LATENT BRIDGE: FROM VIDEO LATENTS TO 3D GEOMETRY VIA FEATURE ALIGNMENT

Anonymous authors
Paper under double-blind review

ABSTRACT

Video generation and feed-forward 3D reconstruction typically use separate representations, requiring RGB decoding and visual re-encoding to recover geometry from generated latents. We propose Latent Bridge, which connects video generative latent spaces with geometric feature spaces to directly predict 3D geometry from video latents. We develop representation-specific bridges for Wan2.1 and V-RAE, enabling feed-forward prediction of cameras, depth, and 3D point clouds without the intermediate RGB pathway. Each bridge is trained on RGB videos through feature distillation, while the video encoder and geometry model remain frozen. Training requires neither ground-truth geometry nor execution of the downstream geometric aggregator or prediction heads. At inference, the same video latents support both RGB decoding and geometry prediction. The same trained bridge can be reused for image-to-video and text-to-video generation within the same video latent space, without bridge retraining. Experiments on Co3Dv2 and WildRGB-D demonstrate competitive geometric reconstruction quality with improved inference efficiency, highlighting latent alignment as an efficient and reusable connection between video generation and 3D reconstruction.

1 INTRODUCTION

Recent advances in video generation and feed-forward 3D reconstruction have enabled compelling visual synthesis and efficient geometric inference. Video generation models, such as CogVideoX (Yang et al., 2025) and Wan2.1 (Wan et al., 2025), combine spatiotemporally compressed latent representations with diffusion transformers to synthesize high-quality, temporally coherent videos. Meanwhile, feed-forward reconstruction models, such as VGGT (Wang et al., 2025), jointly predict camera parameters, depth maps, and 3D point maps from multi-view images in a single forward pass. Recent work has begun to explore the joint execution of these two tasks: generating videos while simultaneously reconstructing the generated content.

Despite their complementary goals, video generation and feed-forward 3D reconstruction typically rely on different representations. Video generators synthesize content in spatiotemporally compressed latent spaces, whereas reconstruction models often aggregate features from visual foundation models such as DINOv2 (Oquab et al., 2024) to infer scene geometry. This representational gap means that combining the two models typically involves decoding generated latents into RGB frames and re-encoding them into geometric features, introducing additional computation. This raises the question: *Can scene geometry be generated directly without RGB decoding?*

Recent work has begun to integrate video generation with explicit geometry prediction. Gen3R (Huang et al., 2026) compresses intermediate VGGT features into geometric latents, aligns their distribution with video appearance latents, and fine-tunes a video diffusion model to jointly generate both. As shown in Fig. 1(a), joint generation in a shared latent space requires additional training and can reduce efficiency due to the increased number of tokens. FantasyWorld (Dai et al., 2026) introduces a geometric branch into video generation. VGGRPO (An et al., 2026) uses latent-space geometry predictions to construct rewards for video model post-training, while Beyond Pixels (Liu et al., 2026) combines latent alignment, LoRA adaptation, and geometry-supervised fine-tuning of prediction heads for direct latent-to-4D prediction. As shown in Fig. 1(b), these methods all aim to train a 3D/4D decoder module, requiring substantial data and retraining of the decoder, while also raising concerns about generalization.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

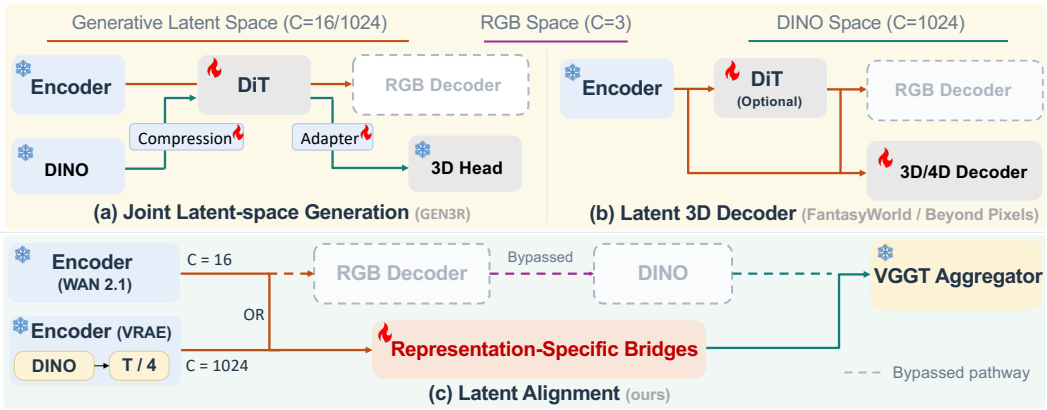


Figure 1: Bypass the video decoder and directly align latent spaces to generate 3D outputs.

Unlike the aforementioned methods, we argue that the key lies in aligning the generative and reconstruction latent spaces. We propose Latent Bridge, which directly maps video latents into the representation space of a frozen feed-forward geometry model (Fig. 1(c)). The frozen downstream VGGT aggregator and depth head can then be used to reconstruct the scene. This provides an elegant solution that eliminates the need for joint feature training or training downstream decoder heads. It also generalizes well, enabling the reuse of existing generative models and downstream tasks built on the same representation. Specifically, we adapt our framework to two representative video encoders, the Wan2.1 VAE and V-RAE (Guo et al., 2026), by designing a dedicated Bridge network for each. This enables models built upon either encoder to directly generate scene geometry.

Our contributions can be summarized as follows:

- We propose Latent Bridge, which aligns the generative latent space with the representation space of a reconstruction model, enabling geometry to be recovered directly from generated latents. The Bridge module is trained through feature distillation and requires only video data for training. This substantially reduces the cost of recovering geometry from the latent spaces of generative models.
- We adapt our framework to two video encoders, Wan2.1 and V-RAE, by designing a dedicated Latent Bridge network for each. Both variants achieve strong performance, demonstrating that the Latent Bridge framework can effectively accommodate different generative latent spaces.
- Experimental results demonstrate that the proposed Latent Bridge enables high-quality and efficient reconstruction. It also exhibits strong generalization: within the same latent space, it can support generative models designed for different generation tasks, as well as other downstream tasks, without retraining.

2 Related Work

Generative latent space. Modern video generation models commonly extend latent diffusion to the spatiotemporal domain, where a video VAE compresses RGB frames into compact latent representations and a diffusion transformer models their temporal dynamics. Stable Video Diffusion (Blattmann et al., 2023) establishes a scalable latent video diffusion framework, while HunyuanVideo (Kong et al., 2024) demonstrates the effectiveness of large diffusion transformers and progressive training for high-fidelity video generation. Wan2.1 (Wan et al., 2025) follows this paradigm and combines a causal spatiotemporal VAE with a diffusion transformer to support both text-to-video and image-to-video generation. REPA (Yu et al., 2025) and VA-VAE (Yao et al., 2025) align diffusion features and VAE latents, respectively, with pretrained visual representations to improve generative modeling. RAE (Zheng et al., 2026) directly adopts frozen visual encoders, while V-RAE (Guo et al., 2026) extends this paradigm to video through temporal pooling.

Feed-forward reconstruction. DUSt3R (Wang et al., 2024) formulates 3D reconstruction as pointmap regression, jointly predicting dense 3D points for an image pair in a shared coordinate frame without requiring known camera parameters. Moving beyond pairwise prediction, VGGT (Wang et al., 2025), built upon a DINOv2-pretrained visual encoder (Oquab et al., 2024), jointly predicts camera parameters, depth maps, 3D point maps, and point tracks from multiple RGB images in a single forward pass, without per-scene optimization. VGGT-Ω (Wang et al., 2026) further initializes its trainable visual backbone from DINOv3 (Siméoni et al., 2026) and introduces scene registers, register attention, and a shared dense prediction head. Together with large-scale self-supervised video training, these designs improve the scalability and efficiency of feed-forward geometry reconstruction for both static and dynamic scenes.

Joint generative-reconstructive method Recent work has explored video and 3D co-generation through complementary approaches. FantasyWorld (Dai et al., 2026) couples video and geometry latents through a trainable geometry branch on a frozen Wan2.1 backbone, while LYRA (Bahmani et al., 2026) introduces a self-distilled 3D Gaussian decoder. Gen3R (Huang et al., 2026) jointly generates video and geometry in a unified latent space, whereas PixWorld (Gao et al., 2026a) operates in pixel space to avoid VAE-induced detail loss. VGGT-World (Sun et al., 2026) instead forecasts geometry-centric VGGT tokens and decodes future geometry using frozen heads, without synthesizing RGB frames. VGGRPO (An et al., 2026) uses latent-space geometry predictions to construct rewards for video DiT post-training, while Beyond Pixels (Liu et al., 2026) directly decodes geometry from video latents. VIST3A (Go et al., 2026) connects video latents to a selected intermediate reconstruction layer and jointly trains the stitching layer and downstream LoRA adapters using teacher-predicted geometry. Pano2World (Li et al., 2026) aligns diffusion hidden features with a pretrained reconstruction model's token space through feature distillation, followed by joint fine-tuning of the adapter and Gaussian decoder under rendering supervision. In contrast, we map video latents to a predefined geometric feature interface through feature alignment, keeping the downstream geometric aggregator and prediction heads entirely frozen.

3 PRELIMINARIES

3.1 VIDEO LATENT REPRESENTATIONS

Video generation models typically use an encoder to compress videos into a latent space. We define an RGB video with T frames, height H, and width W as $\mathbf{X} \in \mathbb{R}^{T\times 3\times H\times W}$. In this paper, we focus on two types of VAEs: the convolutional VAE used in Wan2.1 (Wan et al., 2025) and the representation-based autoencoder V-RAE (Guo et al., 2026).

Wan2.1 VAE. Wan2.1 (Wan et al., 2025) employs a causal spatiotemporal VAE with a spatial compression factor of 8 and 16 latent channels. The first frame is compressed only spatially, while subsequent frames undergo an additional temporal downsampling factor of 4. For $T-1$ divisible by 4, its encoder produces

$$\mathbf{Z}_{\mathrm{W}} = E_{\mathrm{W}}(\mathbf{X}) \in \mathbb{R}^{T_{\mathrm{W}}\times 16\times (H/8)\times (W/8)}, \qquad T_{\mathrm{W}} = 1 + \frac{T-1}{4}, \tag{1}$$

where E_{W} denotes the encoding operation and T_{W} is the latent temporal length. The corresponding decoder reconstructs RGB frames from $\mathbf{Z}_{\mathrm{W}}$.

V-RAE. V-RAE (Guo et al., 2026) instead constructs its latent space using a frozen visual representation encoder. We consider its DINOv3-based variant, which extracts frame-wise features with 1024 channels and a spatial compression factor of 16. Throughout this paper, V-RAE refers specifically to the pretrained weights of this DINOv3-based variant. A learned temporal attention pooler aggregates adjacent feature maps while preserving their channel dimension:

$$\mathbf{Z}_{\mathrm{R}} = P_{\mathrm{R}}(E_{\mathrm{R}}(\mathbf{X})) \in \mathbb{R}^{T_{\mathrm{R}}\times 1024\times (H/16)\times (W/16)}, \tag{2}$$

where E_{R} is the frozen representation encoder, P_{R} is the temporal pooler, and T_{R} denotes the pooled temporal length. With four-frame pooling, each latent step summarizes a four-frame feature window, yielding $T_{\mathrm{R}} = T/4$. A spatiotemporal Transformer decoder maps these representations back to RGB videos. Unlike the low-channel VAE bottleneck, V-RAE retains high-dimensional visual features as its generative representation.

3.2 Feed-Forward Geometry Reconstruction

VGGT (Wang et al., 2025) extracts DINO-based image tokens and aggregates them through alternating frame-wise and global attention. Its prediction heads estimate camera parameters, depth maps, and point maps. VGGT-Ω (Wang et al., 2026) introduces register attention for efficient cross-frame information exchange and simplifies dense prediction to a single depth head, alongside a camera head. For the camera–depth pathway:

$$\{\mathbf{D}_t, \mathbf{K}_t, \mathbf{R}_t, \mathbf{t}_t\}_{t=1}^{T} = \mathcal{H}_{\text{geo}}(\mathcal{A}_{\text{geo}}(E_{\text{img}}(\mathbf{X}))), \tag{3}$$

where E_{img} is the image feature encoder, $\mathcal{A}_{\text{geo}}$ is the multi-frame aggregator, and $\mathcal{H}_{\text{geo}}$ denotes the prediction heads and camera-parameter conversion. Here, $\mathbf{D}_t \in \mathbb{R}^{H\times W}$ is the depth map, $\mathbf{K}_t \in \mathbb{R}^{3\times 3}$ is the intrinsic matrix, and $\mathbf{R}_t \in \mathbb{R}^{3\times 3}$ and $\mathbf{t}_t \in \mathbb{R}^3$ define the world-to-camera rotation and translation. The predicted depth maps and camera parameters can be used to recover 3D point clouds through standard depth unprojection.

4 Method

We propose an alignment framework between generative latents and the representations of a feed-forward geometry model, which we term the **Latent Bridge**. The key idea is to jointly support video generation and 3D reconstruction by aligning generative and reconstruction representations, without retraining the existing vision autoencoder, generative model, or 3D reconstruction heads. For the Wan2.1 VAE and V-RAE, we design distinct Latent Bridge networks and develop a corresponding training procedure, as illustrated in Fig. 2. At inference, the bridge enables the frozen geometry model to predict 3D geometry directly from generated video latents.

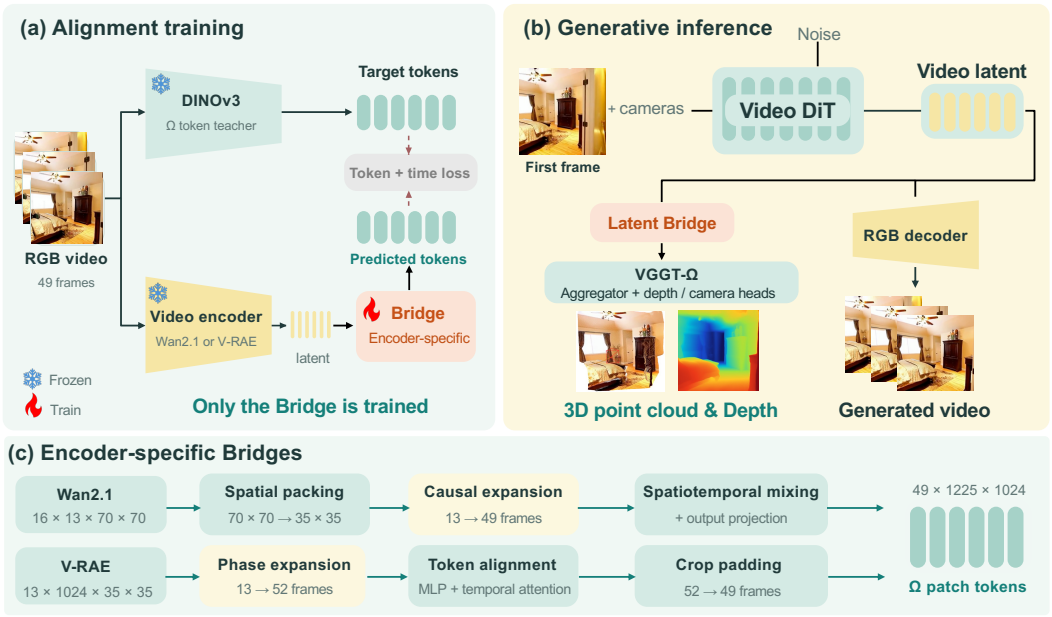


Figure 2: Overview of Latent Bridge, showing (a) feature alignment training, (b) video generation and geometry prediction from a shared latent, and (c) the architectures of bridges.

4.1 Generative–Reconstruction Latent-Space Alignment.

Given an RGB video $\mathbf{X} \in \mathbb{R}^{T\times 3\times H\times W}$, we consider a frozen video encoder E_e, where $e \in \{\text{V}, \text{A}\}$ identifies the V-RAE or Wan representation. The encoder produces a compressed video latent $\mathbf{Z}^e = E_e(\mathbf{X})$. An encoder-specific bridge B_{θ_e} maps this latent to frame-level tokens that match the outputs of the frozen VGGT-Ω patch encoder Φ_Ω:

$$\begin{aligned} \widehat{\mathbf{Y}}^{e} &= B_{\theta_e}\left(E_e(\mathbf{X})\right) \in \mathbb{R}^{T\times N\times C}, \\ \mathbf{Y} &= \Phi_{\Omega}(\mathbf{X}) \in \mathbb{R}^{T\times N\times C}, \qquad e \in \{\mathrm{V}, \mathrm{A}\}, \end{aligned} \tag{4}$$

where $\widehat{\mathbf{Y}}^{e}$ denotes the tokens predicted from latent representation e, and $\mathbf{Y}$ denotes the target tokens extracted from RGB frames by the frozen Omega patch encoder Φ_{Ω}. Here, T is the number of frames, N is the number of spatial tokens per frame, and C is the token dimension. For videos with $T = 49$ frames and spatial resolution $H = W = 560$, the native Omega representation contains $N = 35 \times 35 = 1225$ tokens per frame with $C = 1024$ channels. Our objective is to align $\widehat{\mathbf{Y}}^{e}$ with $\mathbf{Y}$. B_{θ_V} and B_{θ_A} do not share parameters because their input representations have different tensor structures. Nevertheless, they share the same target token space, alignment objective, and frozen geometry decoder.

4.2 Encoder-Specific Latent Bridges

Wan Raw-Latent Bridge. The Wan branch operates directly on the raw output of the frozen causal video VAE. For a 49-frame 560× 560 video, the Wan encoder produces $\mathbf{Z}^{\mathrm{A}} = E_A(\mathbf{X}) \in \mathbb{R}^{T_z\times H_z\times W_z\times C_z}$, where $C_z = 16$, $T_z = 13$, and $H_z = W_z = 70$. The target dimension is $\widehat{\mathbf{Y}}^{A} \in \mathbb{R}^{T\times N\times C}$, where $C = 1024$, $T = 49$, and $N = (H/16 \times W/16) = 35 \times 35$. Therefore, we need to spatially downsample the features, restore the temporal dimension to 49 frames, and map them from a lower-dimensional space to a higher-dimensional one.

As shown in Fig. 2(c), the Wan bridge first applies 2×2 spatial packing to map the 70×70 latent grid to the 35×35 VGGT-Ω patch grid, followed by a learned channel projection. Causal phase expansion then maps $T_z = 13$ latent steps to $T = 49$ frame positions according to $T = 1 + r(T_z - 1)$, where $r = 4$. The first latent step corresponds to the first frame, while each subsequent step produces four learned temporal phases. Sinusoidal time embeddings and learned phase embeddings distinguish the resulting frame positions. The expanded features are refined through local spatial mixing, per-location cross-frame attention, and channel-wise feed-forward layers. Finally, an output projection produces $\widehat{\mathbf{Y}}^{\mathrm{A}} \in \mathbb{R}^{49\times1225\times1024}$.

V-RAE Semantic-Latent Bridge. The V-RAE branch operates on a semantically richer latent representation. A frozen DINOv3 K7 encoder extracts frame-level feature grids, and a frozen temporal pooler compresses each group of four frames into one latent chunk. For a 49-frame video at 560×560 resolution, the V-RAE latent has shape $\mathbf{Z}^{\mathrm{V}} \in \mathbb{R}^{13\times1024\times35\times35}$, matching the spatial grid and channel dimension of the VGGT-Ω tokens. The target token tensor has shape $\widehat{\mathbf{Y}}^{\mathrm{V}} \in \mathbb{R}^{49\times1225\times1024}$. Thus, temporal expansion is sufficient to match the target dimensions.

To align the conditioning frame with the following 48 frames, we prepend three copies of the first frame before temporal pooling, producing a 52-frame input. The temporal pooling module produces 13 latent chunks, and the V-RAE bridge expands each chunk into four phase-specific frame features, yielding 52 intermediate features. Because the V-RAE latent already matches the spatial grid size and channel dimension of the VGGT-Ω tokens, no spatial resampling is required. The bridge applies a residual spatial MLP to align the expanded features with VGGT-Ω tokens, followed by per-patch temporal attention to model relationships across frames. Finally, the first three padding phases are removed to recover the 49-frame sequence, yielding $\widehat{\mathbf{Y}}^{\mathrm{V}} \in \mathbb{R}^{49\times1225\times1024}$.

4.3 Shared Token Alignment Objective

Both bridges are trained using the same frozen Omega patch tokens as distillation targets. Only the selected bridge is optimized. All pretrained components remain frozen, and the downstream aggregator and geometry heads are not executed during bridge training. The alignment objective combines token reconstruction, directional feature alignment, adjacent-frame dynamics, and sequence-level temporal relations:

$$\begin{aligned} \mathcal{L}_{\mathrm{align}} = \lambda_{\mathrm{mse}}\,\mathrm{MSE}\left(\widehat{\mathbf{Y}}, \mathbf{Y}\right) + \lambda_{\cos}\left[1 - \mathrm{Cos}\left(\widehat{\mathbf{Y}}, \mathbf{Y}\right)\right] \\ + \lambda_{\Delta}\,\mathrm{Huber}\left(\Delta_t\widehat{\mathbf{Y}}, \Delta_t\mathbf{Y}\right) + \lambda_{\mathrm{rel}}\,\mathrm{MSE}\left(\mathcal{R}\left(\widehat{\mathbf{Y}}\right), \mathcal{R}\left(\mathbf{Y}\right)\right), \end{aligned} \tag{5}$$

where $\mathrm{MSE}(\cdot,\cdot)$ is the mean squared error, $\mathrm{Cos}(\cdot,\cdot)$ is the mean token-wise cosine similarity, and $\mathrm{Huber}(\cdot,\cdot)$ denotes the smooth ℓ_1 loss. The temporal difference is defined as $\Delta_t \mathbf{Y}_t = \mathbf{Y}_{t+1} - \mathbf{Y}_t$.

For a token sequence $\mathbf{Y} \in \mathbb{R}^{T\times N\times C}$, we average the tokens within each frame and ℓ_2-normalize the resulting vectors. Stacking these vectors as rows gives $\mathbf{F} \in \mathbb{R}^{T\times C}$, from which we compute $\mathcal{R}(\mathbf{Y}) = \mathbf{F}\mathbf{F}^\top \in \mathbb{R}^{T\times T}$. Each entry $R(Y)_{ij}$ measures the cosine similarity between frames i and j. The MSE and cosine terms align feature values and directions, respectively. The temporal difference term preserves local feature changes between adjacent frames, while the relation term captures broader temporal relationships across the sequence. Together, these terms encourage feature fidelity and temporal consistency with the teacher. We set $\lambda_{\mathrm{mse}} = \lambda_{\mathrm{cos}} = 1$, $\lambda_\Delta = 0.5$, and $\lambda_{\mathrm{rel}} = 0.1$.

4.4 Direct Geometry Decoding

At inference, the trained bridge takes the final denoised latent from a compatible video generator as input. Let G_e denote the generator associated with representation e. Given a generation condition $\mathbf{c}$, such as a reference image and a target camera trajectory, and a noise variable $\boldsymbol{\epsilon}$, the generator produces a latent $\mathbf{Z}^{g,e}$:

$$\begin{aligned} \mathbf{Z}^{g,e} &= G_e(\mathbf{c}, \boldsymbol{\epsilon}), \\ \widehat{\mathbf{X}}^e &= D_e\left(\mathbf{Z}^{g,e}\right), \\ \left(\widehat{\mathbf{d}}^e, \widehat{\mathbf{K}}^e, \widehat{\mathbf{E}}^e\right) &= H_\Omega\left(A_\Omega\left(B_{\theta_e}\left(\mathbf{Z}^{g,e}\right)\right)\right). \end{aligned} \tag{6}$$

Here, D_e denotes the optional RGB decoder of the corresponding video representation, and $\widehat{\mathbf{X}}^e$ is the generated RGB video. The frozen Omega aggregator and geometry heads are denoted by A_Ω and H_Ω, respectively. Their outputs $\widehat{\mathbf{d}}^e$, $\widehat{\mathbf{K}}^e$, and $\widehat{\mathbf{E}}^e$ represent the predicted depth maps, camera intrinsics, and camera extrinsics.

We obtain world-space point clouds $\hat{P}^e = \Pi^{-1}(\hat{d}^e, \hat{K}^e, \hat{E}^e)$, where Π^{-1} denotes depth unprojection into a shared world coordinate system using the predicted camera parameters. The geometry branch operates directly on generated latents, bypassing RGB decoding and image feature extraction.

5 Experiments

Data We train the camera-conditioned video generation model and the alignment model on a mixture of DL3DV (Ling et al., 2024), WildRGB-D (Xia et al., 2024), CO3Dv2 (Reizenstein et al., 2021), and RealEstate10K (Zhou et al., 2018). To balance their contributions, we sample datasets at a ratio of 2:1:1:1, then randomly sample a contiguous 49-frame video window from an eligible segment in the selected dataset. All video frames are resized and cropped to 560×560, with corresponding adjustments to the camera intrinsics.

Baseline and Metrics. We selected methods that perform simultaneous generation and reconstruction for comparison, including Aether (Zhu et al., 2025), FantasyWorld (Dai et al., 2026) and Gen3R (Huang et al., 2026). Aether's maximum generation supports only 41 frames, so its metrics are calculated based on 41 frames. We evaluate the quality of the reconstruction using Accuracy, Completeness, and Chamfer Distance (CD) (Fan et al., 2017). We evaluate the generated results against the ground truth using PSNR, SSIM (Wang et al., 2004), and LPIPS (Zhang et al., 2018), and assess the model's generation capabilities using the I2V Subject (I2V Subj.), I2V Background (I2V BG), and Imaging Quality (I.Q.) metrics from VBench (Huang et al., 2024).

Implementation Details. The Wan2.1 bridge is trained for 94K steps with global batch size 32 on 32 GPUs (NVIDIA RTX PRO 6000, then RTX 5090). The V-RAE bridge is trained for 50K steps using 16 RTX PRO 6000 GPUs. Both bridges use AdamW, BF16, weight decay of 10^{-4}, and gradient clipping at 1. Learning rates are 5×10^{-5} with 1K-step warmup for Wan2.1 and a constant 5×10^{-5} for V-RAE. These stages cost approximately 1,501 and 498 GPU-hours, respectively.

Comparison of Reconstruction Results. We selected 300 Co3Dv2 and 158 WildRGB-D sequences to evaluate the reconstruction performance of different methods. These datasets include ground-truth point clouds. We also present the end-to-end latency from input image to generated

Table 1: Geometry reconstruction results on Co3Dv2 and WildRGB-D. Geometry metrics are measured in meters and latency in seconds. The best and second-best results are highlighted in pink and yellow, respectively.

Method	Co3Dv2			WildRGB-D			Latency ↓
	Accuracy ↓	Completeness ↓	CD ↓	Accuracy ↓	Completeness ↓	CD ↓	
Aether (Zhu et al., 2025)	0.6982	0.5437	0.6209	0.0591	0.1340	0.0965	181.33
FantasyWorld (Dai et al., 2026)	0.5939	0.3698	0.4818	0.0440	0.1258	0.0849	448.63
Gen3R (Huang et al., 2026)	0.3820	**0.1796**	0.2808	0.0247	0.1129	0.0688	189.57
Ours (Wan2.1)	0.3536	0.1833	0.2685	**0.0189**	0.1209	0.0699	**70.09**
Ours (V-RAE)	**0.3231**	0.2093	**0.2662**	0.0237	**0.1076**	**0.0657**	165.57

video and point cloud. Our comparison method is shown in Tab. 1. Our V-RAE variant achieves the lowest Chamfer Distance on both datasets, while the Wan2.1 variant achieves the lowest accuracy error on WildRGB-D. Our Wan2.1 variant also achieves the lowest end-to-end latency among the evaluated systems. Table 3 separately isolates the computational savings from replacing RGB decoding and image encoding with the Bridge. Figure 3 presents our I2V results alongside the corresponding depth maps and 3D visualizations, demonstrating that Latent Bridge recovers sharp depth maps and produces accurate 3D point clouds.

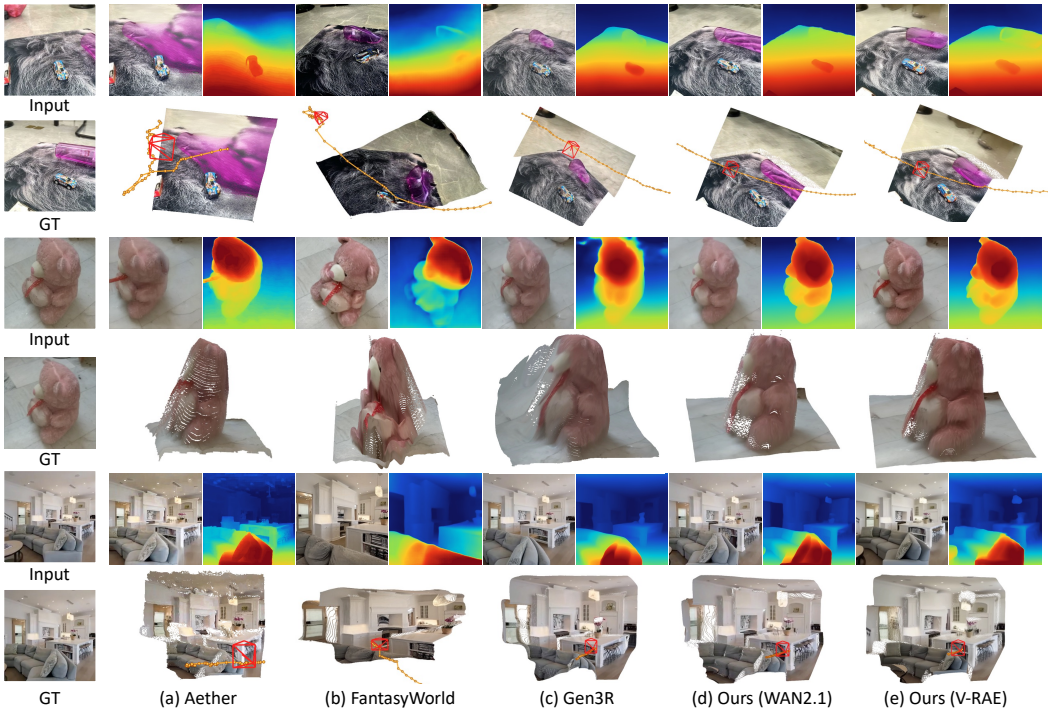


Figure 3: Visual comparison of generation and reconstruction results, including the generated videos, depth maps, and point clouds produced by each method.

Comparison of generation results. We randomly selected 300 Co3Dv2, 158 WildRGB-D, 300 RealEstate10K, and 140 DL3DV sequences to compare the generation quality. Our comparison method is shown in the Tab. 2. In terms of video generation, the Wan2.1 base model produces the highest quality output. V-RAE exhibits slightly lower generation quality than the Wan2.1 VAE. Notably, V-RAE uses a compression ratio of $16 \times 16 \times 4$, which is higher than the $8 \times 8 \times 4$ compression ratio of the Wan2.1 VAE. Despite this, V-RAE still outperforms Wan2.1 on the 3D metrics evaluated on CO3D, as its native DINOv3 features facilitate more effective alignment.

Comparison of RGB decoding solutions. We further analyzed the differences between our alignment latent scheme and the RGB decoding and reconstruction scheme, and the results are shown in Tab. 3. For the same generated latents, we compare feed-forward reconstruction via RGB decoding

Table 2: Video Generation Quality Comparison. All results are evaluated at 560×560 resolution. I2V Subj., I2V BG, and I.Q. are multiplied by 100.

Method	Co3Dv2						WildRGB-D					
	PSNR ↑	SSIM ↑	LPIPS ↓	I2V Subj. ↑	I2V BG ↑	I.Q. ↑	PSNR ↑	SSIM ↑	LPIPS ↓	I2V Subj. ↑	I2V BG ↑	I.Q. ↑
Aether	13.9595	0.5689	0.5942	90.64	92.13	38.46	12.4790	0.4466	0.5660	95.82	95.94	52.60
FantasyWorld	12.0307	0.4807	0.6741	89.08	91.27	46.20	12.9733	0.4385	0.5116	97.07	96.72	59.41
Gen3R	16.5289	0.6027	0.4405	94.79	96.05	45.20	15.2252	0.5075	0.4025	98.38	98.20	56.58
Ours (V-RAE)	16.3780	0.6025	0.4303	94.91	95.99	**47.84**	**16.5769**	**0.5516**	**0.3217**	97.87	97.60	56.85
Ours (Wan2.1)	**17.0388**	**0.6166**	**0.4121**	**95.42**	**96.41**	46.73	16.1703	0.5412	0.3394	**98.44**	**98.24**	**59.67**
Method	DL3DV						RealEstate10K					
	PSNR ↑	SSIM ↑	LPIPS ↓	I2V Subj. ↑	I2V BG ↑	I.Q. ↑	PSNR ↑	SSIM ↑	LPIPS ↓	I2V Subj. ↑	I2V BG ↑	I.Q. ↑
Aether	12.9066	0.4211	0.5576	93.93	94.90	46.21	16.5257	0.5865	0.3439	98.82	98.60	47.21
FantasyWorld	12.1378	0.3633	0.5944	95.09	95.85	47.60	14.4308	0.5353	0.4776	97.89	97.84	**50.98**
Gen3R	14.5473	0.4234	0.4477	94.09	95.28	47.32	19.9467	0.6781	**0.2115**	**99.31**	99.08	49.82
Ours (V-RAE)	14.7968	0.4448	0.4351	94.75	95.62	48.25	17.5092	0.6384	0.3028	98.42	97.79	47.61
Ours (Wan2.1)	**14.8615**	**0.4453**	**0.4169**	**95.22**	**95.90**	**50.96**	**20.0350**	**0.6917**	0.2125	99.25	**99.22**	49.02

and image encoding with reconstruction through the Bridge, using identical frozen VGGT-Ω aggregators and prediction heads. The results show that the reconstruction accuracy of the Bridge-based approach is comparable to that of the RGB-decoding pipeline, indicating that the Bridge achieves sufficient fitting. The Bridge preserves comparable reconstruction quality while reducing the measured latent-to-geometry latency from 7.92 s to 4.92 s for Wan2.1 and from 5.76 s to 4.64 s for V-RAE, corresponding to speedups of 1.61× and 1.24×, respectively.

Table 3: Comparison of RGB Decoding Solutions.

Method	CO3Dv2			WildRGB-D			Time (s) ↓
	Accuracy ↓	Completeness ↓	CD ↓	Accuracy ↓	Completeness ↓	CD ↓	
Latent → RGB → VGGT-Ω (Wan2.1)	0.3559	**0.1754**	**0.2657**	**0.0174**	0.1227	0.0700	7.92
Latent → Bridge → VGGT-Ω (Wan2.1)	**0.3536**	0.1833	0.2685	0.0189	**0.1209**	**0.0699**	**4.92**
Latent → RGB → VGGT-Ω (VRAE)	0.3241	**0.1925**	**0.2583**	**0.0209**	0.1114	0.0661	5.76
Latent → Bridge → VGGT-Ω (VRAE)	**0.3231**	0.2093	0.2662	0.0237	**0.1076**	**0.0657**	**4.64**

6 Ablation Study and Generalization Evaluation

Parameter settings. In this ablation study, we investigate how bridge capacity affects latent-space alignment. As shown in Fig. 4, we vary only the internal hidden width of the bridge network to control its capacity, keeping all other settings fixed within each model family. For the Wan Bridge, widths of 256, 512, and 768 yield 9.11M, 35.65M, and 79.62M trainable parameters, respectively. For the VRAE Bridge, we vary the hidden width of the spatial mapping MLP and temporal FFN across 1024, 2048, 4096, and 6144, yielding 12.61M, 16.80M, 25.20M, and 33.59M parameters. All alignment variants are trained from scratch on a mixture of four datasets and evaluated for teacher-depth consistency on a fixed set of 64 validation videos. We compute teacher-normalized L1 and AbsRel over all pixels in all 49 frames, without foreground masking or prediction-scale fitting.

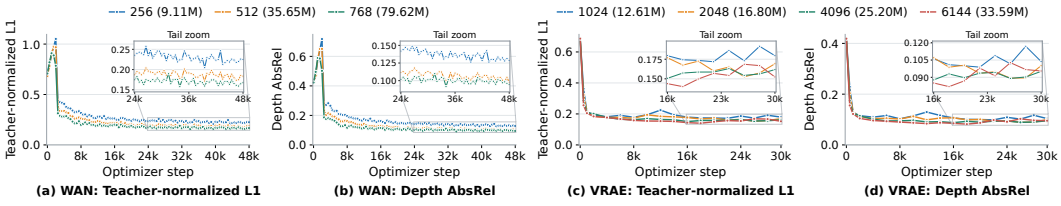


Figure 4: Ablation study on the number of parameters.

Network structure. We conduct structural ablations of the Bridge on the Wan branch. Starting from the full Bridge, we separately replace spatial packing with average pooling, learnable causal temporal expansion with fixed linear interpolation, and spatiotemporal mixing with a token-wise MLP to assess each component's contribution. All variants are trained from scratch using the same training data, frozen Wan encoder, VGGT-Ω downstream network, distillation losses, and optimization settings. We additionally introduce a simple baseline in which all three aforementioned components are replaced. We evaluate them at matched training steps on a fixed validation set of 64

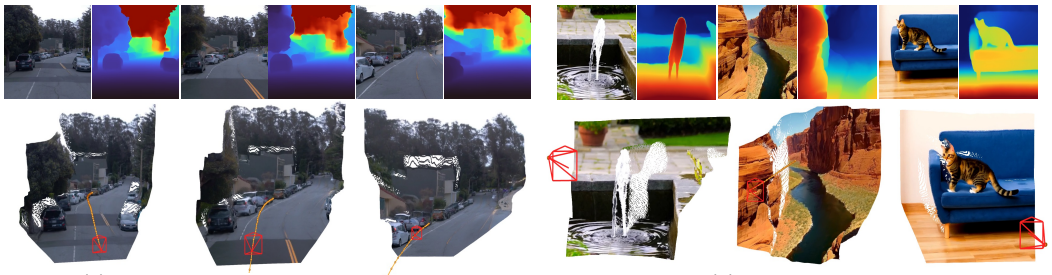
videos, each containing 49 frames, using full-image teacher-normalized L1 and depth AbsRel to measure depth distillation consistency. As shown in Tab. 4, replacing any individual component leads to varying degrees of accuracy degradation, demonstrating that our network architecture is well-designed and has a strong capacity for feature fitting.

Loss function. We conduct an ablation study on the training loss function. As shown in Tab. 4, we refer to the second, third, and fourth terms in equation 5 as the cosine loss, delta loss, and relation loss, respectively. We ablate each loss term and evaluate its fitting capability. As shown in the table, removing any loss term leads to a degradation in accuracy, demonstrating that each constraint contributes to the overall performance. In practice, the weights of the individual loss terms can be adjusted according to the specific characteristics of each Bridge.

Table 4: Ablations of the Wan Bridge at 50K optimization steps. All variants are evaluated on the same 64 clips of 49 frames.

Structural ablation	Norm. L1 ↓	Depth L1 ↓	AbsRel ↓	Loss ablation	Norm. L1 ↓	Depth L1 ↓	AbsRel ↓
w/o 2×2 PixelUnshuffle	**0.18057**	0.15849	0.10628	w/o Cosine loss	0.18823	0.16459	0.10851
w/o Causal Conv3D + phase rearrangement	0.19177	0.16593	0.10766	w/o Delta loss	0.18780	0.16235	0.10461
w/o Spatiotemporal mixing blocks	0.25196	0.22100	0.15228	w/o Relation loss	0.18412	0.15863	0.10313
Simple baseline (joint replacements)	0.31666	0.27490	0.19346	**Full Bridge & loss**	0.18133	**0.15644**	**0.10163**

Scene generalization evaluation. The alignment module can be transferred across different generative models trained using the same VAE. As shown in Fig. 5(a), we utilized the Waymo dataset (Sun et al., 2020) to train a generative model for driving scenarios, based on the Wan 2.1 VAE. The alignment network can be directly used to decode 3D output without the need for retraining.



(a) Autonomous driving scene evaluation. (b) T2V scene evaluation.

Figure 5: The Bridge can adapt to generative models for different tasks that use the same VAE, without retraining.

T2V model generalization evaluation. To verify the reusability of the geometry interface, we directly integrate the existing Bridge with the official Wan2.1-T2V-1.3B model. The generator operates in the latent space of its accompanying Wan2.1 VAE, allowing its final denoised latents to be fed directly into Latent Bridge without RGB re-encoding. The frozen VGGT-Ω then predicts depth and camera parameters. This integration requires no additional T2V-specific training, parameter updates, or test-time optimization. As shown in Fig. 5(b), the experiment demonstrates that the geometry decoding interface can be reused with another generator sharing a compatible VAE latent space, without retraining the generative model.

7 Conclusion

We presented Latent Bridge, a framework that connects video generation and feed-forward 3D reconstruction through latent-space feature alignment. By mapping Wan2.1 and V-RAE video representations to a predefined geometric feature interface, our method enables direct prediction of cameras, depth, and point clouds while keeping the downstream geometric model frozen. This design bypasses RGB decoding and visual re-encoding in the geometry branch, allowing a shared video latent representation to support both video synthesis and geometry prediction. Experiments on Co3Dv2 and WildRGB-D demonstrate competitive reconstruction quality with improved inference efficiency. More broadly, our results suggest that feature alignment offers a practical route to reusing pretrained geometric models with generative video representations.

AI USE STATEMENT

In this work, we used generative AI tools to assist with writing and language polishing, including refining English expressions. We also used these tools for literature retrieval and discovery to identify potentially overlooked references. In addition, we used them to support research ideation and execution, particularly in writing visualization scripts and developing a 3D visualization interface. In addition, we used AI to review drafts, identify potential issues, and provide suggestions. We have not used generative AI tools to directly write any sections of the paper, and generating synthetic datasets and proving mathematical claims are not applicable to this work. We have reviewed all AI-assisted work. We verified the accuracy of the related-work information provided by generative AI and reviewed and validated the generated code. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Zhaochong An, Orest Kupyn, Théo Uscidda, Andrea Colaco, Karan Ahuja, Serge Belongie, Mar Gonzalez-Franco, and Marta Tintore Gazulla. Vggrpo: Towards world-consistent video generation with 4d latent reward. *arXiv preprint arXiv:2603.26599*, 2026.

Sherwin Bahmani, Tianchang Shen, Jiawei Ren, Jiahui Huang, Yifeng Jiang, Haithem Turki, Andrea Tagliasacchi, David Lindell, Zan Gojcic, Sanja Fidler, et al. Lyra: Generative 3d scene reconstruction via video diffusion model self-distillation. In *International Conference on Learning Representations*, volume 2026, pp. 82850–82880, 2026.

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. *arXiv preprint arXiv:2311.15127*, 2023.

Yixiang Dai, Fan Jiang, Chiyu Wang, Mu Xu, and Yonggang Qi. Fantasyworld: Geometry-consistent world modeling via unified video and 3d prediction. In *International Conference on Learning Representations*, volume 2026, pp. 103603–103622, 2026.

Haoqiang Fan, Hao Su, and Leonidas J Guibas. A point set generation network for 3d object reconstruction from a single image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 605–613, 2017.

Sensen Gao, Zhaoqing Wang, Qihang Cao, Dongdong Yu, Changhu Wang, and Jia-Wang Bian. Pixworld: Unifying 3d scene generation and reconstruction in pixel space. *arXiv preprint arXiv:2607.05373*, 2026a.

Yulu Gao, Bohao Zhang, Zongheng Tang, Jitong Liao, Wenjun Wu, and Si Liu. Vggt-segmentor: Geometry-enhanced cross-view segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21690–21700, June 2026b.

Hyojun Go, Dominik Narnhofer, Goutam Bhat, Prune Truong, Federico Tombari, and Konrad Schindler. Text-to-3d by stitching a multi-view reconstruction network to a video generator. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=kI27Niy4xY.

Minghui Guo, Shengqiong Wu, and Hao Fei. V-rae: Rethinking video latent spaces for generation. *arXiv preprint arXiv:2608.13556*, 2026.

Jiaxin Huang, Yuanbo Yang, Bangbang Yang, Lin Ma, Yuewen Ma, and Yiyi Liao. Gen3r: 3d scene generation meets feed-forward reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 25358–25369, 2026.

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 *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21807–21818. IEEE, 2024.

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. *arXiv preprint arXiv:2412.03603*, 2024.

Zhenjia Li, Jinrang Jia, and Yifeng Shi. Pano2world: End-to-end 3d generation via unified multi-view sequences. *arXiv preprint arXiv:2607.00832*, 2026.

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 *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22160–22169. IEEE, 2024.

Zihao Liu, Xiaolong Shen, Zhenglin Zhou, Ruijie Quan, and Yi Yang. Beyond pixels: From video priors to 4d worlds. *arXiv preprint arXiv:2608.10744*, 2026.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel HAZIZA, Francisco Massa, Alaaeldin El-Nouby, Mido Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=a68SUt6zFt. Featured Certification.

Jeremy Reizenstein, Roman Shapovalov, Philipp Henzler, Luca Sbordone, Patrick Labatut, and David Novotny. Common objects in 3d: Large-scale learning and evaluation of real-life 3d category reconstruction. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 10881–10891. IEEE, 2021.

Oriane Siméoni, Huy V. Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seung Eun Yi, Michael Ramamonjisoa, Francisco Massa, Daniel HAZIZA, Luca Wehrstedt, Jianyuan Wang, Timothée Darcet, Théo Moutakanni, Leonel Sentana, Claire Roberts, Andrea Vedaldi, Jamie Tolan, John Brandt, Camille Couprie, Julien Mairal, Herve Jegou, Patrick Labatut, and Piotr Bojanowski. DINOv3. *Transactions on Machine Learning Research*, 2026. ISSN 2835-8856. URL https://openreview.net/forum?id=2NlGyqNjns. Featured Certification.

Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2443–2451. IEEE, 2020.

Xiangyu Sun, Shijie Wang, Fengyi Zhang, Lin Liu, Caiyan Jia, Ziying Song, Zi Huang, and Yadan Luo. Vggt-world: Transforming vggt into an autoregressive geometry world model. *arXiv preprint arXiv:2603.12655*, 2026.

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. *arXiv preprint arXiv:2503.20314*, 2025.

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

Jianyuan Wang, Minghao Chen, Shangzhan Zhang, Nikita Karaev, Johannes Schönberger, Patrick Labatut, Piotr Bojanowski, David Novotny, Andrea Vedaldi, and Christian Rupprecht. VGGT-Ω. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026.

Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20697–20709. IEEE, 2024.

Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

Hongchi Xia, Yang Fu, Sifei Liu, and Xiaolong Wang. Rgbd objects in the wild: Scaling real-world 3d object learning from rgb-d videos. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22378–22389. IEEE, 2024.

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. In *International Conference on Learning Representations*, volume 2025, pp. 83048–83077, 2025.

Jingfeng Yao, Bin Yang, and Xinggang Wang. Reconstruction vs. generation: Taming optimization dilemma in latent diffusion models. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15703–15712. IEEE, 2025.

Sihyun Yu, Sangkyung Kwak, Huiwon Jang, Jongheon Jeong, Jonathan Huang, Jinwoo Shin, and Saining Xie. Representation alignment for generation: Training diffusion transformers is easier than you think. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025.

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *2018 IEEE/CVF conference on computer vision and pattern recognition*, pp. 586–595. IEEE, 2018.

Boyang Zheng, Nanye Ma, Shengbang Tong, and Saining Xie. Diffusion transformers with representation autoencoders. In *International Conference on Learning Representations*, volume 2026, pp. 35791–35820, 2026.

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

Haoyi Zhu, Yifan Wang, Jianjun Zhou, Wenzheng Chang, Yang Zhou, Zizun Li, Junyi Chen, Chunhua Shen, Jiangmiao Pang, and Tong He. Aether: Geometric-aware unified world modeling. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 8535–8546. IEEE, 2025.

A APPENDIX

In the appendix, we provide further methodological and experimental details. In Section A.1, we provide a detailed comparison between our method and existing approaches. In Section A.2, we examine downstream reuse for cross-view segmentation by connecting a VGGT-aligned Bridge to a frozen pretrained segmentation module. In Section A.3, we provide detailed information on data usage and evaluation protocols. In Section A.4, we discuss the limitations of our method. **We also include visualization videos in the supplementary material.**

A.1 COMPARISON WITH EXISTING METHODS

As shown in Tab. 5, we compare our method with existing approaches from two perspectives: compatibility with existing models and training overhead. Cross-DiT checkpoint reuse refers to whether the method can produce 3D outputs using different generative DiT checkpoints without retraining. FantasyWorld trains a geometry branch and cross-branch adapters while keeping the video backbone frozen, whereas Gen3R fine-tunes the video DiT to jointly generate appearance and geometry latents. Beyond Pixels and our method instead learn geometry pathways from video latents without updating the DiT, enabling reuse across compatible generators that share the same VAE checkpoint and latent conventions. We demonstrate this reuse by applying the same Wan Bridge checkpoint to camera-conditioned generation and the official Wan2.1-T2V-1.3B model without additional Bridge training. Pretrained task-head reuse refers to the ability to reuse existing downstream heads because our method outputs features in VGGT's DINO feature space. Consequently, the resulting features can be used not only for reconstruction with VGGT but also for cross-view segmentation with VGGT-S, as shown in Section A.2. Gen3R reconstructs intermediate VGGT geometry tokens, which are decoded by pretrained geometry heads. In contrast, FantasyWorld and Beyond Pixels incorporate their decoder heads into training and therefore do not offer this advantage. Therefore, our method offers substantially greater versatility than existing approaches.

In addition, we compare the additional training costs required to support both generation and reconstruction. FantasyWorld trains a geometry branch and cross-branch adapters, while Gen3R fine-tunes the video DiT for joint video and geometry generation. Beyond Pixels trains an alignment module, LoRA adapters, and geometry prediction heads. In contrast, we train only the Bridge through feature alignment, keeping the video encoder, geometric aggregator, and prediction heads frozen. Keeping the decoder heads frozen further reduces the training cost. Furthermore, FantasyWorld and Gen3R introduce additional geometry tokens. Our bridge is trained using feature supervision alone, without executing the downstream geometric aggregator or prediction heads during optimization.

Table 5: Differences Between Our Method and Existing Approaches.

Method	Pretrained Checkpoint Reuse without Further Training			Trainable Modules and Token Overhead.	
	Cross-DiT checkpoint	T2V/I2V checkpoint	Pretrained task-head	Trainable components	Additional Geometry Tokens
FantasyWorld	No	No	No	Geometry branch & adapters	Yes
Gen3R	No	No	Yes	DiT & geometry adapter	Yes
Beyond Pixels	Yes	Yes	No	Alignment module, LoRA & heads	No
VIST3A	Yes	-	No	Stitching layer & geometry/DiT LoRA	No
Latent Bridge	Yes	Yes	Yes	Bridge only	No

To compare the alignment efficiency of feature supervision with that of geometry supervision provided through the frozen decoder heads, we conduct an experiment, as shown in Table 6. Our feature-supervised bridge requires only 0.166 s and 3.17 GiB per forward–loss–backward pass, compared with 5.896 s and 31.27 GiB for geometry supervision with frozen VGGT-Ω. Unfreezing its aggregator and camera/depth heads further increases the cost to 6.375 s and 35.48 GiB. This demonstrates that feature distillation before the frozen decoder heads is substantially less computationally expensive than applying supervision after the features have passed through the frozen decoder heads.

A.2 DOWNSTREAM REUSE BEYOND GEOMETRY.

Beyond geometry reconstruction, we extend our framework to the standard VGGT feature space and examine downstream reuse. This experiment uses a separately trained VGGT-aligned Bridge, rather than the VGGT-Ω Bridge evaluated above. Within the VGGT-aligned branch, the Bridge checkpoint

Table 6: Computational cost of different supervision strategies. All settings use the same 6000 Pro GPU with a batch size of 1 and 49 frames at 560×560 resolution.

Supervision	Trainable modules	Params (M) ↓	Time (s) ↓	Memory (GiB) ↓
Feature (ours)	Bridge	**35.65**	**0.166**	**3.17**
Geometry	Bridge	**35.65**	5.896	31.27
Geometry	Bridge + geometry model	876.31	6.375	35.48

used for geometry reconstruction also supports cross-view segmentation through a frozen pretrained VGGT-S (Gao et al., 2026b) module. This reuse requires only resolution and feature-interface adjustments, without additional training or fine-tuning. As shown in Fig. 6, given a first-frame object mask, the pipeline propagates masks across generated frames without decoding the generated video into RGB. We present additional results in Fig. 7. Associating these masks with the predicted 3D coordinates further produces label-colored object point clouds. These results demonstrate task reuse within the VGGT-aligned branch, beyond its original geometry application.

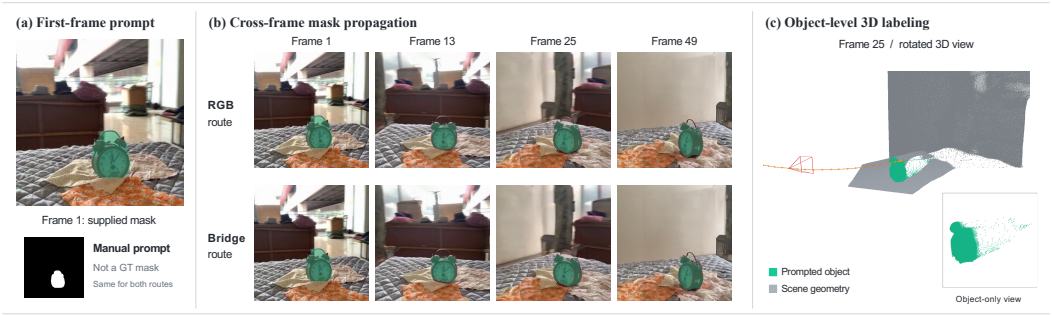


Figure 6: Downstream reuse with a frozen VGGT-S module. Given the same first-frame mask prompt, the RGB and Bridge routes propagate object masks across generated frames. Predicted geometry enables object-level 3D labeling.



Figure 7: Additional downstream reuse examples showing generated RGB frames, propagated object masks, and the corresponding label-colored 3D point clouds.

A.3 Datasets and Evaluation Protocol.

Datasets. Table 7 summarizes dataset usage and split sizes. Counts refer to unique source scenes, sequences, or video clips rather than sampled training windows. We follow the official scene lists for DL3DV, the official novel-view-synthesis split for WildRGB-D, and the official train/test split for RealEstate10K (RE10K). For CO3Dv2, we construct a category-stratified 90/10 split at the sequence level. Evaluation uses 140 DL3DV scenes, 158 valid WildRGB-D scenes, 300 CO3Dv2 sequences,

and 300 RE10K clips. For WildRGB-D, we exclude two of the 160 evaluation candidates because they contain no valid global ground-truth points.

Table 7: Dataset usage and splits. Counts refer to unique source scenes, sequences, or video clips, rather than sampled training windows.

Dataset	Usage	Training	Validation Pool	Test	Split Basis
DL3DV	Training; video evaluation	6,209	140	140	Official scene lists with validity filtering
WildRGB-D	Training; geometry & video evaluation	5,786	160	158	Official NVS train/validation split
CO3Dv2	Training; geometry & video evaluation	19,154	2,111	300	Custom category-stratified, sequence-level 90/10 split
RE10K	Training; video evaluation	57,240	6,331	300	Official train/test split; seed-42 test sampling

Geometry metrics. We evaluate reconstruction using Accuracy, Completeness, and Chamfer Distance (CD), with lower values indicating better results. Accuracy measures the mean distance from each predicted point to its nearest GT point:

$$\text{Accuracy}(P, G) = \frac{1}{|P|} \sum_{p \in P} \min_{g \in G} \|p - g\|_2. \tag{7}$$

Completeness measures the reverse distance, quantifying how well the prediction covers the GT geometry:

$$\text{Completeness}(P, G) = \frac{1}{|G|} \sum_{g \in G} \min_{p \in P} \|g - p\|_2. \tag{8}$$

We define CD as the arithmetic mean of these two directional errors:

$$\text{CD}(P, G) = \frac{\text{Accuracy}(P, G) + \text{Completeness}(P, G)}{2}. \tag{9}$$

All distances are Euclidean rather than squared. Metrics are computed on the merged point cloud of each sequence and then averaged with equal weight per sequence within each dataset.

Depth filtering. Depth filtering removes potentially unreliable predicted points and is distinct from foreground segmentation. Candidate pixels are first restricted by the supplied dataset masks and GT-depth validity checks. For each frame, we compute the median and median absolute deviation of finite positive predicted log depths:

$$m = \text{median}_i(\log d_i), \qquad a = \text{median}_i \, |\log d_i - m|. \tag{10}$$

Log-MAD filtering retains candidates satisfying

$$|\log \max(d_i, 10^{-8}) - m| \leq 3 \times 1.4826 \, a. \tag{11}$$

If $a \leq 10^{-8}$, all finite positive candidates are retained. We then check the surviving points for multi-view consistency by back-projecting them into 3D and projecting them into available neighboring frames at offsets $\{-2, -1, 1, 2\}$ using the reconstruction cameras. A neighboring view supports a point when its rounded projected pixel is in bounds, both depths are valid, and

$$\frac{|d_{\text{obs}} - d_{\text{proj}}|}{\max(d_{\text{obs}}, d_{\text{proj}}, 10^{-8})} \leq 0.10. \tag{12}$$

Only points supported by at least two neighboring views are retained. Neighboring depth observations are read directly from the predicted depth maps without requiring them to pass their own foreground or log-MAD masks. Together, these filters suppress depth outliers and cross-view inconsistencies. All methods use the same depth-filtering procedure and thresholds, ensuring consistency in filtering across methods for a fair comparison.

A.4 LIMITATION.

Our Bridge network has the following limitations: First, our model depends on a pretrained geometry model. The bridge learns to match teacher features while keeping the downstream geometry model frozen, and therefore cannot be expected to correct the teacher's geometric errors. Second, inconsistencies in generated videos are not automatically corrected. Our method extracts geometry from generated latents but does not explicitly constrain the video generation process. Therefore, if the generated content exhibits structural changes or object inconsistencies across frames, the Bridge cannot guarantee the reconstruction of a coherent 3D scene.