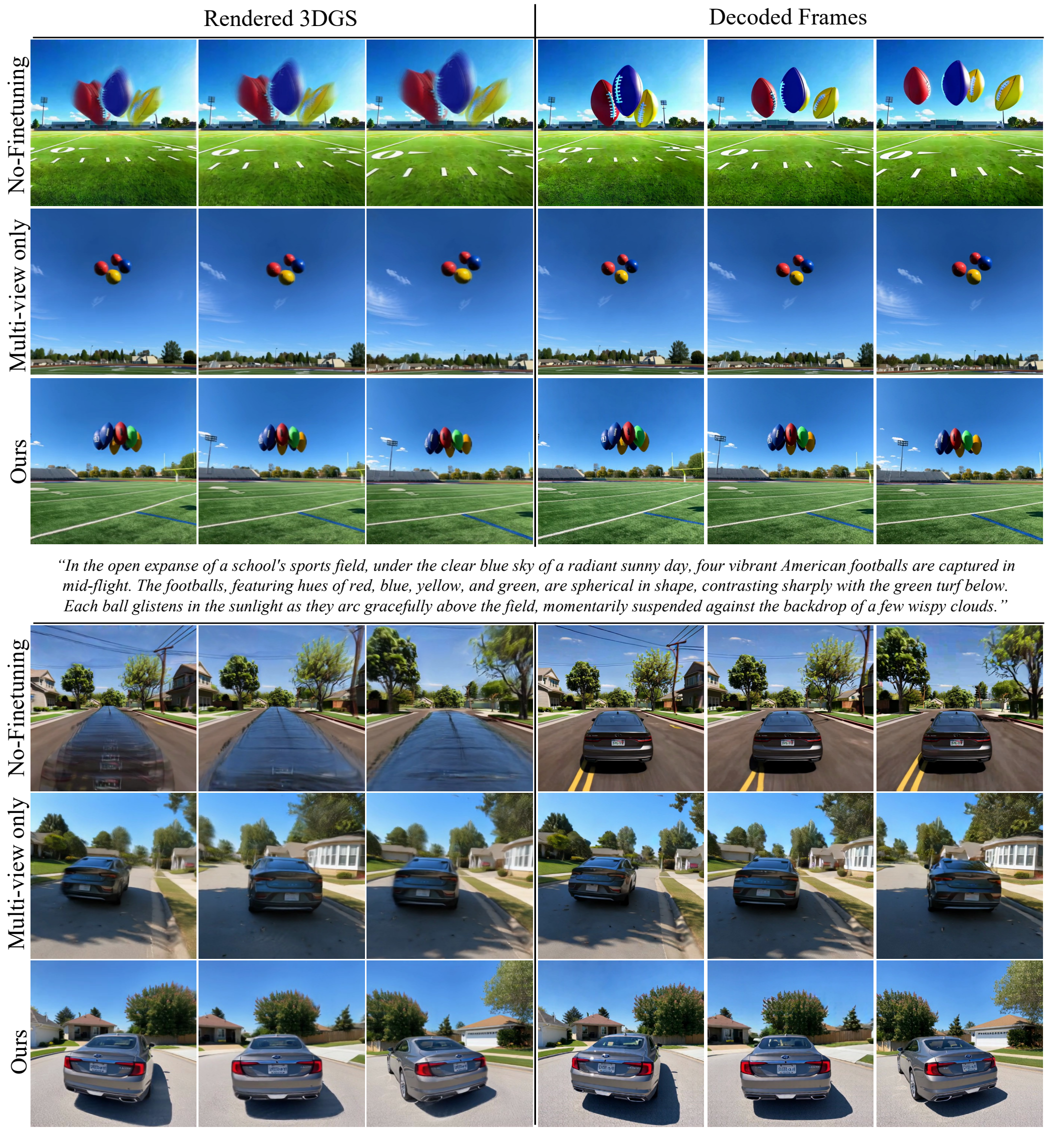
D.1 Impact of Direct Reward Finetuning

In the following, we conduct an ablation study to analyze the effects of our direct reward finetuning, comparing our full method against four well-defined baselines:

- (1) Finetuning-free: Here, we use the original pretrained video model. Since our finetuning freezes the encoder, its latent space remains compatible with our 3D stitched decoder.
- (2) Multi-view Only: The model finetuned with only the flow-matching loss on multi-view data, serving as our primary baseline before rewards are introduced.
- (3) Multi-view + Consistency: The model finetuned with both the multi-view loss and the 3D-consistency reward. This isolates the impact on the 3D consistency reward.
- (4) Multi-view + Quality: The model finetuned with both the multi-view loss and the quality reward. This isolates its impact on quality reward.

To ensure a fair comparison against reward-based methods, which often take more time for one training iteration, the finetuning variant on multi-view data was trained for the same wall-clock duration.

Table 5 reports the quantitative results. The finetuning-free baseline yields the lowest performance. Lacking any 3D-aware training, it frequently produces geometrically inconsistent outputs and suffers



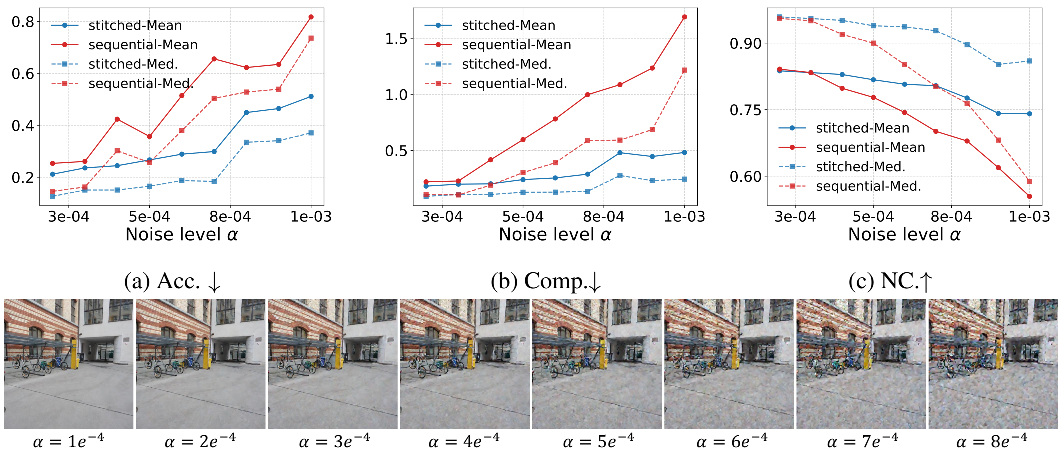
"In the open expanse of a school's sports field, under the clear blue sky of a radiant sunny day, four vibrant American footballs are captured in mid-flight. The footballs, featuring hues of red, blue, yellow, and green, are spherical in shape, contrasting sharply with the green turf below. Each ball glistens in the sunlight as they arc gracefully above the field, momentarily suspended against the backdrop of a few wispy clouds."

A car parked on a street, its rear end facing away from the viewer against a house and trees backdrop.

Figure 7: **Qualitative comparison of different finetuning strategies.** Pretrained video model (No-Finetuning) produces dynamic videos, causing severe ghosting in the reconstructed 3D scenes. Multi-view finetuning (Multi-view only) reduces this motion and improves 3D consistency but introduces semantic and quality degradations, while our direct reward tuning (Ours) yields sharper renderings that better align with the input text prompts.

from significant visual artifacts when its native resolution is adapted to our 3D decoder. Introducing multi-view supervision (Multi-view Only) substantially improves 3D consistency and overall performance, confirming the value of this training signal.

The reward components have distinct effects when added to the multi-view objective. Training with the 3D-consistency reward (Multi-view + Consistency) leads to a notable performance drop, as the model optimizes for geometric correctness at the expense of detail, resulting in overly blurred images. Conversely, adding the quality reward (Multi-view + Quality) achieves substantial improvements across most metrics by enhancing prompt coherence and aesthetic appeal.



(a) Acc. ↓ (b) Comp.↓ (c) NC.↑

(d) Reconstructed images through VAE according to α

Figure 8: **Pointmap estimation performance comparison on ETH3D dataset between the stitched VGGT and the sequential approach (VAE followed by VGGT) under varying noise scales injected into the latent space.** The stitched model demonstrates greater robustness to noise injection in the VAE.

Finally, our full method, which combines both rewards with multi-view training, achieves the best imaging quality and Unified Reward scores. While its aesthetic and CLIP scores are slightly below the Multi-view + Quality variant, the marked improvement in imaging quality demonstrates that our combined objective successfully guides the model to generate visually sharp and geometrically faithful 3D representations.

To further analyze the effect of our training strategy, we provide a qualitative comparison between the pretrained video model, the multi-view-only finetuning baseline, and our reward-tuned model in Fig. 7. As shown in the decoded frames, the **No-Finetuning** baseline (pretrained video generator) often produces dynamic videos with temporal motion. This leads to severe ghosting artifacts in the generated 3D scenes, clearly visible as multiple outlines of the footballs and the car.

The **Multi-view only** baseline (second row) effectively suppresses this motion, enforcing 3D consistency. However, relying solely on a multi-view dataset limits the model's generalizability, and the lack of alignment with the decoder hinders the model from generating latents that are well reconstructible by the decoder. As a result, the generations exhibit semantic and quality degradations: the "green" football is missing in the first example, and the car becomes blurrier in the second example.

In contrast, our direct reward tuning (**Ours**) applies rewards to the decoded 3D representation of the stitched decoder, explicitly encouraging high reconstruction quality, 3D consistency, and better alignment with the text prompt. Consequently, it produces sharper and more faithful renderings than the baselines in both examples.

D.2 Benefits of integrated vs. sequential 3D generation

In our model-stitching design, generation and reconstruction take place in the shared latent space of the video diffusion VAE and the stitched 3D decoder. A common alternative is a sequential pipeline that decodes latents into RGB frames before applying a feedforward 3D model (e.g., VGGT) without further adaptation. To probe the core benefit of our unified formulation, we injected controlled perturbations into the latent representation, using

$$z' = z + \alpha \,\|z\|\, \epsilon, \quad \epsilon \sim \mathcal{N}(0, I), \tag{7}$$

where α is a scalar controlling the perturbation strength. We then compared two paths: (i) decode the corrupted latent to RGB and feed the images sequentially into the original VGGT (baseline), and (ii) directly input the noised latent into our stitched 3D decoder (unified latent framework).

Table 6: **User study results.** Participants rank five methods in terms of text alignment and visual quality of rendered videos from generated 3DGS (lower average rank is better).

Method	Text Alignment (Avg. Rank ↓)	Visual Quality (Avg. Rank ↓)
Director3D	3.03	2.99
SplatFlow	3.38	3.88
Prometheus3D	3.25	3.71
VideoRFSplat	2.74	2.92
VIST3A (Wan+ AnySplat)	**1.54**	**1.45**

Figure 8 reports pointmap estimation performance on ETH3D as a function of noise level α. Our stitched VGGT consistently outperforms the sequential decode-and-reconstruct pipeline under noise injection, indicating that the VAE decoder in the sequential path amplifies errors. Moreover, as shown in Fig. 8d, the performance gap is observed even at noise levels ($\alpha = 1e^{-4}$ to $2e^{-4}$) where visual artifacts are hardly perceptible. This suggests that the unified design offers stronger robustness, as imperceptible perturbations from the noise of generative processes can already degrade the sequential pipeline.

D.3 Human evaluation

To further support the effectiveness of our method, we conduct a user study comparing VIST3A against four baselines (Director3D, SplatFlow, Prometheus3D, and VideoRFSplat). We ask participants to rank the methods based on two criteria: (1) Text Alignment and (2) Visual Quality of rendered videos from the generated 3DGS. A total of 28 participants evaluate 14 randomly selected samples from T3Bench, SceneBench, and DPG-Bench.

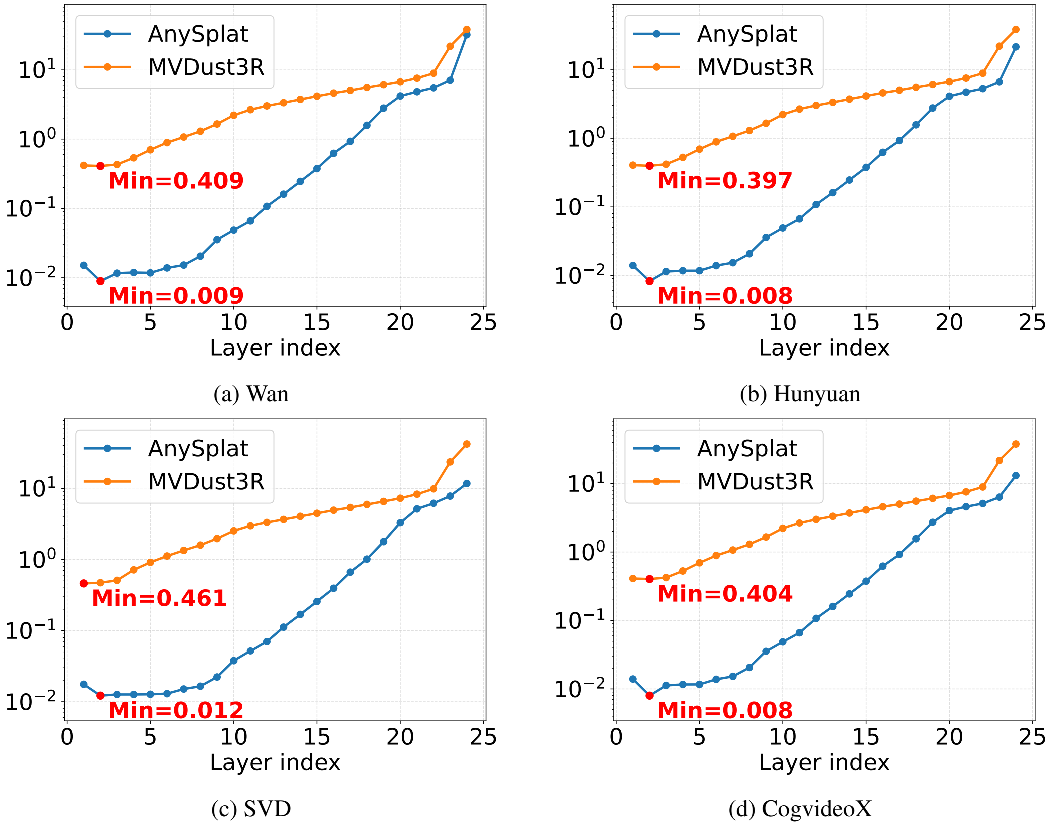
As summarized in Table 6, VIST3A achieves the best performance (lowest average rank) on both criteria. Notably, participants rank VIST3A as the top-1 method in 68.68% of cases for text alignment and 87.91% for visual quality, demonstrating its superiority in generating high-fidelity and semantically consistent 3D scenes.

E Additional Results

To comprehensively validate each component of **VIST3A**, we present additional experiments in this section.

Analysis on searched stitching index. In Section 4.4, we showed that earlier layers in the network tend to be more linearly correspondent. We extend this analysis to various VAE architectures, including CogVideoX, SVD, Hunyuan, and Wan, paired with MVDUSt3R and AnySplat, to observe the generalizability of this finding.

Figure 9 shows the log-MSE values measuring linear transferability between latents and the feedforward 3D model's representations. From the results, early layers of 3d models consistently show lower MSE values across all VAE-feedforward 3D model combinations. This supports the hypothesis that latent representations capture low-level features for input reconstruction, which are more linearly transferable to the early layers of the feedforward 3D model that also encode such features. However, the results reveal an important distinction: while relative MSE ordering within each VAE architecture correlates with stitching performance (as in Section 4.4), absolute MSE values across different VAEs do not predict cross-architecture performance. For instance, CogVideoX + AnySplat achieves the lowest absolute MSE (0.008) but delivers 21.32 PSNR in Table 3, while SVD + AnySplat with a higher MSE (0.012) achieves superior performance at 21.48 PSNR. This indicates that optimal stitching layers must be identified independently for each VAE-3D model pair.



(a) Wan

(b) Hunyuan

(c) SVD

(d) CogvideoX

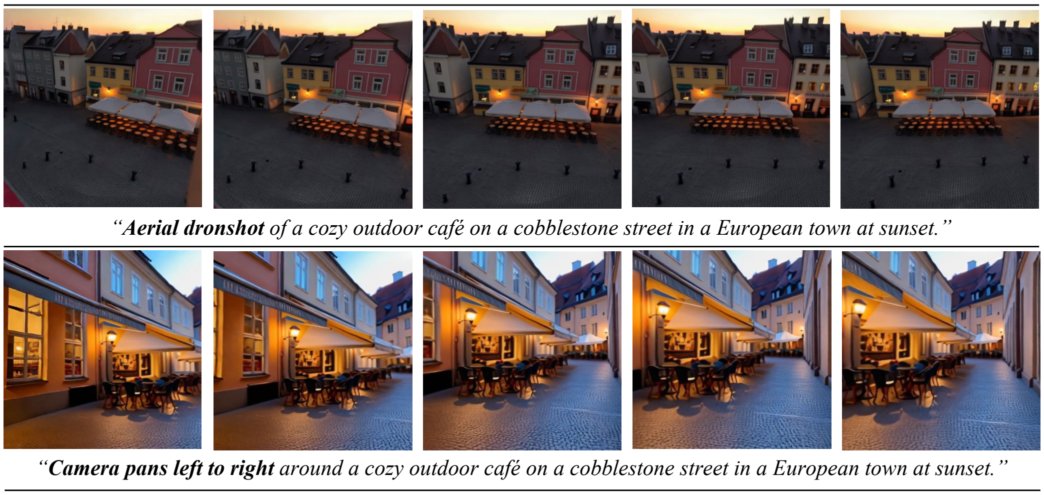
Figure 9: **Log-MSE values in Eq. 2 across various video VAEs.** Early layers of feedforward models show lower MSE values within each VAE architecture. While lower MSE correlates with better stitching performance within the same VAE (e.g., layer 2 outperforms layer 16 for Wan in Fig 5), absolute MSE values cannot predict performance across different VAE architectures. For instance, despite CogVideoX and Hunyuan + AnySplat having the lowest absolute MSE (0.008), SVD + AnySplat achieves the best performance (21.48 PSNR) in Table 3.

Table 7: **Video generation performance comparison between the original video generator and VIST3A on VBench.**

Model	**Quality Score**	**Semantic Score**	**Total Score**
Original video generator	0.7827	**0.7275**	0.7716
VIST3A (Wan + AnySplat)	**0.8143**	0.7143	**0.7943**

Prompt-based camera control. Modern video generative model, including Wan 2.1, can reflect camera-related prompts such as “aerial view” and “camera pans left to right” in the generated videos. As our framework is built upon the Wan 2.1 backbone, we inherit this property, enabling text-driven viewpoint control in 3D generation. As illustrated in Fig. 10, the prompt containing “Aerial dronshot” induces a high-angle, downward-looking perspective (Top), whereas the prompt with “Camera pans left to right” results in a horizontal sweeping motion that traverses the scene from left to right (Bottom). This shows that our model effectively inherits the semantic camera instruction understanding capability of pretrained video generative models.

Impact of VIST3A finetuning on video generation performance. We aim for video generative models to produce 3D-consistent frames, and we propose a finetuning strategy toward this goal. Here, we evaluate how much this finetuning degrades the original model’s generative capability. To measure this, we use VBench (Huang et al., 2024), a well-established benchmark for video generation. We generate videos using the original model by matching the resolution to the VIST3A



*“**Aerial dronshot** of a cozy outdoor café on a cobblestone street in a European town at sunset.”*

*“**Camera pans left to right** around a cozy outdoor café on a cobblestone street in a European town at sunset.”*

Figure 10: **Viewpoint control in 3DGS generation with prompts.** **(Top)** The prompt containing “Aerial dronshot” results in a high-angle downward perspective. **(Bottom)** The prompt containing “Camera pans left to right” generates a trajectory where the viewpoint sweeps across the scene from left to right.

inference setting, and compare the video results of the finetuned model with VIST3A with Wan and AnySplat.

As shown in Table 7, the VIST3A-adapted model achieves a higher quality score and a higher total score than the original video generator, with only a marginal difference in the Semantic Score. This demonstrates that VIST3A’s finetuning does not significantly degrade the video model’s generative capability—in fact, the overall VBench performance is slightly improved under the VIST3A setup.

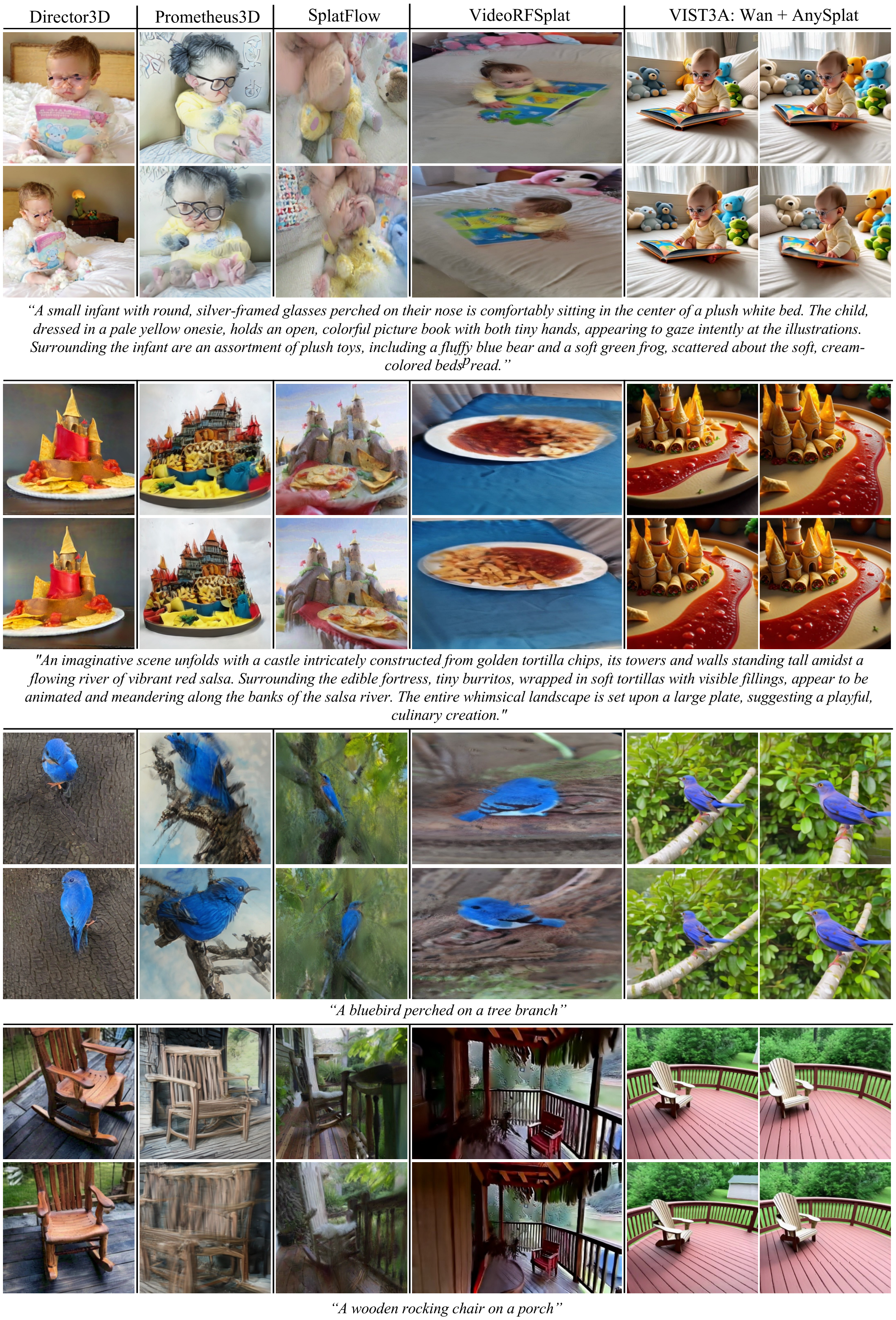
Additional qualitative results. We present additional qualitative results of VIST3A with Wan + AnySplat in Fig. 11–13. Text-to-pointmap generation results obtained by combining VGGT with Wan through VIST3A are shown in Fig. 14. Finally, Fig. 15 illustrates VIST3A results with MV-Dust3R + Wan.

F LIMITATIONS

While our approach demonstrates strong results, it also has certain limitations. Our stitched model inherits its encoder from a video generation model, which is inherently designed for sequential, temporally coherent video input. Consequently, its performance is not guaranteed for arbitrarily unordered inputs, such as typical multi-view image datasets. To ensure the encoder operates effectively, the input images must be arranged into a coherent sequence that simulates the smooth view transitions of a video clip.

G USE OF LARGE LANGUAGE MODELS

LLMs were used exclusively for text polishing and grammar refinement.



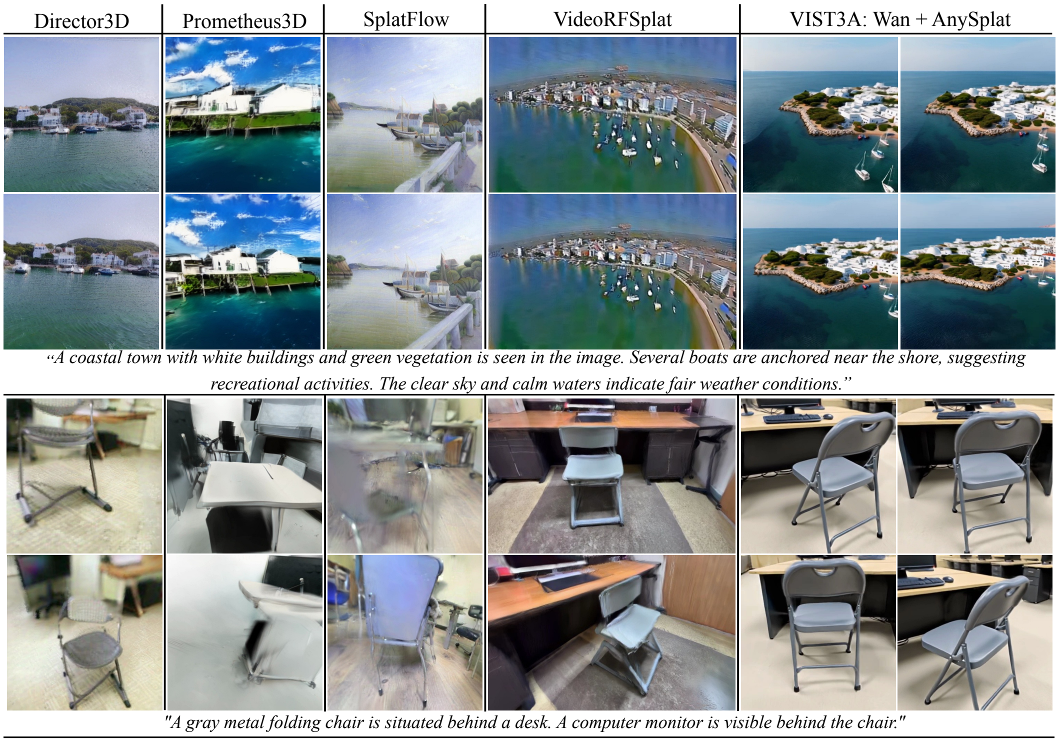
“A small infant with round, silver-framed glasses perched on their nose is comfortably sitting in the center of a plush white bed. The child, dressed in a pale yellow onesie, holds an open, colorful picture book with both tiny hands, appearing to gaze intently at the illustrations. Surrounding the infant are an assortment of plush toys, including a fluffy blue bear and a soft green frog, scattered about the soft, cream-colored bedspread.”

"An imaginative scene unfolds with a castle intricately constructed from golden tortilla chips, its towers and walls standing tall amidst a flowing river of vibrant red salsa. Surrounding the edible fortress, tiny burritos, wrapped in soft tortillas with visible fillings, appear to be animated and meandering along the banks of the salsa river. The entire whimsical landscape is set upon a large plate, suggesting a playful, culinary creation."

“A bluebird perched on a tree branch”

“A wooden rocking chair on a porch”

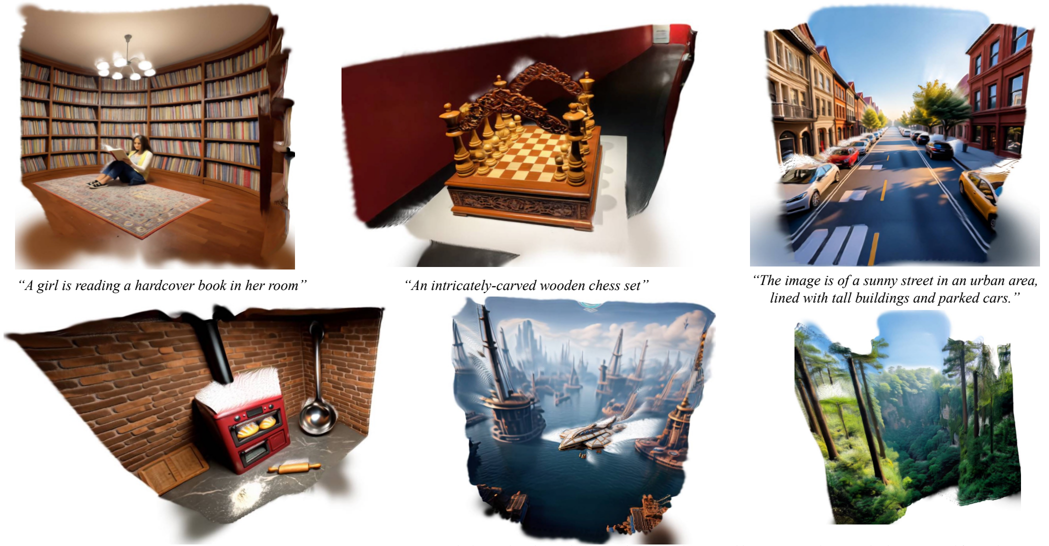
Figure 11: **Qualitative comparison of 3DGS generation.** The top two rows show samples from DPG-Bench, and the bottom two rows present samples from T3Bench. VIST3A generates realistic scenes with fine-grained details that faithfully reflect the input prompt, outperforming baselines.



“A coastal town with white buildings and green vegetation is seen in the image. Several boats are anchored near the shore, suggesting recreational activities. The clear sky and calm waters indicate fair weather conditions.”

"A gray metal folding chair is situated behind a desk. A computer monitor is visible behind the chair."

Figure 12: **Qualitative comparison of 3DGS generation on SceneBench.** VIST3A outperforms baselines by generating higher-fidelity scenes with accurate geometry and appearance.



“A girl is reading a hardcover book in her room”

“An intricately-carved wooden chess set”

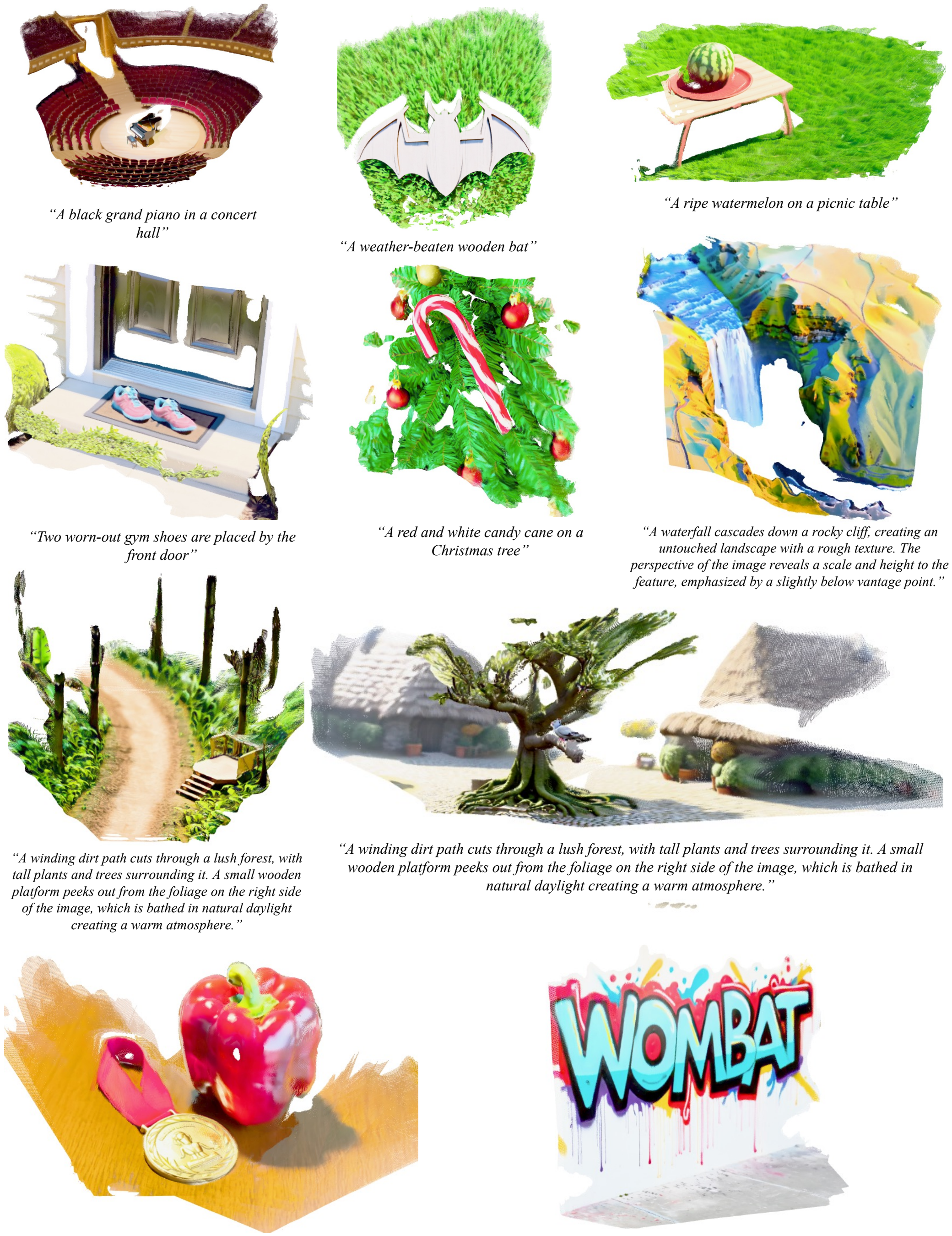
“The image is of a sunny street in an urban area, lined with tall buildings and parked cars.”

“An aged crimson oven occupies the corner of a rustic kitchen, its window revealing the golden-brown crust of bread as it bakes within. Next to it, a towering, polished metallic spoon leans against a weathered brick wall, reflecting the soft kitchen light. Scattered nearby are a scattering of flour and a wooden rolling pin on a worn, marble countertop.”

An intricately designed airship, with sleek steel panels and ornate golden trims, hovers gracefully above a bustling port. The city skyline, a fantastical fusion of floating islands and elevated platforms, echoes the artistic vision of Ivan Shishkin's creations on ArtStation, reminiscent of the game Bioshock Infinite. Captured with the depth of field effect of a 35mm lens, the image exudes a cinematic quality, with the airship's cables and anchors creating a stark contrast against the backdrop of the sky-high metropolis.

“The image displays a tranquil forested canyon with towering trees, surrounded by dense vegetation. The sky is clear and blue, indicating calm weather conditions. Sunlight filters through the canopy, casting dappled shadows on the ground below. Despite the lush foliage, ...”

Figure 13: **Generated 3D scenes from VIST3A: Wan + AnySplat.** These are 3DGS viewed directly in the interactive viewer. VIST3A preserves high visual quality even under noticeably altered camera trajectories, demonstrating robustness and stability across novel viewpoints.



"A black grand piano in a concert hall"

"A weather-beaten wooden bat"

"A ripe watermelon on a picnic table"

"Two worn-out gym shoes are placed by the front door"

"A red and white candy cane on a Christmas tree"

"A waterfall cascades down a rocky cliff, creating an untouched landscape with a rough texture. The perspective of the image reveals a scale and height to the feature, emphasized by a slightly below vantage point."

"A winding dirt path cuts through a lush forest, with tall plants and trees surrounding it. A small wooden platform peeks out from the foliage on the right side of the image, which is bathed in natural daylight creating a warm atmosphere."

"A winding dirt path cuts through a lush forest, with tall plants and trees surrounding it. A small wooden platform peeks out from the foliage on the right side of the image, which is bathed in natural daylight creating a warm atmosphere."

"A dining room setting showcasing an unusually large red bell pepper with a shiny, slightly wrinkled texture, prominently placed beside a diminutive golden medal with a red ribbon on a polished wooden dining table. The pepper's vibrant hue contrasts with the medal's gleaming surface. The scene is composed in natural light, highlighting the intricate details of the pepper's surface and the reflective quality of the medal."

A vibrant graffiti artwork displaying the word "WOMBAT" in bold, multicolored letters, each character outlined in black to create a striking contrast against the stark white wall. The letters are embellished with various shades of blue, green, red, and yellow, with dramatic splashes of paint scattered around the composition. The texture of the dripping paint adds a dynamic and tactile quality to the mural.

Figure 14: **Qualitative results on text-to-poinmap generation.** By integrating VGGT, VIST3A generates structurally consistent pointmaps and fine-grained details across diverse prompts.

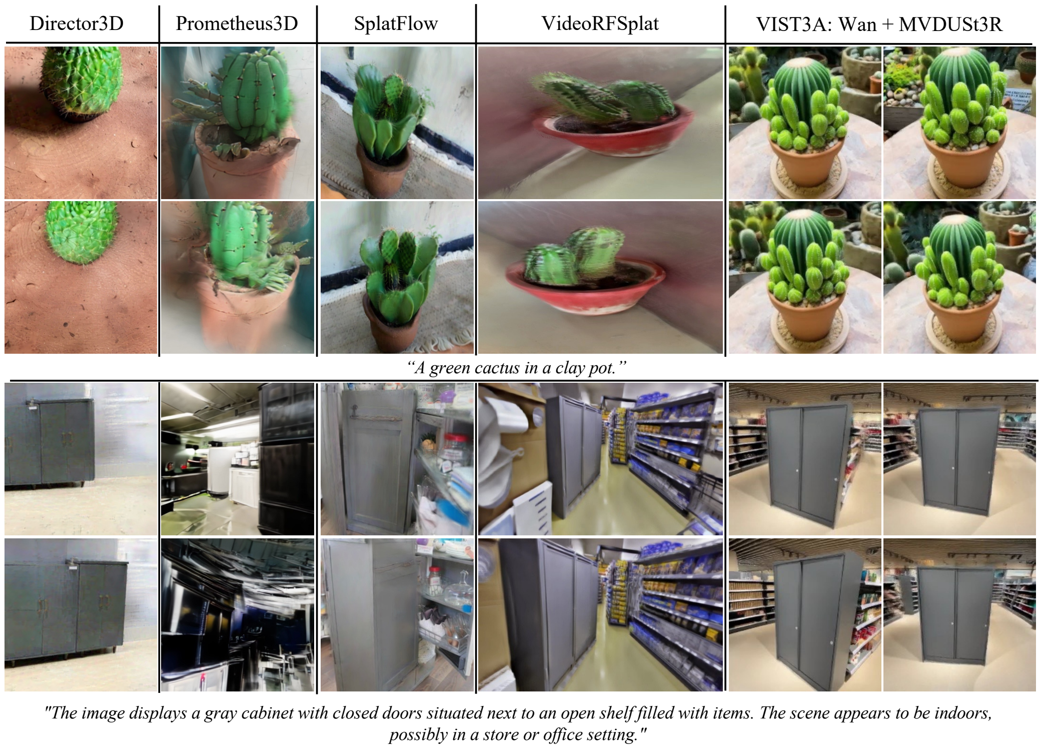

Director3D
Prometheus3D
SplatFlow
VideoRFSplat
VIST3A: Wan + MVDUSt3R
"A green cactus in a clay pot."
"The image displays a gray cabinet with closed doors situated next to an open shelf filled with items. The scene appears to be indoors, possibly in a store or office setting."

Figure 15: **Qualitative comparison of 3DGS generation on SceneBench - VISTA: Wan+MVDUSt3R.**

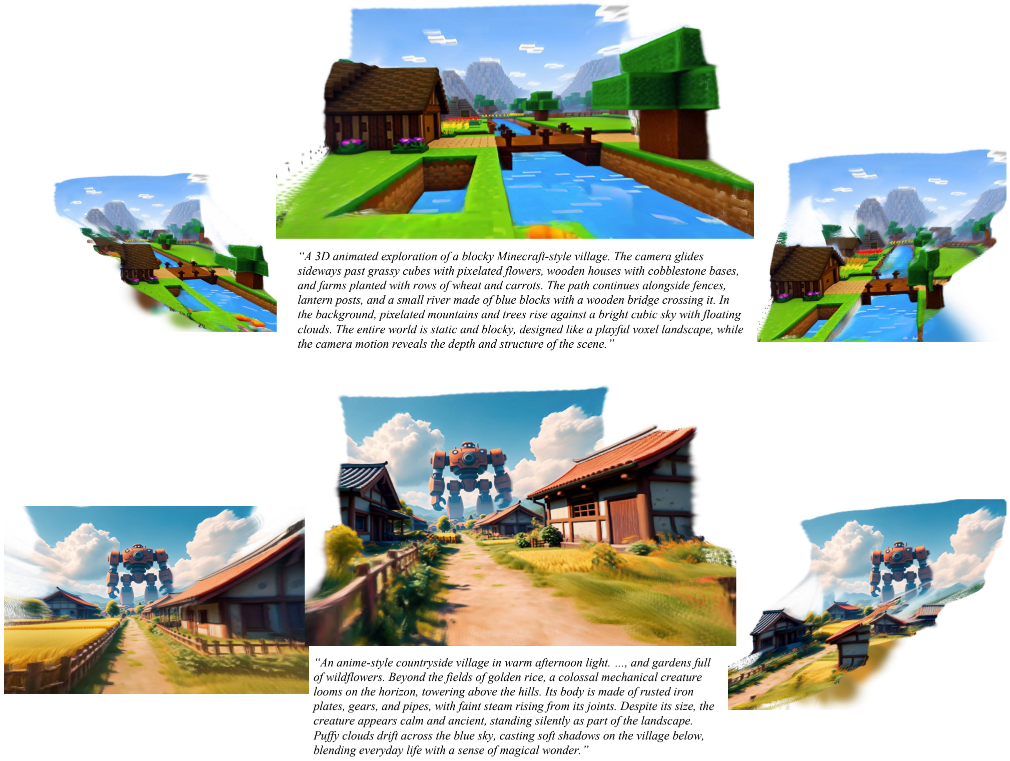

"A 3D animated exploration of a blocky Minecraft-style village. The camera glides sideways past grassy cubes with pixelated flowers, wooden houses with cobblestone bases, and farms planted with rows of wheat and carrots. The path continues alongside fences, lantern posts, and a small river made of blue blocks with a wooden bridge crossing it. In the background, pixelated mountains and trees rise against a bright cubic sky with floating clouds. The entire world is static and blocky, designed like a playful voxel landscape, while the camera motion reveals the depth and structure of the scene."
"An anime-style countryside village in warm afternoon light. ..., and gardens full of wildflowers. Beyond the fields of golden rice, a colossal mechanical creature looms on the horizon, towering above the hills. Its body is made of rusted iron plates, gears, and pipes, with faint steam rising from its joints. Despite its size, the creature appears calm and ancient, standing silently as part of the landscape. Puffy clouds drift across the blue sky, casting soft shadows on the village below, blending everyday life with a sense of magical wonder."

Figure 16: **Generated 3D scenes from VIST3A: Wan + AnySplat bt extending the number of frames.** These are 3DGS viewed directly in the interactive viewer. VIST3A preserves high visual quality even under noticeably altered camera trajectories, demonstrating robustness and stability across novel viewpoints.