VGGA: GEOMETRY-GROUNDED AUTOENCODER FOR VIDEO GENERATION

Anonymous authors
Paper under double-blind review

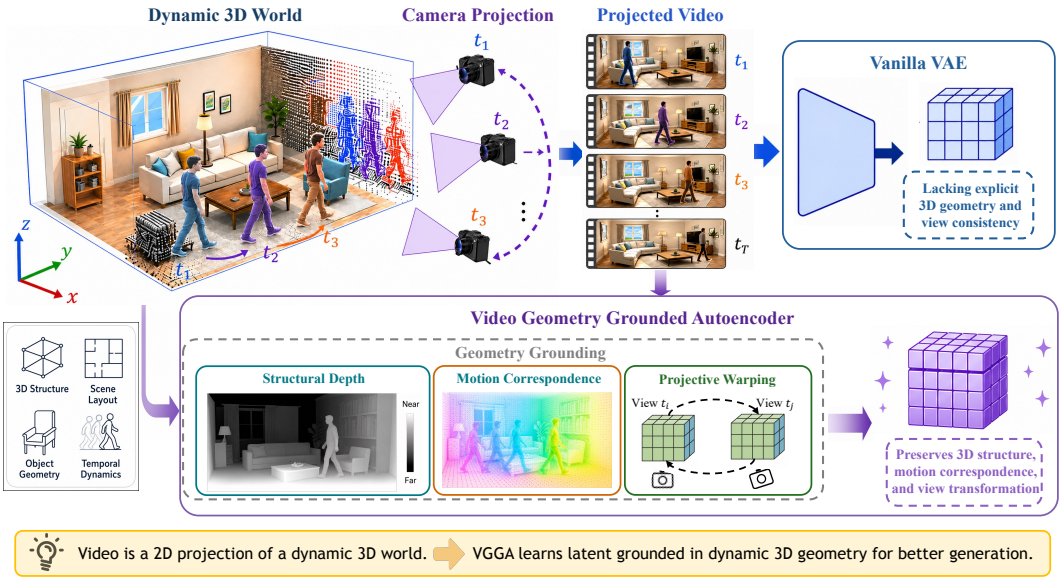


Figure 1: **Geometry-grounded video autoencoding.** Videos arise from the 2D projection of dynamic 3D worlds, whose underlying geometry naturally contains scene layout, dense motion correspondence, and camera-induced projective transformations. VGGA exploits these geometric regularities to organize video latents beyond reconstruction-only supervision, yielding a more compact, structured, and geometry-aware latent space for video generation.

ABSTRACT

Autoencoders are the core component of latent video generative models, compressing high-dimensional videos into compact latent spaces for downstream generation. However, existing video autoencoders are primarily optimized for pixel-level reconstruction, providing limited supervision for learning structured latent representations. We argue that videos should not be treated merely as signals to be reconstructed, but rather as 2D projections of dynamic 3D worlds that exhibit explicit and well-structured geometric properties. Building on this insight, we propose **V**ideo **G**eometry **G**rounded **A**utoencoder (**VGGA**), a geometry-aware video autoencoder that explicitly incorporates geometric constraints into the reconstruction process. Specifically, VGGA grounds the learned representations along three complementary geometric dimensions: *Structural Depth*, *Motion Correspondence*, and *Projective Warping*. These objectives are unified within a Video Geometry Decoder, which employs a shared spatio-temporal upsampling trunk to jointly optimize geometric learning and RGB reconstruction. Through these geometry-grounded objectives, VGGA encourages the latent representations to capture the underlying 3D structure and motion of dynamic scenes. VGGA consistently improves video generation across both geometry-centric and human-motion domains, reducing gFVD by 28.1% on RealEstate10K and 37.7% on Taichi-HD. Notably, these substantial generation improvements are achieved while maintaining comparable reconstruction quality, demonstrating that geometry-grounded representation learning yields a more effective latent space for video generation.

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.

1 INTRODUCTION

Latent diffusion models (Seedance et al., 2026; Gao et al., 2025; Wan et al., 2025; Yang et al., 2024b) have become a dominant paradigm for high-fidelity image and video generation. They typically employ a Variational AutoEncoder (VAE) (Kingma & Welling, 2013) to compress visual signals into a compact latent space, over which a diffusion transformer performs generative modeling (Rombach et al., 2022; Peebles & Xie, 2023; Ma et al., 2024b; Lin et al., 2024; Yu et al., 2024). Beyond compression, the VAE determines what information is preserved and how the latent space is structured, thereby substantially affecting downstream optimization and generation quality (Yao et al., 2025b; Skorokhodov et al., 2025).

Despite its importance, most video autoencoders are still trained primarily for pixel-level reconstruction, providing limited guidance for learning generation-friendly latent representations. Recent works address this issue by aligning VAE latents with vision foundation models (Yu et al., 2025; Ma et al., 2025; Leng et al., 2025), incorporating self-supervised representation learning (Ma et al., 2025; Yao et al., 2025a), or imposing latent regularization such as smoothness and scale equivariance (Liu et al., 2025; Skorokhodov et al., 2025; Kouzelis et al., 2025). These approaches improve semantic structure and latent diffusability beyond reconstruction alone.

However, these efforts still leave two limitations for video autoencoders. First, understanding-oriented supervision often encourages invariance to instance-specific details (Achille & Soatto, 2018; Tian et al., 2020; Li et al., 2023), whereas autoencoding requires preserving fine-grained appearance and motion for faithful reconstruction (Yao et al., 2025b), creating an inherent objective mismatch (Shi et al., 2025) (see Table 2). Second, existing objectives provide little video-specific grounding. Most are inherited from image representation learning or generic latent regularization, while video arises from the projection of a dynamic 3D world, as illustrated in Figure 1. Without explicitly modeling this underlying geometry, downstream generators must infer depth, motion, and viewpoint consistency implicitly from compressed RGB latents.

Therefore, we propose **V**ideo **G**eometry **G**rounded **A**utoencoder (**VGGA**), a geometry-aware video autoencoder for learning generation-friendly video latents. Concretely, we introduce a Video Geometry Decoder that grounds the learned representations along three complementary geometric dimensions: *Structural Depth Grounding* for per-frame 3D structure, *Motion Correspondence Grounding* for dense temporal correspondence, and *Projective Warping Grounding* for camera-induced view transformations. To build an effective VGGA, we focus on two key challenges: **(1) how to harmonize geometric learning with RGB reconstruction**, allowing geometric supervision to enrich the latent representations without degrading reconstruction quality; and **(2) how to enable geometric grounding on large-scale videos** without explicit geometric annotations.

For the first challenge, a key difficulty is that video latents are compressed in both space and time, whereas geometric supervision is defined at the frame level. VGGA therefore adopts a partially shared decoder with a shared spatio-temporal upsampling trunk before the RGB reconstruction and geometry branches (see Figure 2). This design allows geometric supervisions to act directly on the decoded frame-level representations used for reconstruction, encouraging the latent space to encode geometric structure without sacrificing visual fidelity.

For the second challenge, we adopt a hybrid geometric supervision strategy. For general videos, we use pseudo depth and motion cues predicted by off-the-shelf models to perform *Structural Depth Grounding* and *Motion Correspondence Grounding*. In contrast, *Projective Warping Grounding* requires reliable camera poses and assumes static scene geometry, so it cannot account for independent object motion. We therefore apply this objective only to a small set of predominantly static-scene videos with known camera poses. Through joint training, the projective geometry learned from these static scenes is absorbed into the shared latent representation and complements the depth and motion supervision from general videos, allowing camera-aware geometric priors to transfer beyond the statically supervised data and improve generation on dynamic videos. Our main contributions are summarized as follows.

- We present a new perspective on video autoencoding by viewing videos as 2D projections of dynamic 3D scenes, and propose **V**ideo **G**eometry **G**rounded **A**utoencoder (**VGGA**) to inject explicit geometric priors into video latent learning.

- We develop an effective geometry-grounded autoencoding framework that harmonizes geometric learning with RGB reconstruction and supports large-scale training without explicit geometric annotations.
- We demonstrate that VGGA substantially improves downstream video generation while preserving reconstruction quality, demonstrating the effectiveness and compatibility of geometric priors for video autoencoding.

2 Related Work

Video Autoencoders for Latent Generation. Latent generative models (Seedance et al., 2026; Lin et al., 2024) hinge on an underlying autoencoder to define the space for generative modeling. While Latent Diffusion Models (LDMs) (Rombach et al., 2022) demonstrated that operating in a compressed latent space reconciles computational efficiency with visual fidelity, Diffusion Transformers (DiTs) (Peebles & Xie, 2023; Ma et al., 2024a) further validated the scalability of this paradigm. This design is particularly pivotal for video, where the tokenizer must encapsulate both spatial appearance and temporal dynamics (Gao et al., 2025; Yang et al., 2024b; Wan et al., 2025). Despite the heavy reliance of current systems on autoencoder quality (Liu et al., 2025), the training objective remains predominantly reconstruction-centric.

Generation-Friendly Tokenizer Learning. Recent studies (Yao et al., 2025b) suggest that reconstruction fidelity alone does not guarantee effective generative latents, sparking interest in generation-centric tokenization (Ma et al., 2025; Yao et al., 2025a; Skorokhodov et al., 2025). One research trajectory aligns VAE latents with vision foundation models (e.g., DINOv2) to infuse semantic and spatial priors (Yu et al., 2025; Leng et al., 2025). Others integrate understanding-oriented objectives, such as contrastive or self-supervised learning, to refine latent organization (Ma et al., 2025; Yao et al., 2025a). Parallel efforts focus on latent "diffusability" through regularization—leveraging spectral biasing or equivariance (Liu et al., 2025; Skorokhodov et al., 2025; Kouzelis et al., 2025). While these methods enhance latent structure, they primarily adapt image-level paradigms or generic constraints, largely overlooking the intrinsic priors of video formation. VGGA fills this gap by explicitly grounding video autoencoding in geometric constraints.

Geometry Grounding for Video as Dynamic 3D Worlds. Videos are inherently projections of dynamic 3D scenes, positioning geometry as a fundamental source of dense, self-contained supervision. While extensive research has focused on individual geometric cues, such as monocular depth (Yang et al., 2024a), optical flow (Teed & Deng, 2020), and point tracking (Karaev et al., 2024), or leveraged projective consistency for novel view synthesis (Wang et al., 2024), these efforts primarily target geometric perception. These works validate that geometry effectively distills the physical regularities manifest in video observations. VGGA departs from these perception-centric goals by repurposing geometric cues as grounding signals for video autoencoding. Consequently, the resulting latent space is explicitly sculpted by 3D layout and temporal correspondence, moving beyond the limitations of purely pixel-based reconstruction.

3 Method

We propose VGGA to ground video autoencoding in the geometry of dynamic 3D worlds. As shown in Figure 2, VGGA replaces a purely reconstruction-driven decoder with a Video Geometry Decoder. The decoder builds a shared decoded representation for RGB reconstruction and three geometry grounding signals: structural depth, motion correspondence, and projective warping. We start with preliminaries in Section 3.1, followed by the Video Geometry Decoder in Section 3.2, Video Geometry Grounding in Section 3.3, Hybrid Geometry Supervision on Unlabelled Videos in Section 3.4.

3.1 Preliminary

We first introduce the notation and basic objectives used throughout the paper. Let $\mathbf{x} = \{I_t\}_{t=0}^{T} \in \mathbb{R}^{(1+T)\times H\times W\times 3}$ denote a video clip with $1+T$ RGB frames, where I_t is the frame at time t.

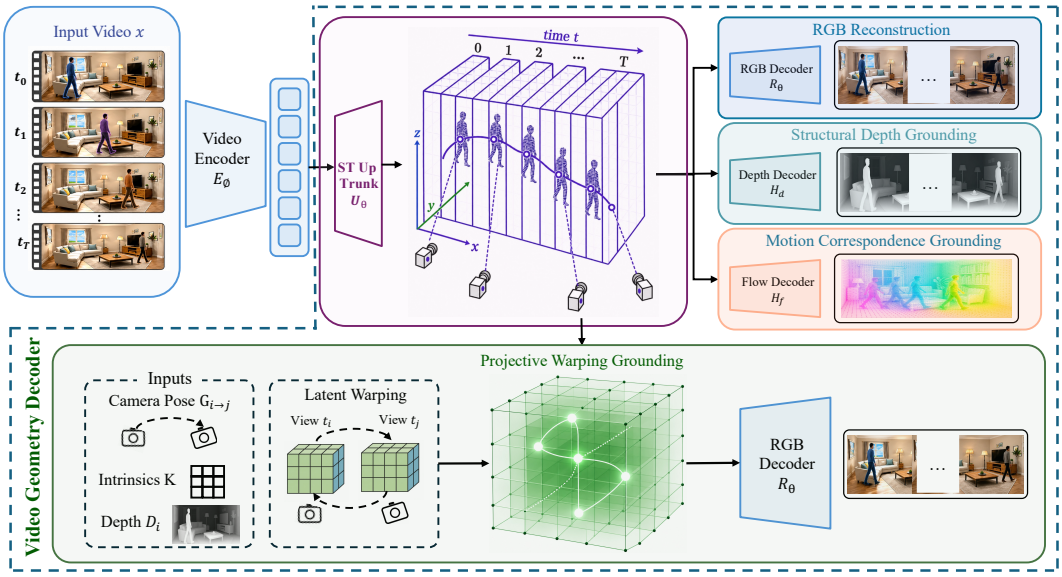


Figure 2: **Overview of VGGA.** Our VGGA introduces a Video Geometry Decoder to ground compressed video latents in dynamic 3D geometry. A shared Spatio-Temporal Upsampling Trunk (ST UP Trunk) lifts the latent to a frame-aligned decoded representation, on which RGB reconstruction jointly acts with three geometry grounding tasks, including structural depth, motion correspondence, and projective warping.

Video Latent Autoencoding. A video VAE (Kingma & Welling, 2014) compresses an input video $\mathbf{x} \in \mathbb{R}^{(1+T)\times H\times W\times 3}$ into a compact spatio-temporal latent $\mathbf{z} \in \mathbb{R}^{(1+T/r_t)\times H/r_s\times W/r_s\times C}$, where r_t and r_s denote the temporal and spatial downsampling ratios and C is the latent channel dimension. The decoder then reconstructs the latent back to RGB space. Specifically, the encoder E_ϕ parameterizes a posterior distribution

$$q_\phi(\mathbf{z} \mid \mathbf{x}) = \mathcal{N}\big(\boldsymbol{\mu}_\phi(\mathbf{x}), \mathrm{diag}(\boldsymbol{\sigma}_\phi^2(\mathbf{x}))\big), \quad \mathbf{z} = \boldsymbol{\mu}_\phi(\mathbf{x}) + \boldsymbol{\sigma}_\phi(\mathbf{x}) \odot \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \tag{1}$$

The decoder Dec_θ reconstructs the video as $\hat{\mathbf{x}} = \mathrm{Dec}_\theta(\mathbf{z})$. The autoencoder is optimized with pixel reconstruction, perceptual, adversarial, and KL regularization losses:

$$\mathcal{L}_{\mathrm{VAE}} = \lambda_{\mathrm{rec}}\|\hat{\mathbf{x}}-\mathbf{x}\|_1 + \lambda_{\mathrm{lpips}}\mathcal{L}_{\mathrm{LPIPS}}(\hat{\mathbf{x}}, \mathbf{x}) + \lambda_{\mathrm{gan}}\mathcal{L}_{\mathrm{GAN}}(\hat{\mathbf{x}}, \mathbf{x}) + \beta\, D_{\mathrm{KL}}\big(q_\phi(\mathbf{z} \mid \mathbf{x}) \,\|\, \mathcal{N}(\mathbf{0}, \mathbf{I})\big) \tag{2}$$

where the first three terms encourage faithful and perceptually realistic video reconstruction, and the KL term regularizes the latent distribution. This standard objective defines a compact reconstructive latent space, but does not explicitly constrain what geometric structure the latent should preserve.

Video Geometry Formation. A video can be viewed as the 2D projection of a dynamic 3D scene. This formation process suggests three natural levels of geometric grounding. First, each frame is tied to an underlying 3D layout, which we represent by a depth map D_i for frame I_i. Second, as the scene and camera evolve, visible points induce dense correspondence across frames, represented by optical flow $F_{i\to j}$, where a pixel $\mathbf{p}$ in frame i is matched to $\mathbf{p}' = \mathbf{p} + F_{i\to j}(\mathbf{p})$ in frame j. Third, for static scene points, depth and camera poses define a rigid projective mapping between views. This mapping accounts for camera motion, but not independent object motion. To reconstruct a target frame I_j from a source frame I_i, we use target-driven backward warping. For simplicity, let $\mathbf{K}$ denote shared camera intrinsics, and let $\mathbf{G}_{j\to i} = [\mathbf{R}_{j\to i}|\mathbf{t}_{j\to i}]$ transform points from the target camera coordinates to the source camera coordinates. A target pixel $\mathbf{p}_j$ with homogeneous coordinate $\tilde{\mathbf{p}}_j$ and depth $D_j(\mathbf{p}_j)$ is back-projected as

$$\mathbf{X}_j(\mathbf{p}_j) = D_j(\mathbf{p}_j)\mathbf{K}^{-1}\tilde{\mathbf{p}}_j \tag{3}$$

The corresponding point in the source camera coordinates and its projected sampling location are

$$\mathbf{X}_i = \mathbf{R}_{j\to i}\mathbf{X}_j(\mathbf{p}_j) + \mathbf{t}_{j\to i}, \qquad \mathbf{p}_i = \Pi(\mathbf{K}\mathbf{X}_i) \tag{4}$$

where $\Pi([x, y, z]^\top) = (x/z, y/z)^\top$. This defines the target-to-source coordinate map:

$$\mathbf{p}_i = \mathcal{W}_{\mathrm{view}}(\mathbf{p}_j, D_j(\mathbf{p}_j), \mathbf{G}_{j\to i}, \mathbf{K}) \tag{5}$$

Backward warping samples the source at $\mathbf{p}_i$ to populate the target location $\mathbf{p}_j$, using target-to-source coordinates for source-to-target reconstruction.

3.2 VIDEO GEOMETRY DECODER

Videos are observations of dynamic 3D worlds, and their latent representations should therefore capture the geometric structure underlying video formation. Motivated by this perspective, VGGA introduces a *Video Geometry Decoder* that explicitly incorporates geometric grounding into the decoding process, rather than treating the decoder solely as an RGB reconstruction module.

A key challenge arises from the spatio-temporal compression performed by video encoders. Given an input video, the resulting latent representation $\mathbf{z} \in \mathbb{R}^{(1+T/r_t)\times H/r_s\times W/r_s\times C}$ is compressed along both spatial and temporal dimensions. In particular, when $r_t > 1$, the latent sequence is not directly aligned with frame-level geometric signals such as depth, and projective geometry.

To bridge this mismatch, VGGA employs a shared spatio-temporal upsampling trunk U_θ that progressively decodes $\mathbf{z}$ into a frame-aligned intermediate representation:

$$\mathbf{h} = U_\theta(\mathbf{z}) = U_2^{\mathrm{st}}\big(U_1^{\mathrm{st}}(\mathbf{z})\big), \tag{6}$$

where U^{st} denotes a spatio-temporal upsampling block that increases both temporal and spatial resolution. The resulting feature representation $\mathbf{h} = \{\mathbf{h}_t\}_{t=0}^{T_h} \in \mathbb{R}^{(1+T_h)\times H_h\times W_h\times C_h}$ has a higher spatio-temporal resolution than $\mathbf{z}$. In our implementation, we set $T_h = T$ to establish frame-level temporal alignment with the input video. The shared representation $\mathbf{h}$ is subsequently used for both RGB reconstruction and geometric grounding. RGB reconstruction is performed by an RGB head, $\hat{\mathbf{x}} = R_\theta(\mathbf{h})$.

3.3 VIDEO GEOMETRY GROUNDING

To ground video latents in the geometry of dynamic 3D worlds, VGGA introduces three complementary grounding tasks. *Structural Depth Grounding* guides the latent to encode per-frame 3D layout. *Motion Correspondence Grounding* preserves dense cross-frame correspondence. *Projective Warping Grounding* further regularizes the latent with camera-induced view transformation, requiring decoded features to be consistent under projective geometry.

Structural Depth Grounding. Depth provides the most direct cue for the per-frame 3D layout of a video. We therefore use it as the first grounding signal to make the latent $\mathbf{z}$ aware of scene structure through the shared decoded feature $\mathbf{h}$. Specifically, our depth grounding requires each frame-level feature $\mathbf{h}_t$ to predict a depth map:

$$\hat{\mathbf{D}}_t = H_d(\mathbf{h}_t) \tag{7}$$

Let $\mathbf{D}$ be the target depth map. Following Video Depth Anything (Chen et al., 2025), we use a standard video depth loss $\mathcal{L}_{\mathrm{depth}}$ for supervision (see Section A.3 for details).

Motion Correspondence Grounding. Structural depth grounds the layout of individual frames, but it does not specify how scene points correspond over time. To make the latent aware of video dynamics, we further introduce *Motion Correspondence Grounding* based on dense optical flow. This task requires neighboring frame-level features to predict the displacement of visible content, encouraging $\mathbf{z}$ to retain the cross-frame correspondence needed for coherent motion. Given neighboring decoded features, the flow head predicts forward and backward flow:

$$\hat{\mathbf{F}}_{t\to t+1}, \hat{\mathbf{F}}_{t+1\to t} = H_f(\mathbf{h}_t, \mathbf{h}_{t+1}) \tag{8}$$

Let $\mathbf{F}$ be the target flow. Following RAFT (Teed & Deng, 2020), we use a simple l_1 endpoint regression loss for flow grounding $\mathcal{L}_{\mathrm{flow}}$ (see Section A.3 for details).

Projective Warping Grounding. We further enforce consistency between feature-space and RGB-space warping: features warped under a given camera transformation should decode to the RGB frame warped by the same geometry. This objective encourages the shared reconstruction features to preserve camera-aware geometric equivariance.

For a source frame I_i and target frame I_j, we construct a backward sampling map in Equation (5) using the target-frame VGGT pseudo depth $\mathbf{D}_j$ and the target-to-source camera transformation $\mathbf{G}_{j\to i}$. The same view-warping operation is applied at both feature and RGB resolutions. Let $\mathbf{D}_j^h$ and $\mathbf{K}_h$ denote the pseudo depth and camera intrinsics at the shared feature resolution, respectively. The target-aligned feature and its decoded RGB reconstruction are

$$\tilde{\mathbf{h}}_{i\to j} = \mathrm{Sample}\big(\mathbf{h}_i, \mathcal{W}_{\mathrm{view}}(\cdot, \mathbf{D}_j^h, \mathbf{G}_{j\to i}, \mathbf{K}_h)\big), \qquad \hat{I}_{i\to j}^{\mathrm{warp}} = R_\theta(\tilde{\mathbf{h}}_{i\to j}). \tag{9}$$

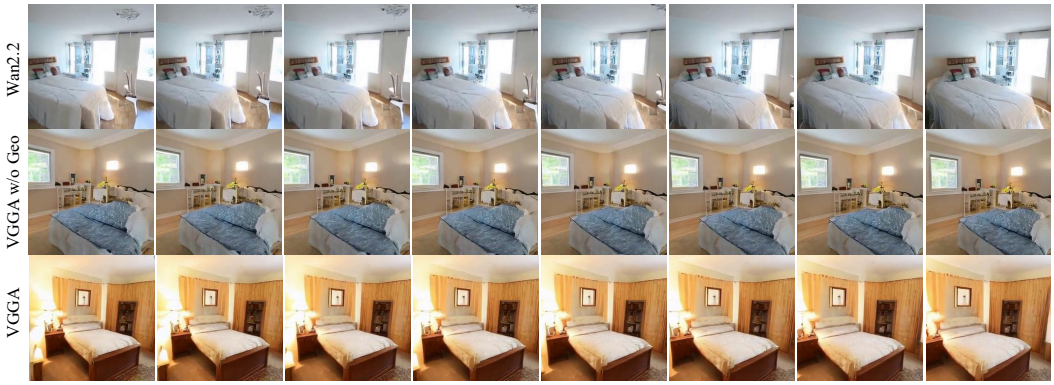


Figure 3: **Qualitative comparison on RealEstate10K.** Compared to Wan2.2 VAE and our reconstruction-only baseline (w/o Geo), VGGA synthesizes videos with more physically plausible view changes and robust structural stability. All models are evaluated under the same unconditional setting. See our video-format visualizations for better comparison.

Correspondingly, the RGB supervision target is obtained by applying the same geometric transformation at image resolution:

$$I^{\text{ref}}_{i\to j} = \text{Sample}(I_i, \mathcal{W}_{\text{view}}(\cdot, \mathbf{D}_j, \mathbf{G}_{j\to i}, \mathbf{K}))\,, \tag{10}$$

where $\mathbf{D}_j$ and $\mathbf{K}$ are defined at RGB resolution. Following reconstruction, we use a simple l_1 loss for projective warping grounding $\mathcal{L}_{\text{warp}}$ (see Section A.3 for details).

3.4 Hybrid Geometry Supervision on Unlabelled Videos

Scaling geometry-grounded video autoencoding is challenging, as large-scale video datasets rarely provide explicit geometric annotations. Dense depth and motion labels are costly to acquire, while reliable camera poses are particularly difficult to obtain for dynamic videos. VGGA therefore adopts a hybrid supervision strategy that exploits geometric cues according to their availability.

For general videos, we use pseudo depth and optical flow predicted by off-the-shelf models to supervise *Structural Depth Grounding* and *Motion Correspondence Grounding* ; sensitivity to pseudo-label quality is analyzed in Section B.

In contrast, *Projective Warping Grounding* requires reliable camera poses and assumes static scene geometry. We therefore apply this objective only to a small subset of static-scene videos with known camera poses, where projective view transformations can be reliably established. The three grounding objectives are jointly optimized with standard VAE reconstruction objective:

$$\mathcal{L}_{\text{VGGA}} = \mathcal{L}_{\text{VAE}} + \lambda_d \mathcal{L}_{\text{depth}} + \lambda_f \mathcal{L}_{\text{flow}} + \lambda_w \mathcal{L}_{\text{warp}}, \tag{11}$$

where $\mathcal{L}_{\text{warp}}$ is applied only when projective supervision is available.

Through the shared Video Geometry Decoder, camera-aware priors learned from static scenes are jointly integrated with depth and motion cues from general videos. This hybrid training enables projective geometric priors to transfer beyond the statically supervised data and benefit dynamic video representation and generation (see Figure 5(b)).

4 Experiments

4.1 Setup

Architecture. The video VAE backbone of VGGA adopts the same architectural design as Wan2.2 VAE (Wan et al., 2025). To isolate the effect of geometry grounding, both VGGA and its reconstruction-only baseline, VGGA w/o Geo, are trained from scratch on identical datasets. The Video Geometry Decoder adds lightweight geometry branches on top of the shared spatio-temporal upsampling trunk. The depth head consists of two spatial upsampling blocks, while the flow head follows the optical-flow prediction design of FlowNet2 (Ilg et al., 2017).

Table 1: **Comparison of generation performance at 17-frame 256×256 resolution.** The best and second-best results are indicated in **bold** and underlined. The notation $tTsScC$ denotes a temporal downsampling factor of T, a spatial downsampling factor of $S \times S$, and a latent channel dimension of C. "VGGA w/o Geo" denotes the reconstruction-only baseline trained without geometry grounding.

Method	Latent config	RealEstate10K			UCF101			Taichi-HD	
		FVD↓	KVD↓	Met3R↓	FVD↓	KVD↓	IS↑	FVD↓	KVD↓
CogX-VAE (Yang et al., 2024b)	t4s8c16	94.12	10.41	-	176.90	16.47	64.19	-	-
IV-VAE (Wu et al., 2025)	t4s8c16	92.37	8.35	-	175.74	22.32	64.51	-	-
WF-VAE (Li et al., 2025)	t4s8c16	107.26	12.56	-	188.19	33.01	67.49	-	-
Hunyuan-VAE (Kong et al., 2024)	t4s8c16	83.45	13.23	-	210.30	52.81	66.40	-	-
Wan2.1 VAE (Wan et al., 2025)	t4s8c16	83.84	10.64	-	167.10	**11.54**	66.04	-	-
VGGA w/o Geo	t4s8c16	88.21	8.06	-	163.21	21.35	67.36	-	-
VGGA	t4s8c16	**69.17**	**5.02**	-	**148.15**	16.21	**68.75**	-	-
Wan2.2 VAE (Wan et al., 2025)	t4s16c48	87.15	10.11	0.0494	180.79	17.80	67.32	218.92	109.58
SSVAE (Liu et al., 2025)	t4s16c48	79.08	8.79	0.0487	168.68	19.71	66.39	189.26	79.37
VGGA w/o Geo	t4s16c48	89.49	4.03	0.0514	163.78	18.72	68.14	195.44	80.34
VGGA	t4s16c48	**64.31**	**3.32**	**0.0421**	**141.27**	**16.05**	**70.32**	**121.76**	**46.89**

Table 2: **Reconstruction performance on the Kinetics-400 validation set at different resolutions.** Unlike SSVAE, which substantially degrades reconstruction quality relative to Wan2.2 VAE, VGGA preserves reconstruction fidelity close to its reconstruction-only baseline ("VGGA w/o Geo"), demonstrating the strong compatibility between geometric grounding and VAE reconstruction. Positive (green) values denote improvements, while negative (red) values denote degradations.

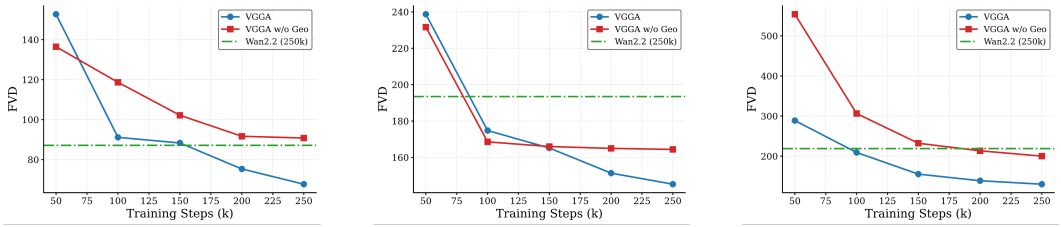
Method	$17 \times 256 \times 256$				$17 \times 512 \times 512$			
	rFVD↓	PSNR↑	SSIM↑	LPIPS↓	rFVD↓	PSNR↑	SSIM↑	LPIPS↓
CogX-VAE (Yang et al., 2024b)	4.90	33.78	0.97	0.027	1.79	36.00	0.99	0.024
IV-VAE (Wu et al., 2025)	2.78	34.08	0.96	0.019	0.97	37.24	0.96	0.016
WF-VAE-L (Li et al., 2025)	3.06	33.48	0.96	0.023	1.08	35.93	0.96	0.023
Hunyuan-VAE (Kong et al., 2024)	2.96	34.30	0.97	0.016	0.90	37.13	0.97	0.015
Wan2.1 VAE (Wan et al., 2025)	2.92	33.21	0.95	0.018	1.02	36.15	0.97	0.017
Wan2.2 VAE (Wan et al., 2025)	3.42	33.78	0.96	0.015	1.22	36.75	0.97	0.015
SSVAE (Liu et al., 2025)	7.50	31.18	0.96	0.036	2.16	34.45	0.97	0.028
Gain vs. Wan2.2 VAE	−4.08	−2.60	0.00	−0.021	−0.94	−2.30	0.00	−0.013
VGGA w/o Geo	3.56	33.61	0.97	0.019	1.34	36.65	0.97	0.017
VGGA	3.42	33.52	0.97	0.018	1.30	36.47	0.97	0.019
Gain vs. VGGA w/o Geo	+0.14	−0.09	0.00	+0.001	+0.04	−0.18	0.00	−0.002

Training Data. We train VGGA on a mixture of a curated high-quality 10M-video dataset and RealEstate10K (Zhou et al., 2018), sampled at a 4:1 ratio. The 10M videos do not require manual depth or flow annotations. We obtain depth pseudo labels using VGGT (Wang et al., 2025) and optical-flow pseudo labels using RAFT (Teed & Deng, 2020). Structural Depth Grounding and Motion Correspondence Grounding are applied to all training videos. Projective Warping Grounding is applied only to RealEstate10K samples. Additional training details are provided in Section A.

Evaluation. We evaluate generation on RealEstate10K (Zhou et al., 2018), UCF101 (Soomro et al., 2012), and Taichi-HD (Siarohin et al., 2019). RealEstate10K contains strong geometric structure and camera motion, and is used to assess view consistency and camera-motion stability. UCF101 evaluates action-conditioned generation, while Taichi-HD evaluates unconditional human motion generation. For reconstruction, we evaluate it on the Kinetics-400 validation set (Carreira & Zisserman, 2017). Additional evaluation details are provided in Section A.

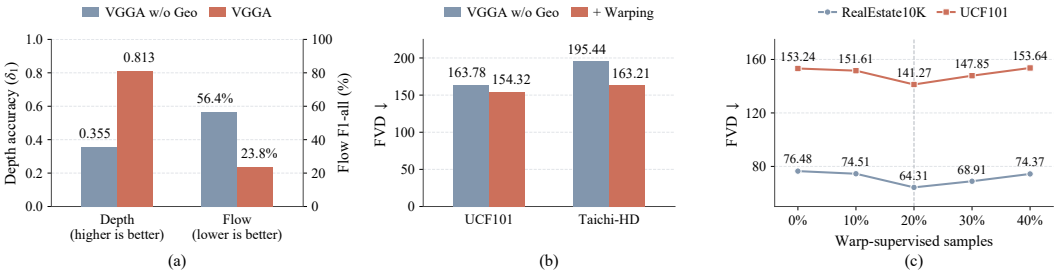
4.2 Video Generation and Reconstruction

Geometry-centric generation. We evaluate VGGA on RealEstate10K to assess its ability to model complex camera motion and projective view changes. As shown in Table 1, VGGA achieves the best FVD of 64.31, outperforming SSVAE by 14.77 points under the same $t4s16c48$ latent configuration and reducing FVD by 25.18 points over the reconstruction-only baseline. VGGA also improves geometric consistency, reducing Met3R from 0.0514 to 0.0421. Together with the qualitative results in Figure 3, these results demonstrate that geometry grounding improves both generation quality and geometric fidelity under complex view transformations.



(a) RealEstate10K (b) UCF101 (c) Taichi-HD

Figure 4: **Effect of geometry grounding on Video DiT training.** We compare VGGA with its reconstruction-only counterpart (VGGA w/o Geo) on three generation benchmarks. Geometry grounding consistently yields lower FVD throughout training, indicating more favorable optimization dynamics and improved diffusability of the resulting video latents.



(a) (b) (c)

Figure 5: **Geometric representation learning and transfer from static-scene supervision. (a) Latent geometric probing:** linear probes on frozen latents recover depth and optical flow on the RealEstate10K test split, evaluated against VGGT and RAFT pseudo-labels, respectively. **(b) Transfer to dynamic generation:** adding only *Projective Warping Grounding* on static-scene RealEstate10K ("+ Warping") improves generation on dynamic-scene datasets, including UCF101 and Taichi-HD. **(c) Warp-supervision proportion:** a 20% proportion of warp-supervised samples (our default, dashed line) achieves the lowest FVD on both datasets.

Action-conditioned generation. Beyond camera-centric scenarios, we evaluate whether geometry-grounded latents support diverse human actions. As reported in Table 1, VGGA achieves an FVD of 141.27 and an Inception Score (IS) of 70.32 on UCF101. Under the t4s16c48 configuration, VGGA reduces FVD by 27.41 points relative to SSVAE (168.68) and by 22.51 points relative to its reconstruction-only counterpart (163.78). Under t4s8c16, VGGA likewise improves FVD from 163.21 to 148.15. These results demonstrate that geometry grounding benefits action-conditioned synthesis across both latent configurations.

Human-motion generation. Furthermore, we extend our evaluation to Taichi-HD to test the model's capability in handling articulated human motion, which demands that the latent distribution encapsulate both fine-grained local deformations and global temporal coherence. As shown in Table 1, VGGA reduces FVD from 218.92 (Wan2.2 VAE) to 121.76 under the same t4s16c48 configuration. Compared to its reconstruction-only counterpart, VGGA reduces FVD by 73.68 points (195.44 to 121.76). These results substantiate the effectiveness of VGGA across different video generation tasks.

Video reconstruction. As shown in Table 2, VGGA maintains reconstruction quality close to its reconstruction-only baseline across resolutions. At $17 \times 256 \times 256$, rFVD improves from 3.56 to 3.42 and LPIPS from 0.019 to 0.018, with only a 0.09 dB drop in PSNR. At $17 \times 512 \times 512$, rFVD improves from 1.34 to 1.30, while PSNR decreases by 0.18 dB and LPIPS increases slightly from 0.017 to 0.019. SSIM remains unchanged at 0.97. In contrast, SSVAE incurs PSNR drops of 2.60 and 2.30 dB relative to Wan2.2 VAE. These results show that VGGA substantially preserves reconstruction fidelity, demonstrating the compatibility of geometric grounding with VAE reconstruction.

4.3 Analysis

We analyze how geometry grounding affects latent diffusability, the recoverability of geometric information, and the transfer of static-scene supervision to dynamic video generation.

Diffusability of video latents. We evaluate VAE's diffusability by tracking FVD during the training of identical Latte-XL/1 generators. As shown in Figure 4, VGGA consistently achieves lower FVD

Table 3: Ablation on geometry grounding.

Components	Generation		Reconstruction	
	FVD↓	KVD↓	rFVD↓	PSNR↑
VGGA w/o Geo	163.78	18.72	3.56	33.61
w/o Depth Grounding	156.31	18.10	**3.38**	33.50
w/o Motion Grounding	146.43	16.96	3.50	**33.63**
w/o Warping Grounding	152.24	17.78	3.58	33.58
VGGA full	**141.27**	**16.05**	3.42	33.52

Table 4: Ablation on geometry branch position.

Components	Frame-aligned	Generation		Reconstruction	
		FVD↓	KVD↓	rFVD↓	PSNR↑
Separate branch	✗	159.12	18.80	3.56	**33.60**
after U_1^{st}	✗	149.27	18.15	3.48	33.31
after U_2^{st} (VGGA)	✓	**141.27**	**16.05**	**3.42**	33.52
after U_3^{s}	✓	153.46	18.66	4.31	32.14
after U_4^{s}	✓	-	-	-	12.08

and faster convergence than the reconstruction-only baseline. This suggests that geometry grounding regularizes the latent space with explicit structural cues, thereby reducing the complexity of the downstream generative task and rendering the latents more diffusion-friendly.

Latent geometric probing. We train linear depth and optical-flow probes on frozen video latents and evaluate them on the RealEstate10K test split, using VGGT and RAFT pseudo-labels as reference targets. As shown in Figure 5(a), VGGA improves depth δ_1 from 0.355 to 0.813 and reduces the flow outlier rate (F1-all) from 56.4% to 23.8%. These results demonstrate that VGGA encodes substantially stronger geometric structure and motion correspondence, making such information more linearly accessible from the latent space.

Transfer from static scenes to dynamic generation. To isolate the contribution of static-scene supervision, we add only the projective warping objective on RealEstate10K to the reconstruction-only baseline. As shown in Figure 5(b), FVD decreases from 163.78 to 154.32 on UCF101 and from 195.44 to 163.21 on Taichi-HD. These results demonstrate that camera-aware geometric priors learned from static scenes transfer effectively to dynamic video generation and provide complementary benefits to depth and motion supervision.

4.4 Ablation Study

Geometry grounding signals. The three grounding signals provide complementary constraints on scene structure, temporal correspondence, and camera-induced geometry. As shown in Table 3, removing geometric supervision increases FVD from 141.27 to 163.78, while all leave-one-out variants degrade generation performance. This confirms that the three geometric cues contribute complementarily to learning generation-friendly video latents.

Shared branching position. As shown in Table 4, branching after U_2^{st} achieves the best overall performance. Earlier branching lacks sufficient frame-level alignment, whereas later branching operates on increasingly RGB-specific features. A separate geometry branch is also less effective, confirming that geometric supervision is most effective when applied to the shared decoded representation.

Proportion of warp-supervised samples. Figure 5(c) studies the proportion of samples receiving projective warping supervision. Increasing this proportion from 0% to 20% reduces FVD from 76.48 to 64.31 on RealEstate10K and from 153.24 to 141.27 on UCF101. Further increasing the proportion to 30% or 40% degrades performance on both datasets. Thus, the default 20% mixture provides the best trade-off, introducing effective camera-aware supervision while retaining general videos as the majority of training data.

5 Conclusion

We presented **VGGA**, which introduces a geometry-aware perspective for learning generation-friendly video latents by grounding autoencoding in the geometric structure of dynamic 3D scenes. VGGA enables geometric learning to be effectively integrated with reconstruction and scaled to large-scale unlabelled videos. Extensive experiments demonstrate that geometry-grounded autoencoding substantially improves downstream video generation while preserving reconstruction quality.

Limitations. VGGA focuses on frame-level and short-range geometric consistency, without explicitly modeling long-horizon dynamics or full 3D scene structure. Extending geometry grounding to longer temporal ranges and scene-level 3D modeling is a promising direction for future work.

USE OF LARGE LANGUAGE MODELS (LLMS)

In preparing this manuscript, we used large language models solely as a lightweight writing aid for grammar, wording, and formatting suggestions. The models were *not* used to generate research ideas, design algorithms, write code, run experiments, analyze data, or draft scientific content. All technical claims, methods, and conclusions were conceived, produced, and verified by the authors. Suggested edits from LLMs were manually reviewed and integrated at the authors' discretion. And we accept full responsibility for the accuracy and integrity of the manuscript, including ensuring that no plagiarized or misrepresented content from a LLM is included.

REFERENCES

Alessandro Achille and Stefano Soatto. Emergence of invariance and disentanglement in deep representations. *Journal of Machine Learning Research*, 19(50):1–34, 2018.

Mohammad Asim, Christopher Wewer, Thomas Wimmer, Bernt Schiele, and Jan Eric Lenssen. MET3R: Measuring multi-view consistency in generated images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6034–6044, 2025.

Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6299–6308, 2017.

Sili Chen, Hengkai Guo, Shengnan Zhu, Feihu Zhang, Zilong Huang, Jiashi Feng, and Bingyi Kang. Video depth anything: Consistent depth estimation for super-long videos. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 22831–22840, 2025.

Yu Gao, Haoyuan Guo, Tuyen Hoang, Weilin Huang, Lu Jiang, Fangyuan Kong, Huixia Li, Jiashi Li, Liang Li, Xiaojie Li, et al. Seedance 1.0: Exploring the boundaries of video generation models. *arXiv preprint arXiv:2506.09113*, 2025.

Quan Huynh-Thu and Mohammed Ghanbari. Scope of validity of psnr in image/video quality assessment. *Electronics Letters*, 44(13):800–801, 2008.

Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2462–2470, 2017.

Nikita Karaev, Ignacio Rocco, Benjamin Graham, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. Cotracker: It is better to track together. In *European conference on computer vision*, pp. 18–35. Springer, 2024.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *ICLR*, 2014.

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.

Theodoros Kouzelis, Ioannis Kakogeorgiou, Spyros Gidaris, and Nikos Komodakis. Eq-vae: Equivariance regularized latent space for improved generative image modeling. *arXiv preprint arXiv:2502.09509*, 2025.

Xingjian Leng, Jaskirat Singh, Yunzhong Hou, Zhenchang Xing, Saining Xie, and Liang Zheng. REPA-E: Unlocking VAE for end-to-end tuning of latent diffusion transformers. In *ICCV*, 2025.

Tianhong Li, Lijie Fan, Yuan Yuan, Hao He, Yonglong Tian, Rogerio Feris, Piotr Indyk, and Dina Katabi. Addressing feature suppression in unsupervised visual representations. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 1411–1420, 2023.

Zongjian Li, Bin Lin, Yang Ye, Liuhan Chen, Xinhua Cheng, Shenghai Yuan, and Li Yuan. Wf-vae: Enhancing video vae by wavelet-driven energy flow for latent video diffusion model. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 17778–17788, 2025.

Bin Lin, Yunyang Ge, Xinhua Cheng, Zongjian Li, Bin Zhu, Shaodong Wang, Xianyi He, Yang Ye, Shenghai Yuan, Liuhan Chen, et al. Open-sora plan: Open-source large video generation model. *arXiv preprint arXiv:2412.00131*, 2024.

Shizhan Liu, Xinran Deng, Zhuoyi Yang, Jiayan Teng, Xiaotao Gu, and Jie Tang. Delving into latent spectral biasing of video vaes for superior diffusability. *arXiv preprint arXiv:2512.05394*, 2025.

Chuofan Ma, Yi Jiang, Junfeng Wu, Jihan Yang, Xin Yu, Zehuan Yuan, Bingyue Peng, and Xiaojuan Qi. UniTok: A unified tokenizer for visual generation and understanding. In *NeurIPS*, 2025.

Nanye Ma, Mark Goldstein, Michael S Albergo, Nicholas M Boffi, Eric Vanden-Eijnden, and Saining Xie. Sit: Exploring flow and diffusion-based generative models with scalable interpolant transformers. In *European Conference on Computer Vision*, pp. 23–40. Springer, 2024a.

Xin Ma, Yaohui Wang, Xinyuan Chen, Gengyun Jia, Ziwei Liu, Yuan-Fang Li, Cunjian Chen, and Yu Qiao. Latte: Latent diffusion transformer for video generation. *arXiv preprint arXiv:2401.03048*, 2024b.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *ICCV*, 2023.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022.

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *Advances in Neural Information Processing Systems*, 2016.

Team Seedance, De Chen, Liyang Chen, Xin Chen, Ying Chen, Zhuo Chen, Zhuowei Chen, Feng Cheng, Tianheng Cheng, Yufeng Cheng, et al. Seedance 2.0: Advancing video generation for world complexity. *arXiv preprint arXiv:2604.14148*, 2026.

Minglei Shi, Haolin Wang, Wenzhao Zheng, Ziyang Yuan, Xiaoshi Wu, Xintao Wang, Pengfei Wan, Jie Zhou, and Jiwen Lu. Latent diffusion model without variational autoencoder. *arXiv preprint arXiv:2510.15301*, 2025.

Aliaksandr Siarohin, Stéphane Lathuilière, Sergey Tulyakov, Elisa Ricci, and Nicu Sebe. First order motion model for image animation. *Advances in neural information processing systems*, 32, 2019.

Ivan Skorokhodov, Sharath Girish, Benran Hu, Willi Menapace, Yanyu Li, Rameen Abdal, Sergey Tulyakov, and Aliaksandr Siarohin. Improving the diffusability of autoencoders. *arXiv preprint arXiv:2502.14831*, 2025.

Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012.

Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *European conference on computer vision*, pp. 402–419. Springer, 2020.

Yonglong Tian, Chen Sun, Ben Poole, Dilip Krishnan, Cordelia Schmid, and Phillip Isola. What makes for good views for contrastive learning? *Advances in neural information processing systems*, 33:6827–6839, 2020.

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

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 *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 5294–5306, 2025.

Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 20697–20709, 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.

Pingyu Wu, Kai Zhu, Yu Liu, Liming Zhao, Wei Zhai, Yang Cao, and Zheng-Jun Zha. Improved video vae for latent video diffusion model. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 18124–18133, 2025.

Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything v2. *Advances in Neural Information Processing Systems*, 37:21875–21911, 2024a.

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

Jingfeng Yao, Yuda Song, Yucong Zhou, and Xinggang Wang. Towards scalable pre-training of visual tokenizers for generation. *arXiv preprint arXiv:2512.13687*, 2025a.

Jingfeng Yao, Bin Yang, and Xinggang Wang. Reconstruction vs. generation: Taming optimization dilemma in latent diffusion models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 15703–15712, 2025b.

Lijun Yu, José Lezama, Nitesh B. Gundavarapu, Luca Versari, Kihyuk Sohn, David Minnen, Yong Cheng, Agrim Gupta, Xiuye Gu, Alexander G. Hauptmann, Boqing Gong, Ming-Hsuan Yang, Irfan Essa, David A. Ross, and Lu Jiang. Language model beats diffusion–tokenizer is key to visual generation. In *ICLR*, 2024.

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 *ICLR*, 2025.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 586–595, 2018.

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.

Appendix

A Implementation Details

This section provides additional implementation details omitted from the main paper for clarity. We describe the evaluation metrics, pseudo-label generation for geometry grounding, autoencoder training losses, and downstream latent generator training protocol.

A.1 Evaluation Metrics

For generation, we follow common video-generation evaluation protocols and compute Fréchet Video Distance (FVD) and Kernel Video Distance (KVD) (Unterthiner et al., 2018) in the I3D feature space. FVD measures the Fréchet distance between Gaussian-fitted real and generated video features, while KVD computes a kernel two-sample distance between the two feature sets. We additionally report Inception Score (IS) (Salimans et al., 2016) on datasets where the metric is applicable. Unless otherwise specified, generation metrics are computed with 2048 generated videos at $17 \times 256 \times 256$ resolution.

For geometric consistency, we follow Met3R (Asim et al., 2025) and evaluate 3D consistency between adjacent frames of generated videos on RealEstate10K to obtain video-level Met3R scores. Lower scores indicate better geometric consistency.

We evaluate reconstruction using rFVD, peak signal-to-noise ratio (PSNR) (Huynh-Thu & Ghanbari, 2008), structural similarity (SSIM) (Wang et al., 2004), and learned perceptual image patch similarity (LPIPS) (Zhang et al., 2018). Reconstruction FVD (rFVD) is computed between original and reconstructed video clips using the same feature extractor as FVD. PSNR, SSIM, and LPIPS are computed frame-wise and averaged over all frames and validation videos.

A.2 Pseudo Geometric Supervision

VGGA does not require manual depth or optical-flow annotations. For the curated 10M-video dataset and RealEstate10K, we generate pseudo depth maps using VGGT (Wang et al., 2025) and pseudo optical flow using RAFT (Teed & Deng, 2020). Depth and flow pseudo labels are produced offline and resized to the training resolution. We mix the curated 10M-video dataset and RealEstate10K with a 4:1 sampling ratio. For projective warping, we use RealEstate10K samples with valid camera poses and predominantly static content. Target-frame VGGT pseudo depth and target-to-source camera transformations define the backward sampling maps; depth maps and intrinsics are resized and scaled to the decoded feature grid, respectively. During mixed-data training, Structural Depth Grounding and Motion Correspondence Grounding are applied to all samples, while Projective Warping Grounding is applied only to RealEstate10K samples with valid camera poses. Sensitivity to alternative depth and flow teachers is evaluated in Section B.

A.3 Autoencoder Training Losses

Structural Depth Grounding Loss. For depth regression, we transform the pseudo depth to $\mathbf{D}' = \log(1 + \mathbf{D})$ element-wise and denote the corresponding prediction in this domain by $\hat{\mathbf{D}}'$. Following Video Depth Anything (Chen et al., 2025), we use a video depth loss $\mathcal{L}_{\text{depth}}$ that combines depth regression, spatial gradient matching, and temporal consistency:

$$\mathcal{L}_{\text{depth}} = \frac{1}{N} \left\| \hat{\mathbf{D}}' - \mathbf{D}' \right\|_1 + \frac{\alpha}{N} \left\| \nabla_{xy} \hat{\mathbf{D}}' - \nabla_{xy} \mathbf{D}' \right\|_1 + \frac{\gamma}{N} \left\| \nabla_t \hat{\mathbf{D}}' - \nabla_t \mathbf{D}' \right\|_1 \tag{12}$$

Here ∇_{xy} and ∇_t denote spatial and temporal depth gradients, N denotes the number of video pixels, and α, γ balance edge preservation and temporal consistency. This transformation is used for depth supervision only; projective warping uses the original pseudo depth $\mathbf{D}$ for geometric back-projection.

Motion Correspondence Grounding Loss. Let $\mathbf{F}$ be the target flow in pixel units at resolution $H \times W$. We scale the displacement as $\mathbf{F}' = \mathbf{F}/\sqrt{\max(H, W)}$ and denote the corresponding prediction by $\hat{\mathbf{F}}'$. Following RAFT (Teed & Deng, 2020), we use an L_1 regression loss for flow

grounding:

$$\mathcal{L}_{\text{flow}} = \frac{1}{N} \left\| \hat{\mathbf{F}}' - \mathbf{F}' \right\|_1 \tag{13}$$

When backward flow is available, the same loss is applied in the reverse direction.

Projective Warping Grounding Loss. For each supervised pair, a projection-validity mask $\mathbf{M}_{i\to j}$ is defined on the RGB reconstruction grid. A target pixel is retained when its back-projected point has positive depth after transformation into the source camera and projects within the source image: $0 \le u \le W-1$ and $0 \le v \le H-1$. For the set of supervised frame pairs $\mathcal{P}$, the loss is

$$\mathcal{L}_{\text{warp}} = \frac{1}{N_w} \sum_{(i,j)\in\mathcal{P}} \left\| \mathbf{M}_{i\to j} \odot (\hat{I}^{\text{warp}}_{i\to j} - I^{\text{ref}}_{i\to j}) \right\|_1 \tag{14}$$

Loss scaling and joint optimization. The dense supervision signals differ substantially in scale and physical units, including RGB intensities, depth values, and pixel displacements. Before combining their losses, we apply task-specific normalization and average each objective over its valid supervised pixels, as summarized in Table 5. Specifically, RGB values are normalized to a bounded range, optical flow is scaled by image resolution, and depth is log-transformed to reduce its dynamic range. These normalizations bring the dense objectives to comparable numerical scales of approximately $O(1)$, without explicitly balancing their gradient magnitudes.

Table 5: Formulation and normalization of the dense training objectives. Normalization is applied before the loss weights in Table 6.

Objective	Formulation	Scale and normalization
$\mathcal{L}_{\text{rec}}$	L_1 reconstruction	Normalize RGB values to $[-1, 1]$ and average the reconstruction error over video pixels.
$\mathcal{L}_{\text{warp}}$	Masked L_1	Use RGB values in $[-1, 1]$ for reconstruction and the warped-source reference; sum errors over valid pixels and divide by their count N_w.
$\mathcal{L}_{\text{flow}}$	L_1 flow regression	Scale pixel displacements as $\mathbf{F}' = \mathbf{F}/\sqrt{\max(H, W)}$; average regression errors in this domain over supervised pixels.
$\mathcal{L}_{\text{depth}}$	L_1 regression with spatial and temporal gradient losses	Transform pseudo depth as $\mathbf{D}' = \log(1 + \mathbf{D})$; compute regression and gradient losses in this domain, each normalized by N.

Here H and W are the training image height and width, N is the number of supervised video pixels, and N_w is the number of valid pixels across the warp-supervised frame pairs. The warping term is zero when no valid pixels are available. Depth normalization applies only to the depth loss; geometric back-projection uses the original pseudo depth.

Total training loss. The video autoencoder is optimized with RGB reconstruction, perceptual, adversarial, KL, and geometry grounding losses. The full loss is

$$\begin{aligned} \mathcal{L} =& \lambda_{\text{rec}}\mathcal{L}_{\text{rec}} + \lambda_{\text{lpips}}\mathcal{L}_{\text{LPIPS}} + \lambda_{\text{gan}}\mathcal{L}_{\text{3D-GAN}} + \beta\mathcal{L}_{\text{KL}} \\ &+ \lambda_d\mathcal{L}_{\text{depth}} + \lambda_f\mathcal{L}_{\text{flow}} + \lambda_w a(\mathbf{x})\mathcal{L}_{\text{warp}} \end{aligned} \tag{15}$$

Here $\mathcal{L}_{\text{rec}}$ is the pixel-level L1 reconstruction loss, $\mathcal{L}_{\text{LPIPS}}$ is the perceptual loss, and $\mathcal{L}_{\text{3D-GAN}}$ is a spatio-temporal adversarial loss implemented with a 3D discriminator. The discriminator receives real and reconstructed video clips and is enabled after a warm-up stage, so the autoencoder first learns a stable reconstruction mapping before adversarial training sharpens spatio-temporal details. The activation indicator $a(\mathbf{x})$ is one for RealEstate10K clips with valid camera poses and zero otherwise, following the supervision scope in Section 3.4.

Loss weight. We use unit weights for RGB reconstruction, LPIPS, and all three geometry terms, with a GAN weight of 0.1 and a KL weight of 10^{-8} (Table 6). The 4:1 data mixture activates warping on the 20% RealEstate10K portion, while RGB, depth, and flow objectives supervise both data sources. Both VGGA and VGGA w/o Geo use the same KL weight.

Table 6: Loss weights and training schedule used for VGGA.

Term	Weight or setting
RGB reconstruction λ_{rec}	1.0
LPIPS λ_{lpips}	1.0
KL regularization β	1e-8
3D GAN λ_{gan}	0.1
GAN start step	60k
Structural depth grounding λ_d	1.0
Motion correspondence grounding λ_f	1.0
Projective warping grounding λ_w	1.0
Depth spatial-gradient weight α	0.5
Depth temporal-gradient weight γ	0.5
Training steps	100K
Learning rate	5×10^{-5}
GPUs	16 H100

We train VGGA on mixed-resolution clips, including $33 \times 256 \times 256$ and $9 \times 512 \times 512$ videos sampled at a 2:1 ratio. Unless otherwise specified, the same optimizer, batching strategy, and latent configuration are used for VGGA and its reconstruction-only counterpart, so the effect of geometry grounding can be isolated from changes in model capacity or training budget.

A.4 Downstream Latent Generator

For downstream video generation, each autoencoder is frozen and a Latte-XL/1 latent video generator (Ma et al., 2024b) is trained on its latents. We use velocity prediction as the diffusion prediction target. All autoencoders are evaluated with the same generator architecture, training data, and optimization strategy. The latent generator is trained for 250K steps on 16 H100 GPUs, and samples are decoded by the corresponding frozen autoencoder for evaluation.

B Sensitivity to Pseudo Labels

We assess sensitivity to pseudo-label teachers by varying the depth or flow estimator while keeping the other teacher and all remaining training settings fixed.

B.1 Sensitivity to the Depth Teacher

We replace VGGT with Depth Anything V2 (Yang et al., 2024a) or Video Depth Anything (Chen et al., 2025), keeping RAFT as the flow teacher. As shown in Table 7, FVD changes by at most 0.47 on RealEstate10K and 1.57 on UCF101 relative to VGGT, while PSNR differs by at most 0.07 dB. These results demonstrate the robustness of VGGA to different depth teachers.

Table 7: Sensitivity to the depth teacher. The flow teacher is fixed to RAFT.

Depth teacher	FVD↓		PSNR↑
	RealEstate10K	UCF101	
VGGT (default)	64.31	141.27	33.61
Depth Anything V2	64.78	142.84	33.67
Video Depth Anything	64.05	140.22	33.54

B.2 Sensitivity to the Flow Teacher

We replace RAFT with FlowNet2 (Ilg et al., 2017), keeping VGGT as the depth teacher. As shown in Table 8, this replacement changes RealEstate10K and UCF101 FVD by -0.53 and $+0.38$, respectively, and decreases PSNR by 0.39 dB. Generation performance therefore remains similar under the tested flow-teacher replacement.

Table 8: Sensitivity to the flow teacher. The depth teacher is fixed to VGGT.

Flow teacher	FVD↓		PSNR↑
	RealEstate10K	UCF101	
RAFT (default)	64.31	141.27	33.61
FlowNet2	63.78	141.65	33.22

Together, these comparisons support robustness of our VGGA to the pseudo-label teachers.

C Visualizations of VGGA Latents

To intuitively understand the impact of geometry grounding on the latent space, we visualize the latent features using Principal Component Analysis (PCA) as shown in Figure 6. Compared to the reconstruction-only baseline, which often results in a scattered or entangled latent distribution, the latent space learned by VGGA exhibits a more compact and semantically organized structure. This clustering behavior suggests that our geometric constraints effectively filter out stochastic noise and redundant textural information, instead prioritizing consistent structural primitives. Despite this higher degree of abstraction, VGGA maintains sufficient expressive power to preserve fine-grained visual details, as evidenced by the high-fidelity reconstructions. These visualizations provide further evidence that geometry grounding yields a more regularized and representative latent manifold for video modeling.



Figure 6: **Latent Visualization.** Compared with the reconstruction-only baseline, VGGA exhibits a more compact and structured video latent space while retaining sufficient fine-grained visual details for faithful reconstruction.

D More Visualization Results of Video Generation

We provide additional qualitative generation results on the three evaluation datasets used in the main paper. As shown in Figures 7 to 9, these visualizations complement the quantitative metrics by showing the effect of VGGA on camera consistency, action dynamics, and human-motion synthesis.

E Geometry Grounding Evaluation

E.1 Depth and Flow Visualization

We provide additional qualitative results for the geometry grounding branches of VGGA. These results (see Figures 10 to 12) verify that the Video Geometry Decoder learns meaningful depth and motion representations, rather than using the auxiliary branches only as weak training regularizers.

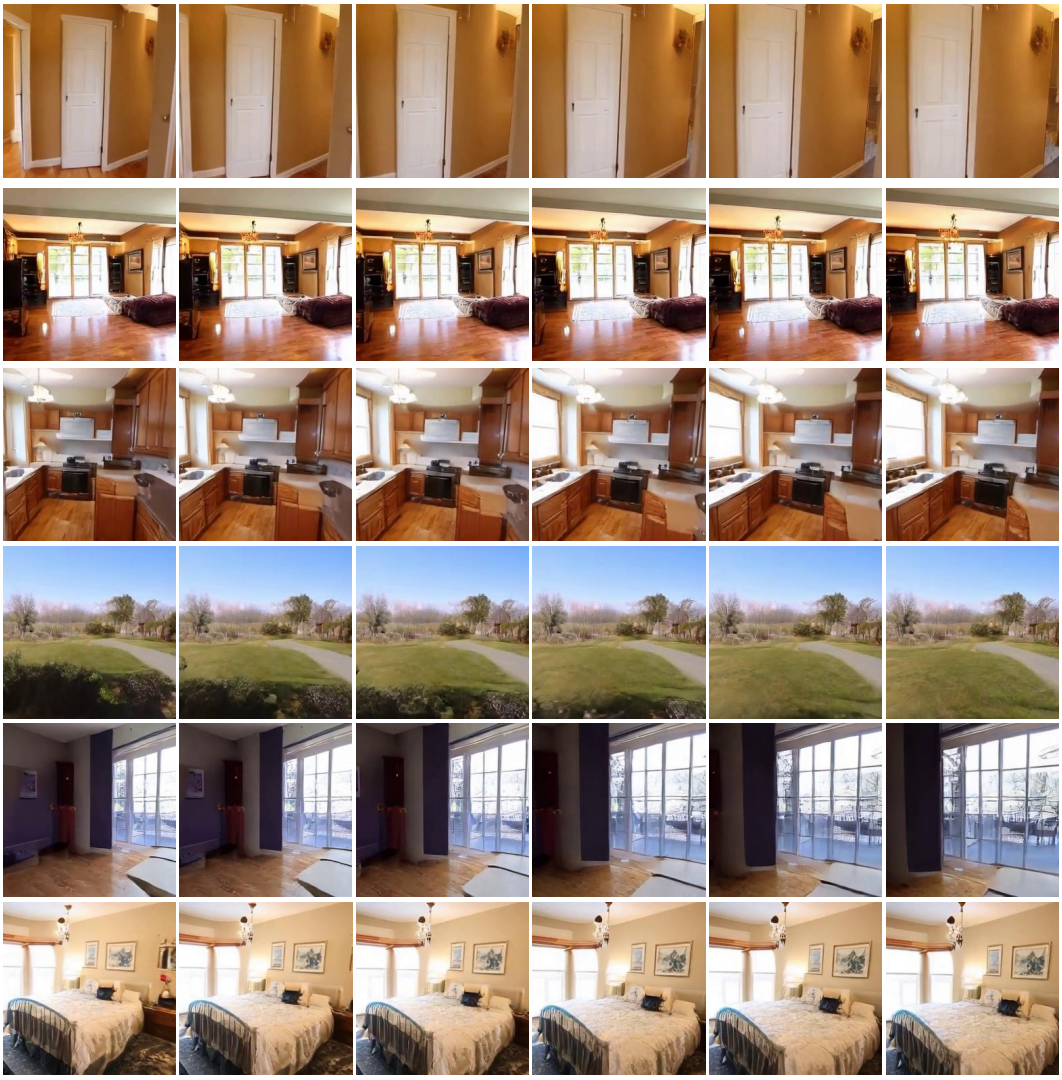


Figure 7: **Generation visualization of VGGA on RealEstate10K.**

Discussion. The qualitative results show that VGGA recovers coherent depth and flow from the shared decoded representation, which demonstrating that dense geometric supervision can shape video latents with scene-layout and motion-correspondence information while remaining compatible with RGB reconstruction.

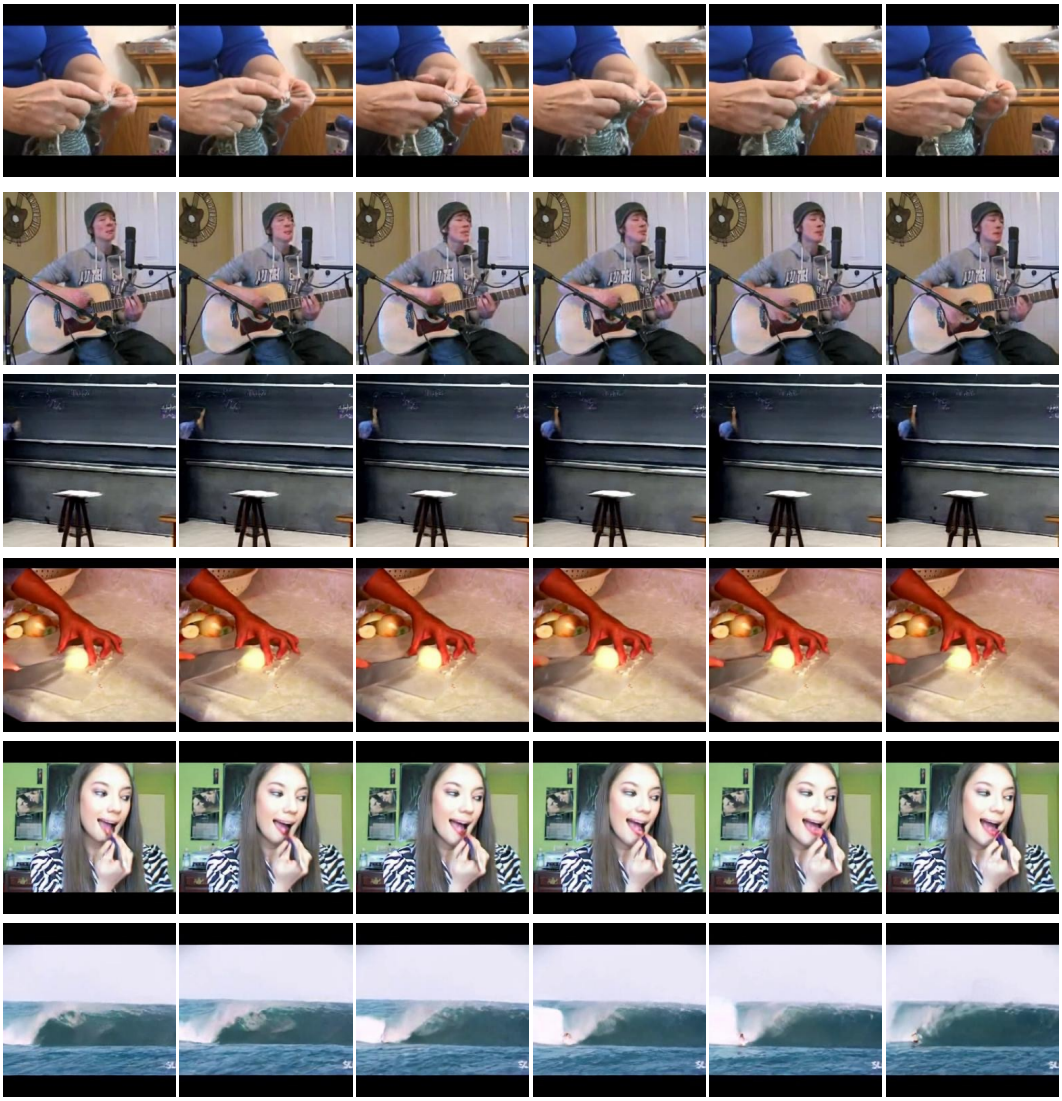


Figure 8: **Generation visualization of VGGA on UCF101.**

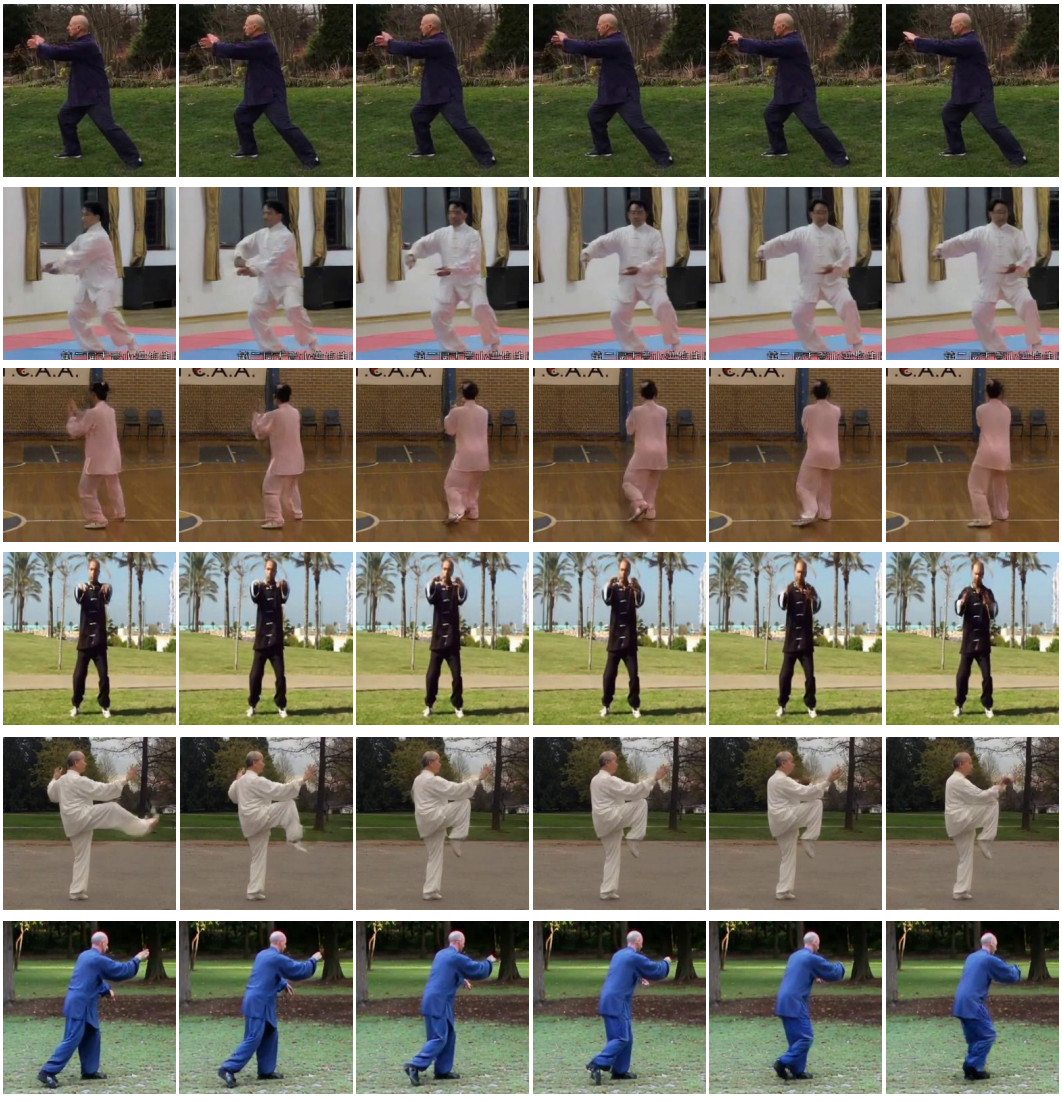


Figure 9: **Generation visualization of VGGA on Taichi-HD.**

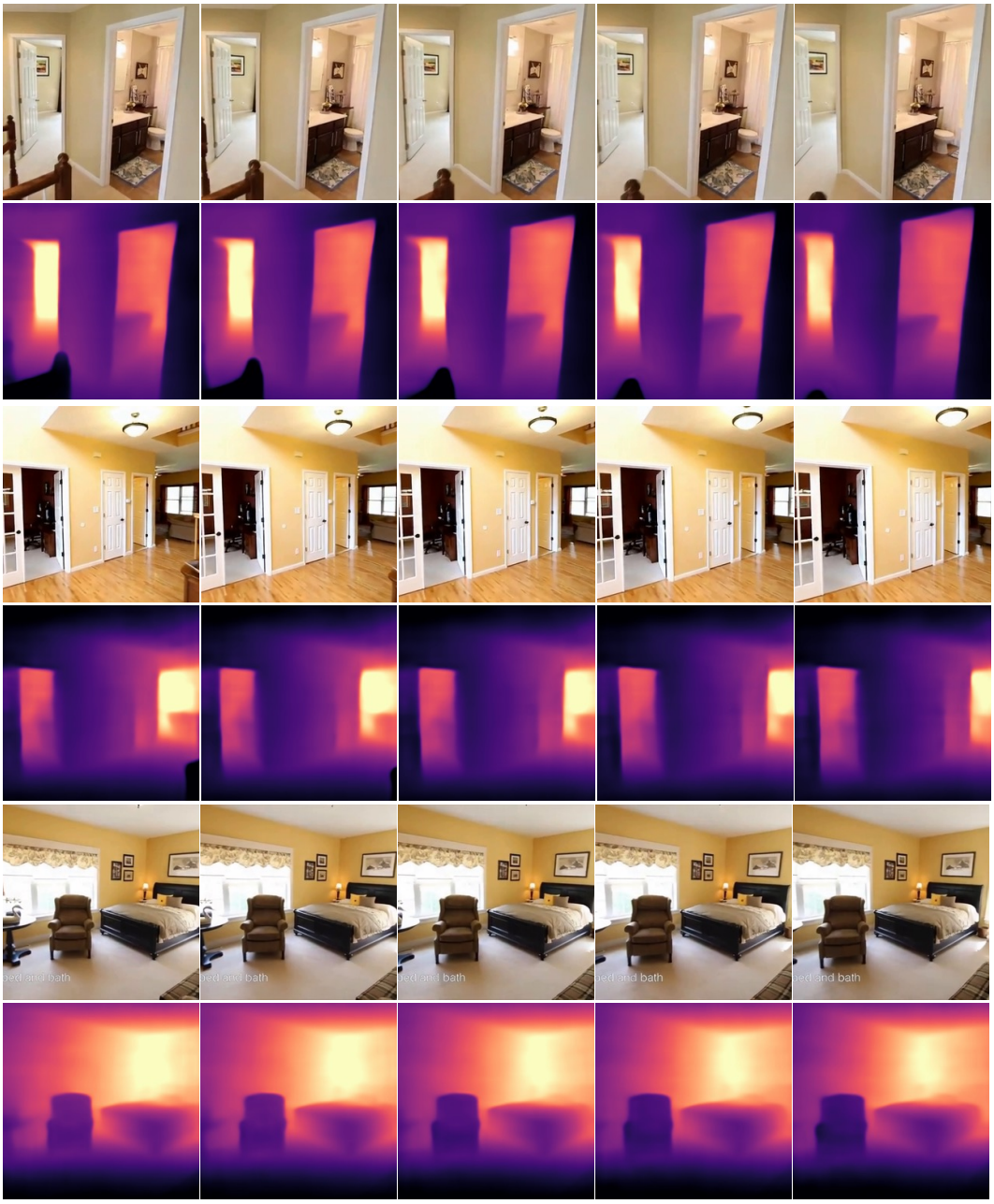


Figure 10: **Visualization of structural depth grounding on dataset RealEstate10K.** VGGA predicts temporally coherent depth maps that capture scene layout and object boundaries.

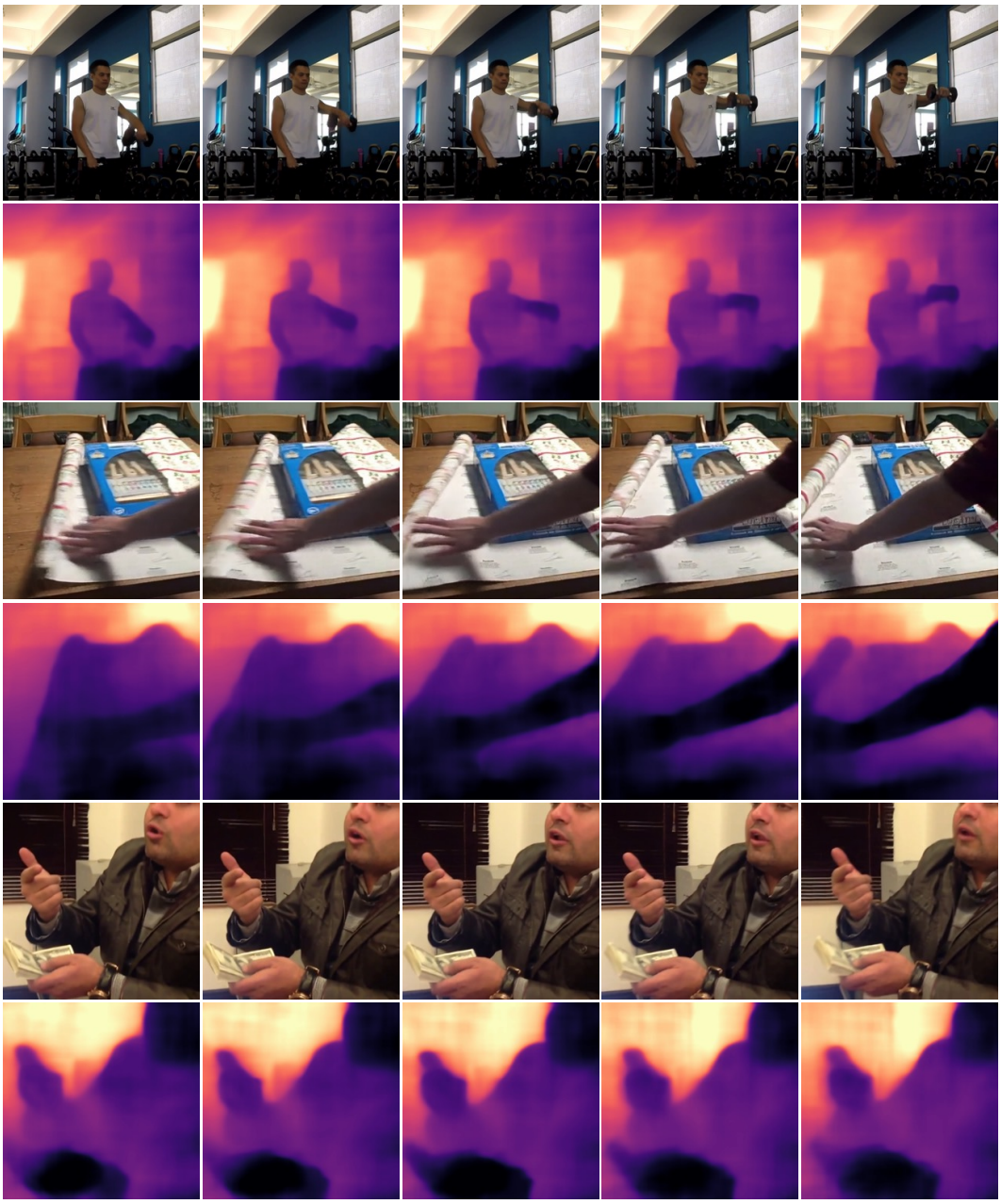


Figure 11: **Visualization of structural depth grounding on dataset K400.** VGGA predicts temporally coherent depth maps that capture scene layout and object boundaries.

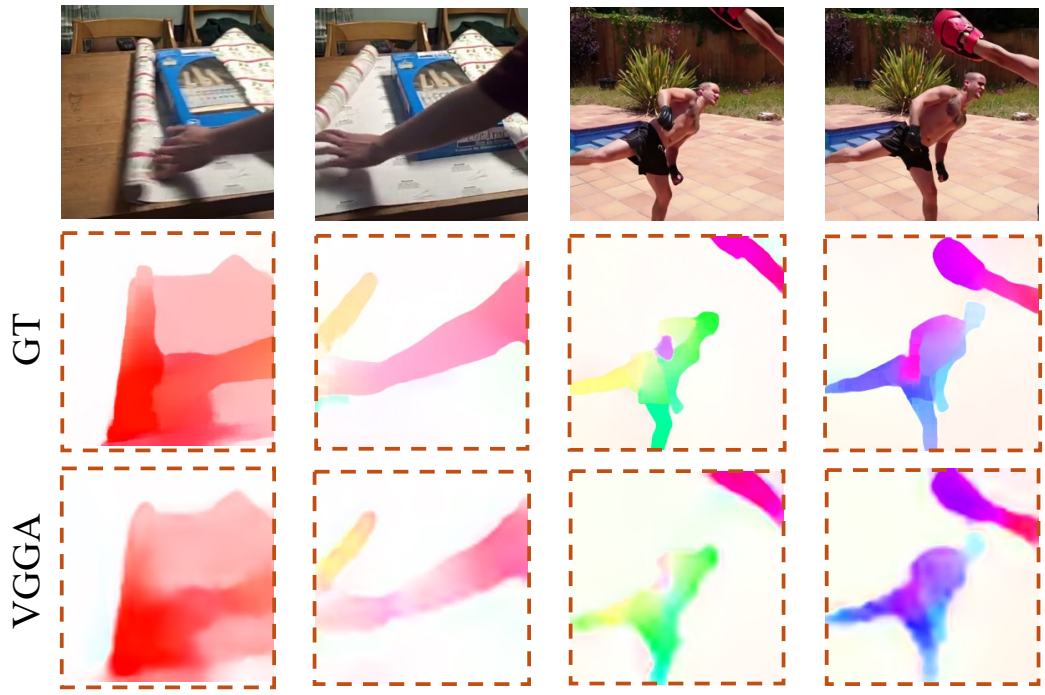


Figure 12: **Visualization of motion correspondence grounding.** VGGA predicts dense optical flow that preserves coherent object and camera motion across frames.