GeoNVS: Geometry Grounded Video Diffusion for Novel View Synthesis

Minjun Kang[1], Inkyu Shin[2], Taeyeop Lee[1], Myungchul Kim[1], In So Kweon[1], and Kuk-Jin Yoon[1]

[1]KAIST, South Korea [2]Luma AI, USA

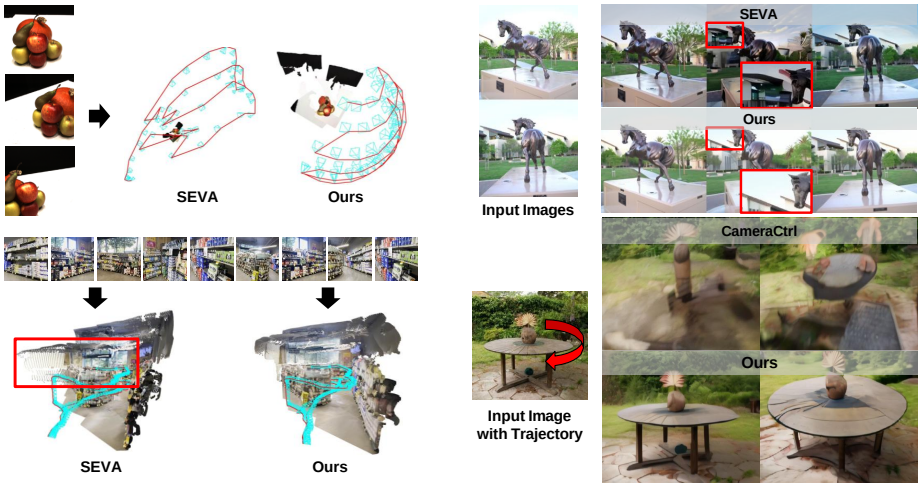


Fig. 1 We demonstrate **GeoNVS**, integrated into SEVA [87] and CameraCtrl [23], consistently enhancing geometric fidelity and camera controllability, while resolving photometric inconsistencies (**right-top**) across challenging large-scale indoor and outdoor scenes. Our approach consistently outperforms existing methods. Project page.

Abstract. Novel view synthesis requires strong 3D geometric consistency and the ability to generate visually coherent images across diverse viewpoints. While recent camera-controlled video diffusion models show promising results, they often suffer from geometric distortions and limited camera controllability. To overcome these challenges, we introduce **GeoNVS**, a geometry-grounded novel-view synthesizer that enhances both geometric fidelity and camera controllability through explicit 3D geometric guidance. Our key innovation is the Gaussian Splatting Feature Adapter (GS-Adapter), which lifts input-view diffusion features into 3D Gaussian representations, renders geometry-constrained novel-view features, and adaptively fuses them with diffusion features to correct geometrically inconsistent representations. Unlike prior methods that inject geometry at the input level, GS-Adapter operates in feature space, avoiding view-dependent color noise that degrades structural consistency. Its plug-and-play design enables zero-shot compatibility with diverse feed-forward geometry models without additional training, and can be adapted to other video diffusion backbones. Experiments across 9 scenes and 18 settings demonstrate state-of-the-art performance, achieving 11.3% and 14.9% improvements over SEVA and CameraCtrl, with up to 2× reduction in translation error and 7× in Chamfer Distance.

1 Introduction

Novel View Synthesis (NVS) aims to synthesize photorealistic images from novel viewpoints while preserving the geometric coherence of the observed scene. Neural Radiance Fields (NeRF) [51] and 3D Gaussian Splatting (3D-GS) [34] are representative per-scene optimization solutions that require a large number of observations for training. These methods struggle with sparse-view inputs and are limited to novel views near input viewpoints. Recently, feed-forward geometry methods [7,9,43,62,64,68,77] and generative approaches [18,23,24,69,86,87] have emerged as strong alternatives for such challenging scenarios.

Feed-forward geometry methods directly estimate explicit 3D representations from sparse-view input images in a single forward pass. One line of work [9,43,62,77] predicts 3D Gaussians directly from inputs and renders novel views via Gaussian splatting. Another line of work [33, 64, 68] first estimates point cloud maps and then requires optimization to fit 3D Gaussians [16]. Despite their strong structural consistency in observed regions, these methods often fail to preserve fine-grained appearance details in low-overlap scenes.

Generative novel-view synthesis represents another powerful paradigm for handling sparse-view inputs, enabling dense and semantically rich scene synthesis. These methods incorporate camera [1, 4, 18, 23, 69, 87] or point trajectories [19, 75, 80] into video diffusion models to guide novel view generation. They exhibit remarkable capability even for scenes with low-overlap inputs, overcoming limitations of feed-forward geometry methods. However, these approaches still suffer from multi-view inconsistency, hallucinating structures that deviate from the input images, and limited camera controllability, failing to follow the intended camera trajectories as shown in Fig. 1. A natural remedy is to incorporate explicit geometry priors from feed-forward geometry models to enforce structural consistency between generated and input views.

Indeed, several studies [6, 10, 47, 57, 71–73, 82] have explored this direction, combining geometry priors with generative models to achieve geometrically consistent and dense novel-view synthesis. These methods feed noisy rendered novel-view images [47,57,71,73,82] or features [6,10,72] into a diffusion model for refinement. However, we observe that these input-level fusion methods in Fig. 2-(b) introduce geometric distortions and hallucinated structures in the final output, as noisy textures in the rendered images obscure structural information.

To overcome this limitation, we propose **GeoNVS**, which couples geometry priors with the diffusion model in feature space, enabling the diffusion model to leverage explicit 3D structural information during the denoising process. Our core component, the Gaussian Splatting Feature Adapter (**GS-Adapter**), bridges 3D Gaussians and the diffusion model through a feature embedding and rendering process, modulating geometrically inconsistent diffusion features. Specifically, GS-Adapter (1) embeds 2D input-view diffusion features into 3D Gaussians, (2) rasterizes geometry-aware novel-view features via Gaussian splatting, and (3) adaptively fuses them with novel-view diffusion features to produce geometrically corrected features. By anchoring diffusion features to explicit 3D structure, GS-Adapter ensures that generated pixels are geometrically aligned with the target camera viewpoint, leading to improved camera controllability.

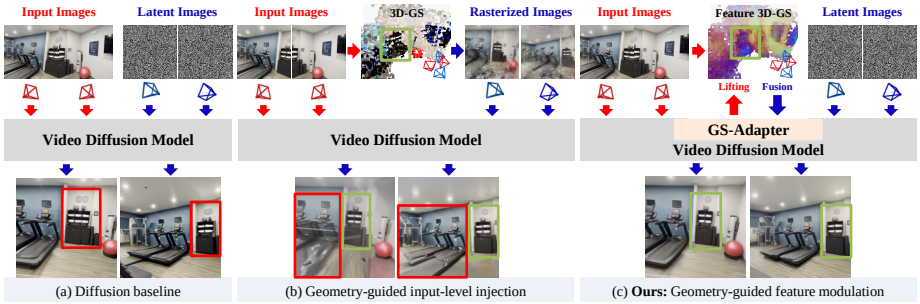


Fig. 2 Geometry-guided generative NVS. (a) Pure diffusion model produces view-inconsistent results. Given sparse input-view images, both (b) and (c) reconstruct 3D Gaussians from input views using a geometry prior. (b) Previous methods inject rasterized novel-view images from 3D-GS as input, causing artifacts from noisy rasterized colors. (c) Our method modulates internal diffusion features via a GS-Adapter conditioned on 3D-GS, achieving superior geometry consistency and visual quality.

Compared to previous methods that perform input-level fusion in Fig. 2-(b), our approach in Fig. 2-(c) leverages only the structural information encoded in 3D Gaussians, avoiding reliance on noisy view-dependent color information. Our modular design enables ***plug-and-play*** integration with diverse feed-forward geometry models without additional training, while existing methods [10,39,70,82] depend on a specific geometry model, requiring retraining upon substitution. To validate the effectiveness of **GeoNVS**, we integrate GS-Adapter into SEVA [87] and CameraCtrl [23] and evaluate across indoor and outdoor scenes in both low- and large-overlap settings as well as long-trajectory generation, demonstrating consistent improvements in geometric consistency, camera controllability, and photorealistic quality. In summary, our key contributions are as follows:

- We propose **GS-Adapter**, which couples 3D Gaussian priors with the diffusion model in feature space, improving geometric consistency and camera controllability in video diffusion-based novel-view synthesis.
- Our modular design supports diverse combinations of feed-forward geometry models and video diffusion backbones without additional training, enabling flexible plug-and-play applicability.
- **GeoNVS** achieves state-of-the-art performance over existing generative NVS methods across diverse indoor and outdoor benchmarks, evaluated on 9 scenes and 18 settings.

2 Related Work

Sparse-view Novel View Synthesis. NeRF [51] and 3D-GS [34] achieve photorealistic novel-view synthesis given dense inputs, but overfit severely under sparse views. To resolve this, prior works introduce depth supervision [11, 12, 63, 90] or normal consistency [61, 84], yet remain sensitive to constraint quality. Feed-forward geometry methods [7, 9, 62, 77] directly predict 3D Gaussians using cost volumes [13, 20, 28, 79] or monocular depth [14, 78, 81], and geometry foundation models [33, 43, 64, 68] further enhance reconstruction quality at scale. We leverage these priors to guide the diffusion model toward structural consistency.

Generative Novel View Synthesis. Diffusion models have evolved into powerful task executors by incorporating controllable conditioning signals [40, 85]. Camera-controlled video diffusion methods embed camera poses [69], Plücker coordinates [1, 4, 18, 23, 24, 87], trajectory-based attention [86], or projected 3D point maps [21] to guide video generation. Recently, SEVA [87] demonstrated strong novel-view synthesis via a multi-view diffusion model trained on large-scale datasets. Building on SEVA and CameraCtrl, we enhance these baselines to achieve geometrically consistent generation and improved controllability.

Feature field distillation for 3D-GS. Beyond radiance fields, embedding semantic features into 3D representations [35, 53] has enabled various downstream applications [27, 30, 38, 42, 66]. Most existing methods [41, 53, 88] distill semantic features [5, 36, 52, 54] into 3D Gaussians via per-scene optimization. Recently, Dr.Splat [32] introduced feed-forward feature distillation into 3D Gaussians, enabling fast 3D localization without per-scene optimization. LUDVIG [48] extends this to integrate DINOv2 [52] features into 3D scene geometry through graph diffusion for semantic segmentation. Unlike prior methods that distill static semantic features, diffusion features vary across denoising timesteps, making per-scene optimization infeasible. We therefore adopt a direct feed-forward lifting approach, extending it to preserve pixel-level feature fidelity rather than coarse instance-level distinctions sufficient for segmentation.

3 Method

In this section, we first provide preliminaries on the problem formulation, geometry prior construction, and Gaussian splatting background in Sec. 3.1. We then introduce **GS-Adapter** in Sec. 3.2, our core module that leverages 3D Gaussian priors to correct geometrically inconsistent diffusion features, thereby enforcing geometric consistency and improving camera controllability. Finally, we describe the integration of GS-Adapter into an existing video diffusion framework via multi-scale feature aggregation in Sec. 3.3.

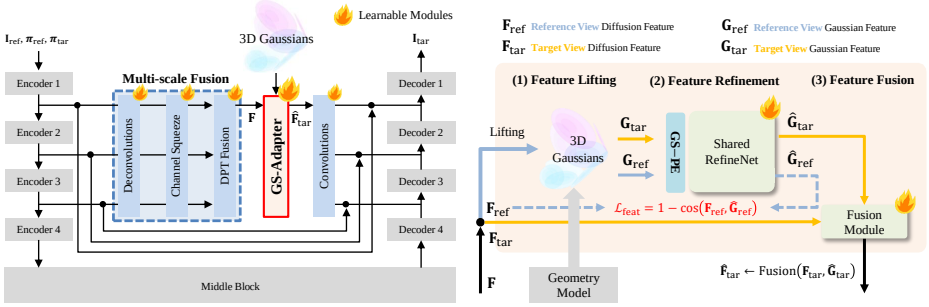
3.1 Preliminary

Problem Formulation. Given P reference images $\mathbf{I}_{\text{ref}} \in \mathbb{R}^{P\times hw\times 3}$ and their corresponding camera poses $\boldsymbol{\pi}^{\text{ref}}$, the NVS task is to synthesize Q target views $\mathbf{I}_{\text{tgt}} \in \mathbb{R}^{Q\times hw\times 3}$ for the target camera poses $\boldsymbol{\pi}^{\text{tgt}}$. **GeoNVS** supports an arbitrary number of reference images, as both SEVA [87] and the geometry models natively handle monocular and multi-view inputs.

Geometry Prior. Our method requires 3D Gaussians $\mathcal{G}$ as structural guidance for the diffusion model. We utilize feed-forward geometry models [9, 62, 64, 68, 77] to estimate this geometry prior from the reference images. For geometry models that estimate both camera poses and point clouds (VGGT [64] and Pi3 [68]), we align the predicted camera scale s^* to the target cameras by solving:

$$s^* = \arg\min_s \left\| \mathbf{C}^{\text{gt}} - s\,\mathbf{C}^{\text{pred}} \right\|_2^2 \tag{1}$$

where $\mathbf{C}^{\text{gt}}$ and $\mathbf{C}^{\text{pred}}$ denote the target and predicted camera centers, respectively. We then fit 3D Gaussians initialized from the estimated point clouds [16].



(a) Architecture Overview (b) GS-Adapter Pipeline

Fig. 3 GeoNVS architecture. (a) Overview of the integration with a video diffusion model. (b) The GS-Adapter pipeline for feature lifting, refinement, and fusion. All learnable modules (🔥) are trained with LoRA [25]. During training, a consistency loss $\mathcal{L}_{\text{feat}}$ is applied to preserve geometric detail lost during feature lifting. Please refer to the supplementary material for details of the multi-scale fusion module and RefineNet.

3D Gaussian Splatting. 3D Gaussian Splatting [34] represents a scene with N Gaussian particles composed of position $\boldsymbol{\mu} \in \mathbb{R}^3$, rotation $\boldsymbol{r} \in \mathbb{R}^4$ via quaternions, scale $\boldsymbol{s} \in \mathbb{R}^3$, opacity $\boldsymbol{\sigma} \in \mathbb{R}^1$, and color represented by spherical harmonics. The color at pixel p for view d is obtained by summing contributions of Gaussians:

$$C(d,p) = \sum_{i=1}^{N} c_i(d)\, w_i(d,p), \quad w_i(d,p) = \alpha_i(d,p) \prod_{j=1}^{i-1} (1 - \alpha_j(d,p)) \tag{2}$$

where $\alpha_i(d,p)$ is the product of the opacity σ_i and the 2D projected Gaussian density $\mathcal{N}_i$ at pixel position p. Similarly, feature rasterization is represented as:

$$\mathbf{F}(d,p) = \sum_{i=1}^{N} \mathrm{f}_i\, w_i(d,p) \tag{3}$$

3.2 GS-Adapter: Geometry-Guided Diffusion Feature Adaptation

Given the diffusion features $\mathbf{F}^t$ at denoising timestep t and 3D Gaussians prior $\mathcal{G}$, GS-Adapter is designed to obtain geometry-augmented features $\hat{\mathbf{F}}^t$, formulated as $\hat{\mathbf{F}}^t = \Phi(\mathbf{F}^t, \mathcal{G})$. Diffusion features $\mathbf{F}^t = [\mathbf{F}^t_{\text{ref}}; \mathbf{F}^t_{\text{tar}}]$ consist of reference-view (input-view) features $\mathbf{F}^t_{\text{ref}} \in \mathbb{R}^{P \times \frac{h}{n} \times \frac{w}{n} \times \mathrm{C}}$ and novel-view features $\mathbf{F}^t_{\text{tar}} \in \mathbb{R}^{Q \times \frac{h}{n} \times \frac{w}{n} \times \mathrm{C}}$, where n denotes the downsampling factor of the Variational Autoencoder (VAE), and C denotes the channel dimension of the diffusion features. For convenience, we omit the superscript t hereafter. As illustrated in Fig. 3-(b), GS-Adapter consists of three major steps: (1) **feature lifting:** projecting reference-view diffusion features onto 3D Gaussians, (2) **feature refinement:** refining the rendered novel-view features to recover fine-grained details, and (3) **feature fusion:** combining the geometry-guided features with the original diffusion features to obtain the revised diffusion features.

Uplifting diffusion feature into 3D-GS. Since the diffusion features vary across timesteps and our method requires obtaining geometry-augmented features at every denoising timestep, per-scene optimization-based feature embedding approaches [41, 53, 88] are not feasible, as they cannot adapt to the continuously changing feature distribution during denoising. Inspired by [32, 48], we

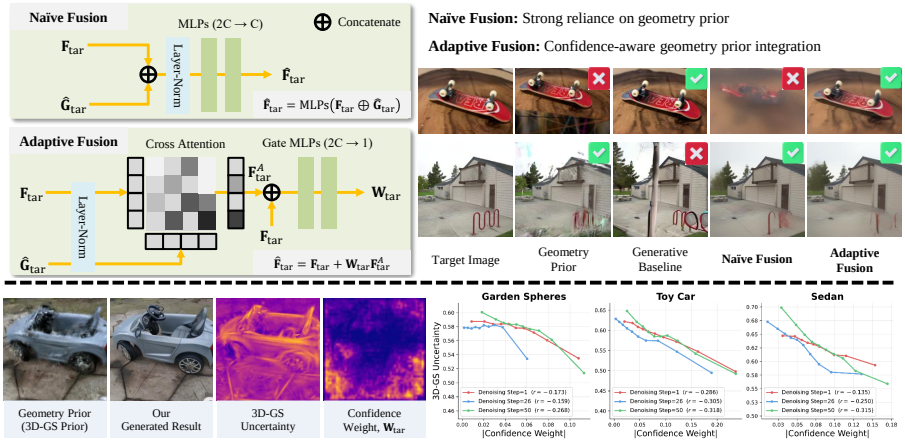


Fig. 4 Feature fusion module of GS-Adapter. Two fusion approaches are proposed to integrate the diffusion feature $\mathbf{F}_{\text{tar}}$ and the geometry-aware feature $\hat{\mathbf{G}}_{\text{tar}}$, producing the updated novel-view feature $\hat{\mathbf{F}}_{\text{tar}}$. We adopt adaptive fusion as it remains effective even when either the geometry prior or the generative model fails.

obtain uplifted diffusion features for each Gaussian as a weighted average of 2D reference diffusion features $\mathbf{F}_{\text{ref}}$. Each 2D feature at the viewing direction d and pixel p contributes to the Gaussian feature $\mathbf{f}_i$ proportionally to the rendering weight $w_i(d,p)$ with normalization, where $\mathcal{I}_i$ denotes the set of view/pixel pairs contributing to each Gaussian. The uplifted feature $\hat{\mathbf{f}}_i \in \mathbb{R}^{\text{C}}$ is defined as:

$$\hat{\mathbf{f}}_i = \frac{\mathbf{f}_i}{\|\mathbf{f}_i\|_2}, \qquad \mathbf{f}_i = \sum_{(d,p)\in\mathcal{I}_i} \frac{w_i(d,p)}{\sum_{(d',p')\in\mathcal{I}_i} w_i(d',p')} \mathbf{F}_{\text{ref}}(d,p) \tag{4}$$

This feed-forward uplifting enables explicit participation of 3D Gaussians at every denoising step, allowing end-to-end training with the diffusion model.

Feature refinement. After uplifting $\mathbf{F}_{\text{ref}}$ into the 3D Gaussians, we rasterize the geometry-constrained novel-view features $\mathbf{G}_{\text{tar}} \in \mathbb{R}^{Q\times\frac{h}{n}\times\frac{w}{n}\times\text{C}}$ using Eq. (3). Our next goal is to fuse these geometry-aware features with the original novel-view diffusion features $\mathbf{F}_{\text{tar}}$. However, we observe that direct fusion alone is insufficient for recovering fine-grained details, as shown in Fig. 5, consistent with the findings of [48]. To compensate for the information loss during lifting and rendering, we augment the rendered features $\mathbf{G}$ with Gaussian positional encoding (GS-PE) and refine them using a lightweight ResNet-based refinement network, $\mathcal{R}$. Specifically, for each pixel (d,p), we identify the most contributing Gaussian $i^*(d,p) = \arg\max_i w_i(d,p)$, normalize its 3D center within the scene bounding box, and encode it alongside the rendering weight $w^*(d,p)$ via sinusoidal encoding $\gamma(\cdot)$ to obtain the positionally-encoded features $\mathbf{G}'$:

$$\mathbf{G}'(d,p) = \mathbf{G}(d,p) + [\,\gamma(\tilde{x}) \,\|\, \gamma(\tilde{y}) \,\|\, \gamma(\tilde{z}) \,\|\, \gamma(w^*)\,] \tag{5}$$

where $\tilde{\mathbf{x}}(d,p) = (\tilde{x},\tilde{y},\tilde{z})$ denotes the normalized 3D center of $i^*(d,p)$, and $\gamma(v) = [\sin(\omega_k v), \cos(\omega_k v)]_{k=0}^{D'/2-1}$ with $\omega_k = \omega_0^{-2k/D'}$ and $D' = \text{C}/4$. This equips the refinement network with explicit 3D structural cues. Afterwards, we get refined features $\hat{\mathbf{G}} = \mathcal{R}(\mathbf{G}')$. Because the original input-view features $\mathbf{F}_{\text{ref}}$ serve as a reliable reference, we supervise $\mathcal{R}$ by aligning the refined input-view features

$\hat{\mathbf{G}}_{\text{ref}} \in \mathbb{R}^{P \times \frac{h}{n} \times \frac{w}{n} \times \text{C}}$ with $\mathbf{F}_{\text{ref}}$ during training via a cosine similarity loss:

$$\mathcal{L}_{\text{feat}} = \left\| 1 - \frac{\hat{\mathbf{G}}_{\text{ref}} \cdot \mathbf{F}_{\text{ref}}}{\|\hat{\mathbf{G}}_{\text{ref}}\|_2 \, \|\mathbf{F}_{\text{ref}}\|_2} \right\|_2 \tag{6}$$

Adaptive feature fusion. Given the refined novel-view features $\hat{\mathbf{G}}_{\text{tar}}$, we fuse them with the diffusion features $\mathbf{F}_{\text{tar}}$ to obtain the geometry-augmented features $\hat{\mathbf{F}}_{\text{tar}}$. A naïve approach is to concatenate $\mathbf{F}_{\text{tar}}$ and $\hat{\mathbf{G}}_{\text{tar}}$, then project them back to the original channel dimension via MLPs:

$$\hat{\mathbf{F}}_{\text{tar}} = \text{MLPs}\Big(\mathbf{F}_{\text{tar}} \oplus \hat{\mathbf{G}}_{\text{tar}}\Big) \tag{7}$$

However, this fusion relies entirely on the geometry prior and fails once it is corrupted (Fig. 4). We instead propose an adaptive fusion that incorporates geometry based on its local reliability. Specifically, we first obtain the geometry-attended feature $\mathbf{F}^{A}_{\text{tar}}$ by cross-attending from $\mathbf{F}_{\text{tar}}$ (query) to $\hat{\mathbf{G}}_{\text{tar}}$ (key/value). We then predict a pixel-wise confidence weight $\mathbf{W}_{\text{tar}} \in [-1, 1]$ via a gating MLP, and combine $\mathbf{F}_{\text{tar}}$ and $\mathbf{F}^{A}_{\text{tar}}$ weighted by $\mathbf{W}_{\text{tar}}$:

$$\mathbf{F}^{A}_{\text{tar}} = \text{CrossAttn}\Big(\mathbf{F}_{\text{tar}}, \hat{\mathbf{G}}_{\text{tar}}\Big), \qquad \mathbf{W}_{\text{tar}} = \text{Tanh}\big(\text{MLP}\big(\mathbf{F}_{\text{tar}} \oplus \mathbf{F}^{A}_{\text{tar}}\big)\big) \tag{8}$$

$$\hat{\mathbf{F}}_{\text{tar}} = \mathbf{F}_{\text{tar}} + \mathbf{W}_{\text{tar}} \cdot \mathbf{F}^{A}_{\text{tar}} \tag{9}$$

To verify that $\mathbf{W}_{\text{tar}}$ behaves as intended, we measure the Pearson correlation between the absolute confidence weights, $|\mathbf{W}_{\text{tar}}|$ and 3D-GS uncertainty [22] across 3 reflective scenes (Fig. 4). The consistent negative correlation ($r \approx -0.30$), which strengthens as denoising progresses and is most pronounced in reflective regions, supports our design claim: Adaptive Fusion downweights geometric guidance in regions where the geometry prior is unreliable.

3.3 Integration into Video Diffusion

Integrating 3D-GS into video diffusion. GeoNVS incorporates explicit 3D structural information into the denoising process by coupling 3D Gaussians with the diffusion model in feature space. At each denoising timestep t, GS-Adapter embeds $\mathbf{F}^{t}_{\text{ref}}$ into 3D Gaussians, rasterizes geometry-aware novel-view features, and adaptively injects them back into the diffusion model to correct geometrically inconsistent features throughout the entire denoising process.

Multi-scale feature aggregation. Multi-scale features play a crucial role in visual understanding [8,44,55,65,67] and generation [46,58]. To effectively combine geometry-guided features with diffusion features across different spatial scales, we leverage multi-scale encoder features via a DPT-style [55] aggregation strategy, upsampling and hierarchically merging them to a unified feature of $\mathbf{F} \in \mathbb{R}^{(P+Q) \times \frac{h}{n} \times \frac{w}{n} \times \text{C}}$ before feeding into GS-Adapter (see Fig. 3). The geometry-corrected features are then downsampled to the original resolutions and added through skip connections to the decoder. We validate this design in Sec. 4.6.

Loss function. We train the multi-scale fusion module and GS-Adapter alongside trainable LoRA [25] layers injected into the attention modules of the frozen video diffusion model. The total training objective is formulated by combining the latent space alignment loss [3] and the feature alignment loss from Eq. (6):

$$\mathcal{L} = \mathcal{L}_{\text{latent}} + 0.05\mathcal{L}_{\text{feat}} \tag{10}$$

4 Experiment

4.1 Overview

We organize our experiments to validate three key claims of GeoNVS. First, GS-Adapter improves geometric consistency and camera controllability, quantified by camera pose error and Chamfer Distance, and further supported by a feature consistency metric (Secs. 4.4 and 4.5). Second, GeoNVS achieves state-of-the-art photorealistic quality across diverse scenes, evaluated in Sec. 4.3 on 9 scenes under small/large-viewpoint and long-trajectory settings. Third, GS-Adapter supports plug-and-play integration with diverse geometry models without additional training, and is compatible with other diffusion backbones; we further show that its computational overhead can be substantially reduced via voxel-based Gaussian pruning with negligible performance degradation (Sec. 4.6).

4.2 Implementation Details

We implement GeoNVS on two baselines, SEVA [87] and CameraCtrl [23], to demonstrate its compatibility. While the SEVA-based GeoNVS supports an arbitrary number of input and output views, the CameraCtrl-based GeoNVS is limited to a single reference view due to constraints of the baseline model.

Training setup. We use the DL3DV-10K [45] dataset to train GeoNVS. Using VGGT [64] and InstantSplat [16], we construct 3D Gaussians paired with 150K clips with varying reference-to-novel view splits $(P, Q) \in \{(1, 20),\ (3, 18),\ (6, 15),\ (9, 12),\ (12, 9)\}$ where the total number of views is fixed at 21. Please refer to Sec. E of the supplementary material for details on dataset preprocessing. Both SEVA- and CameraCtrl-based GeoNVS are trained at resolutions (h, w) of 384×384 and 320×576 with initial learning rates of 3×10^{-5} and 1×10^{-4}, respectively. Our model is fine-tuned using LoRA [25] on 8 NVIDIA A6000 GPUs (48GB). We set the rank and α to 16 for SEVA, and 4 for CameraCtrl. The training process takes approximately 80 hours with a total batch size of 8.

Inference setup. We evaluate our method on the SEVA benchmark [87], consisting of 9 scenes [2, 29, 37, 45, 50, 56, 74, 76, 89] with input views P varying from 1 to 9. We include Tanks and Temples [37] under the CF-3DGS sampling protocol [17] for the view interpolation task (T&T O split). For fair comparison, all combined methods (generative + geometry prior) use the same best-performing geometry prior per dataset. Note that MVSplat360 [10] is limited to MVSplat [9] by design. We evaluate all methods on a single A6000 GPU. Please refer to Sec. B of the supplementary material for details, including evaluation dataset splits.

Input-level injection of geometry prior. As a baseline for input-level geometry conditioning into the diffusion model in Fig. 2-(b), we implement an input-level injection approach following prior work [49, 71, 73]. Given the VAE-encoded latents $\mathbf{z}_{\text{tar}}$ of 3D-GS rendered novel-view images, we initialize the noisy latents as $\mathbf{x}_{\text{tar}} \leftarrow \mathbf{x} \cdot \sigma_{t_0} + \mathbf{z}_{\text{tar}}$, where σ_{t_0} is the noise scale at the start timestep $t_0 = \lfloor T \cdot s \rfloor$ determined by a strength $s \in [0, 1]$. A larger s allows the diffusion model more freedom to deviate from the geometry prior, while a smaller s enforces stronger adherence to the rendered geometry; we set $s = 0.2$. We compare this approach against our GS-Adapter, which modulates internal diffusion features with a geometry prior rather than relying on rasterized images.

Table 1 Perceptual metrics on small-viewpoint set NVS. The small-viewpoint set contains scenes with high view overlap between reference and target views. P denotes the number of reference images. For P=1, we sweep the unit length for camera normalization [87] to resolve scale ambiguity. All combined methods (generative + geometry) use the best feed-forward geometry model per dataset as the geometry prior.

(a) PSNR ↑ on small-viewpoint set NVS.

Method	dataset	OO3D [74]	RE10K [89]				LLFF [50]		DTU [29]		CO3D [56]		WRGBD [76]		Mip360 [2]	DL3DV [45]		T&T [37]		Average
	split	S	D	P	R		R		R		V	R	S_e	S_h	R	S	L	O		
	P	3	1	2	1	3	1	3	1	3	1	3	3	6	6	6	9	2	3	
Feed-forward geometry models																				
MVSplat [9]		21.82	10.68	25.56	11.26	23.35	5.69	13.68	7.47	11.35	10.12	12.04	12.89	12.38	13.45	13.89	14.07	12.52	13.45	13.65
HiSplat [62]		21.98	12.38	25.50	13.89	25.81	5.75	15.16	7.11	12.37	10.59	12.55	13.49	12.48	13.90	14.04	14.20	12.38	15.57	14.40
DepthSplat [77]		22.66	13.02	**25.57**	14.49	26.99	5.91	15.02	8.79	14.19	11.29	15.08	13.92	13.50	15.04	15.87	16.88	15.76	17.99	15.67
VGGT [64]		22.12	13.73	23.98	13.95	27.48	10.89	**20.08**	7.05	17.15	10.48	15.39	11.09	12.56	15.49	17.09	18.31	**20.92**	**23.96**	16.76
Pi3 [68]		18.48	10.56	24.35	11.00	**27.66**	8.80	15.54	8.82	7.81	12.08	12.60	10.54	10.46	13.16	16.03	17.23	14.46	16.98	14.25
Generative models																				
MotionCtrl [69]		-	13.10	-	13.86	-	11.40	-	8.41	-	13.64	-	-	-	-	-	-	-	-	12.08
CameraCtrl [23]		-	13.08	-	14.09	-	11.88	-	8.72	-	13.70	-	-	-	-	-	-	-	-	12.29
ViewCrafter [83]		18.56	14.37	-	17.44	15.83	11.79	11.39	9.22	8.49	15.87	9.89	9.23	9.37	10.81	10.51	9.75	10.74	11.13	12.02
SEVA [87]		26.28	14.51	22.77	15.25	24.44	12.22	17.12	10.54	17.03	16.29	17.22	15.96	15.71	15.38	15.84	16.87	16.35	18.85	17.15
Generative model + Geometry prior																				
MVSplat360 [10]		21.45	13.68	20.20	13.44	20.97	12.79	15.31	9.10	11.38	12.69	14.29	14.34	13.22	14.15	15.09	15.26	14.22	15.84	14.86
GenFusion [73]		20.90	14.87	22.48	15.88	22.69	11.39	17.87	8.36	13.75	12.15	12.99	13.38	12.47	11.99	13.11	14.78	18.25	20.14	15.30
Difix3D [71]		23.05	12.92	25.23	14.37	27.14	10.85	19.59	8.79	17.22	11.93	15.23	12.60	12.67	14.95	16.33	17.33	20.68	23.36	16.89
GeoNVS (**Ours**)		**26.58**	**16.66**	24.78	**17.62**	27.11	**14.10**	18.29	**12.12**	**19.18**	**17.14**	**19.26**	**18.23**	**17.12**	**16.57**	**18.11**	**19.66**	18.76	22.24	**19.09**

(b) SSIM ↑ on small-viewpoint set NVS.

Method	dataset	OO3D [74]	RE10K [89]				LLFF [50]		DTU [29]		CO3D [56]		WRGBD [76]		Mip360 [2]	DL3DV [45]		T&T [37]		Average
	split	S	D	P	R		R		R		V	R	S_e	S_h	R	S	L	O		
	P	3	1	2	1	3	1	3	1	3	1	3	3	6	6	6	9	2	3	
Feed-forward geometry models																				
HiSplat [62]		0.877	0.444	**0.872**	0.540	0.902	0.01	0.384	0.149	0.467	0.286	0.434	0.434	0.419	0.290	0.419	0.457	0.360	0.468	0.456
DepthSplat [77]		0.897	0.468	0.868	0.529	**0.918**	0.02	0.332	0.262	0.485	0.337	0.492	0.441	0.434	0.303	0.465	0.528	0.472	0.535	0.488
VGGT [64] + InstantSplat [16]		0.892	0.499	0.798	0.561	0.914	0.234	**0.672**	0.215	0.640	0.297	0.524	0.409	0.432	0.362	0.518	0.571	**0.716**	**0.807**	0.559
Generative models																				
CameraCtrl [23]		-	0.491	-	0.578	-	0.294	-	0.400	-	0.451	-	-	-	-	-	-	-	-	0.443
ViewCrafter [83]		0.861	0.551	-	**0.693**	0.647	0.283	0.253	0.370	0.274	0.520	0.300	0.233	0.215	0.194	0.241	0.230	0.286	0.300	0.379
SEVA [87]		**0.926**	0.534	0.777	0.616	0.842	0.294	0.467	0.430	0.659	0.507	0.551	0.527	0.522	0.339	0.450	0.513	0.521	0.608	0.560
Generative model + Geometry prior																				
GenFusion [73]		0.876	0.559	0.818	0.618	0.864	0.271	0.554	0.355	0.556	0.438	0.449	0.453	0.425	0.282	0.394	0.492	0.630	0.689	0.540
Difix3D [71]		0.898	0.463	0.857	0.528	0.903	0.230	0.638	0.342	0.637	0.375	0.479	0.350	0.333	0.301	0.481	0.519	0.696	0.779	0.545
GeoNVS (**Ours**)		0.904	**0.612**	0.826	0.680	0.895	**0.346**	0.514	**0.480**	**0.691**	**0.550**	**0.627**	**0.607**	**0.577**	**0.383**	**0.562**	**0.635**	0.603	0.695	**0.622**

(c) LPIPS ↓ on small-viewpoint set NVS.

Method	dataset	OO3D [74]	RE10K [89]				LLFF [50]		DTU [29]		CO3D [56]		WRGBD [76]		Mip360 [2]	DL3DV [45]		T&T [37]		Average
	split	S	D	P	R		R		R		V	R	S_e	S_h	R	S	L	O		
	P	3	1	2	1	3	1	3	1	3	1	3	3	6	6	6	9	2	3	
Feed-forward geometry models																				
HiSplat [62]		0.155	0.522	0.110	0.484	0.094	1.02	0.479	0.783	0.483	0.645	0.581	0.496	0.610	0.691	0.627	0.650	0.612	0.474	0.529
DepthSplat [77]		0.134	0.444	0.100	0.414	0.074	0.979	0.351	0.550	0.362	0.553	0.498	0.431	0.493	0.552	0.487	0.477	0.332	0.235	0.415
VGGT [64] + InstantSplat [16]		0.172	0.441	0.138	0.454	0.078	0.517	**0.153**	0.779	0.340	0.608	0.530	0.696	0.623	0.618	0.466	0.497	0.194	0.127	0.413
Generative models																				
CameraCtrl [23]		-	0.514	-	0.492	-	0.610	-	0.611	-	0.540	-	-	-	-	-	-	-	-	0.553
ViewCrafter [83]		0.211	0.405	-	0.338	0.385	0.527	0.562	0.634	0.690	0.425	0.727	0.715	0.727	0.675	0.649	0.680	0.542	0.516	0.553
SEVA [87]		**0.062**	0.400	0.131	0.402	0.095	0.469	0.212	0.524	0.210	0.399	0.333	0.339	0.331	0.367	0.305	0.288	0.238	0.164	0.293
Generative model + Geometry prior																				
GenFusion [73]		0.158	0.408	0.183	0.384	0.159	0.554	0.301	0.702	0.430	0.621	0.616	0.507	0.577	0.655	0.537	0.520	0.278	0.223	0.434
Difix3D [71]		0.106	0.447	**0.097**	0.422	0.083	0.520	**0.153**	0.648	0.240	0.544	0.456	0.508	0.549	0.425	0.313	0.326	**0.158**	**0.101**	0.339
GeoNVS (**Ours**)		0.074	**0.325**	0.113	**0.319**	**0.071**	**0.412**	0.208	**0.449**	**0.174**	**0.393**	**0.279**	**0.275**	**0.290**	**0.329**	**0.245**	**0.235**	0.225	0.138	**0.253**

Table 2 Large-viewpoint set NVS. The large-viewpoint set contains scenes with low view overlap between reference and target views. P denotes the number of reference images. We use DepthSplat [77] as the geometry prior for both combined methods.

(a) PSNR ↑ results.

Method	dataset	CO3D [56]	WRGBD [76]		Mip360 [2]		DL3DV [45]		Average
	split	R	S_h		R		S		
	P	1	1	3	1	3	1	3	
Geometry prior									
DepthSplat [77]		8.54	7.31	11.94	9.20	14.22	7.37	14.67	10.46
Generative models									
MotionCtrl [69]		11.77	10.33	-	11.04	-	10.99	-	11.03
CameraCtrl [23]		11.71	10.38	-	10.94	-	11.05	-	11.02
ViewCrafter [83]		11.11	-	9.32	10.79	10.38	11.18	10.20	10.50
SEVA [87]		12.27	11.27	13.72	11.19	14.30	11.48	14.27	12.64
Generative model + Geometry prior									
GenFusion [73]		10.21	9.60	11.37	9.84	11.88	10.11	13.33	10.91
Difix3D [71]		8.56	7.33	11.81	9.30	13.83	7.48	14.24	10.36
GeoNVS (**Ours**)		**14.66**	**12.72**	**15.08**	**12.34**	**15.47**	**13.07**	**16.09**	**14.20**

Table 3 Long trajectory NVS. The long trajectory NVS measures the long-term generation ability of generative NVS. P denotes the number of reference images. We use VGGT [64] as the geometry prior for our method.

(a) PSNR ↑ results.

Method	dataset	Mip360 [2]			DL3DV [45]			T&T [37]		
	P	3	6	9	3	6	9	2	3	6
DepthSplat [77]		13.22	13.91	14.57	14.79	15.90	16.68	15.69	17.51	18.11
VGGT [64]		12.16	14.45	15.80	15.43	16.99	18.11	**20.99**	**24.06**	**26.84**
SEVA [87]		13.01	14.32	15.16	14.11	15.65	16.73	16.59	19.27	23.09
Ours		**14.16**	**15.52**	**16.45**	**16.03**	**17.99**	**19.32**	19.26	22.31	24.52

(b) Pose error and chamfer distance ↓.

Dataset (P)	RE10K [89] (3)			DL3DV [45] (3)			T&T [37] (3)		
Metric (↓)	T_{err}	R_{err}	CD	T_{err}	R_{err}	CD	T_{err}	R_{err}	CD
SEVA [87]	1.62	5.84	1.35	35.0	6.44	1.03	6.08	2.87	1.42
Ours	**0.76**	**4.83**	**0.19**	**28.3**	**5.13**	**0.62**	**3.56**	**1.52**	**0.92**

Table 4 Geometry Fidelity of GeoNVS. T_{err}, R_{err}, and CD measure geometric fidelity. PSNR_{V} and PSNR_{U} denote synthesis quality on co-visible and non-co-visible regions w.r.t. the reference views. All the results are under the Trajectory NVS setting. "Input-level injection" denotes input-level integration of 3D Gaussian priors.

Method	Mip360 Dataset [2] (3 reference views)					DL3DV Dataset [45] (6 reference views)					WRGBD-S_h Dataset [76] (3 reference views)				
	$T_{\text{err}}\downarrow$	$R_{\text{err}}\downarrow$	$CD\downarrow$	$\text{PSNR}_{\text{V}}\uparrow$	$\text{PSNR}_{\text{U}}\uparrow$	$T_{\text{err}}\downarrow$	$R_{\text{err}}\downarrow$	$CD\downarrow$	$\text{PSNR}_{\text{V}}\uparrow$	$\text{PSNR}_{\text{U}}\uparrow$	$T_{\text{err}}\downarrow$	$R_{\text{err}}\downarrow$	$CD\downarrow$	$\text{PSNR}_{\text{V}}\uparrow$	$\text{PSNR}_{\text{U}}\uparrow$
Baselines															
SEVA [87]	18.76	3.29	**0.07**	13.85	13.01	19.68	2.80	0.57	15.68	16.90	8.10	8.86	1.58	15.53	15.79
Generative model + VGGT [64] prior															
Difix3D [71]	219.39	51.64	0.68	12.92	11.82	64.4	9.66	3.76	16.11	17.29	19.59	32.48	10.11	11.07	11.36
SEVA + Input-level injection	233.03	99.79	1.02	13.14	12.03	81.14	17.19	3.82	16.76	17.76	29.27	52.51	2.01	11.36	11.42
Ours	**16.24**	**2.98**	**0.07**	**15.11**	**14.07** (+1.06)	**13.27**	**2.01**	**0.41**	**18.11**	**18.98** (+2.08)	**4.00**	**4.56**	**1.29**	**17.33**	**17.51** (+1.72)

4.3 Perceptual Quality

Metrics. We evaluate perceptual quality using PSNR, SSIM, and LPIPS — standard metrics in NVS — and report PSNR and SSIM as the primary metrics in the main manuscript; full results are in Sec. F of the supplementary material.

Robustness to viewpoint overlap. Following SEVA [87], we evaluate on Set NVS, which generates a set of target views in arbitrary order, decomposed into small- and large-viewpoint settings based on the CLIP [54] distance between reference and target views (threshold: 0.11). As shown in Tabs. 1 and 2a and Figs. 6 and 7, our method consistently outperforms comparison methods across most datasets, improving over our baseline SEVA [87] on all benchmarks and demonstrating robustness to both small and large viewpoint changes.

Long trajectory generation. Trajectory NVS organizes target views as a smooth camera trajectory forming a video sequence. Leveraging the geometry prior from reference views, we adopt the two-pass sampling method from SEVA [87] to generate consistent, non-flickering long-term videos. As shown in Tabs. 3a and 3b, our method consistently improves over baseline SEVA by a large margin, with performance scaling favorably with more reference views.

4.4 Geometric Consistency and Camera Controllability

Metrics. To validate geometric fidelity, we reconstruct point clouds and estimate camera trajectories from generated videos using ViPE [26]. Camera controllability is evaluated by the translation error T_{err}[cm] and rotation error R_{err}[deg] between estimated and ground-truth camera poses following [23]. Geometric consistency is quantified by mean Chamfer Distance (CD) between point clouds reconstructed from generated and ground-truth videos [6,15]. We exclude scenes where ViPE [26] fails to reconstruct the ground-truth video (T_{err} or $R_{err} > 50$).

Geometric consistency. As shown in Tab. 3b, GeoNVS substantially improves geometric consistency over the SEVA baseline across all datasets. This is further corroborated by Tab. 4, where our method consistently outperforms input-level geometry injection approaches — Difix3D [71] and SEVA with input-level injection — by a large margin, demonstrating that feature-space modulation via GS-Adapter is more effective than injecting geometry at the input level.

Camera controllability. GeoNVS substantially improves camera controllability over the SEVA baseline. As shown in Tab. 3b and Fig. 9, the gains in synthesis quality stem primarily from improved camera controllability and geometry enhancement, with our method being especially effective in reducing translation deviation. Tab. 4 further supports this: GeoNVS achieves consistently lower pose errors than the SEVA baseline across all three datasets, while competing methods that inject geometry priors at the input level suffer catastrophic degradation in controllability, with translation and rotation errors far exceeding those of the baseline. These results indicate that anchoring diffusion features to explicit 3D structure is key to reliable camera controllability.

Geometry prior extrapolates to unseen regions. To evaluate whether geometry priors improve synthesis beyond observed regions, we construct co-visibility masks using VGGT-derived [64] point clouds following the co-visible mask construction in [16]. Specifically, we project only the input-view 3D points back onto each novel view, yielding per-pixel co-visibility masks; we measure PSNR_V on co-visible and PSNR_U on non-co-visible (unseen) regions. As shown in Table 4, our method consistently outperforms baselines on PSNR_U across all datasets, with notable gains of $+\mathbf{1.06}$ on Mip360 and $+\mathbf{2.08}$ on DL3DV over SEVA. This demonstrates that our geometry guidance, explicitly conditioned on visible regions, facilitates geometry-consistent extrapolation into unseen areas.

4.5 Feature analysis of GS-Adapter

To qualitatively validate GS-Adapter, we visualize intermediate features throughout the denoising process in Fig. 5. The initial diffusion features $\mathbf{F}^0_{\text{tar}}$ are spatially incoherent due to high noise levels, whereas the rendered Gaussian features $\mathbf{G}_{\text{tar}}$ consistently exhibit sharp structures derived from the 3D geometry prior. After refinement, $\hat{\mathbf{G}}_{\text{tar}}$ recovers fine-grained details lost during lifting and rendering. Upon fusion, the geometry-corrected diffusion features progressively align with the underlying 3D structure as denoising advances, ultimately yielding photorealistic novel-view images with improved geometric consistency. These observations confirm that GS-Adapter effectively anchors diffusion features to explicit 3D geometry throughout the entire denoising process.

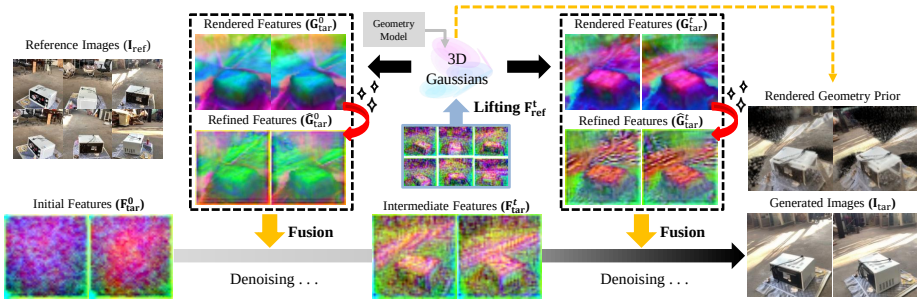


Fig. 5 Feature modulation by GS-Adapter. We visualize intermediate diffusion features during the denoising process. GS-Adapter consists of three stages: (1) **lifting** reference-view features $\mathbf{F}^t_{\text{ref}}$ into 3D Gaussians, (2) **refining** the novel-view features $\mathbf{G}_{\text{tar}}$ into $\hat{\mathbf{G}}_{\text{tar}}$, and (3) **fusing** $\hat{\mathbf{G}}_{\text{tar}}$ with $\mathbf{F}^t_{\text{tar}}$ to generate geometry-corrected outputs.

Table 5 GeoNVS with SEVA [87]. We extend GeoNVS with various geometry priors in a zero-shot manner. All methods are evaluated with 3 reference images.

Method	DTU		CO3D		WRGBD-S_e		T&T	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Baseline								
SEVA [87]	17.03	0.66	17.22	0.55	15.96	0.53	18.85	0.61
Geometry priors								
MVSplat [9]	11.35	0.41	12.04	0.39	12.89	0.41	13.45	0.38
DepthSplat [77]	14.19	0.49	15.08	0.49	**13.92**	**0.44**	17.99	0.54
VGGT [64]	**17.15**	**0.64**	**15.39**	**0.52**	11.09	0.41	**23.96**	**0.81**
Pi3 [68]	7.81	0.31	12.60	0.47	10.54	0.37	16.98	0.48
Ours w/ Naïve Fusion								
+ MVSplat	17.04	0.65	18.79	0.62	17.62	0.59	21.18	0.68
+ DepthSplat	17.94	0.68	**19.00**	**0.63**	**17.97**	**0.61**	22.05	**0.71**
+ VGGT	**18.89**	**0.69**	18.60	0.60	17.60	0.59	**22.22**	0.70
+ Pi3	14.83	0.59	18.59	0.61	17.38	0.58	21.92	0.69
Ours w/ Adaptive Fusion								
+ MVSplat	19.15	0.69	19.23	0.63	**18.22**	0.61	**22.24**	0.69
+ DepthSplat	19.14	0.69	19.24	0.63	**18.22**	0.61	**22.24**	0.69
+ VGGT	**19.18**	0.69	**19.26**	0.63	18.21	0.61	**22.24**	0.69
+ Pi3	19.02	0.69	19.22	0.63	18.19	0.61	22.22	0.69

Table 6 PSNR ↑ on GeoNVS with CameraCtrl [23]. GeoNVS built on CameraCtrl generalizes to various geometry priors in a zero-shot manner, without additional training. Following the original CameraCtrl setting, all methods are evaluated using a single reference image.

Method	dataset	CO3D		DTU	RE10K	Mip360	WRGBD	Average
	split	V	R	R	R	R	S_h	
Baseline								
CameraCtrl [23]		13.23	11.71	8.72	14.09	10.94	10.38	11.14
Geometry priors								
MVSplat [9]		10.12	9.14	7.37	11.26	8.55	6.48	8.82
DepthSplat [77]		11.29	8.54	8.79	**14.49**	9.20	7.31	9.94
VGGT [64]		10.48	8.43	7.04	13.94	8.89	6.27	8.67
Pi3 [68]		**12.08**	**11.80**	**8.82**	11.00	**10.60**	**8.86**	**10.16**
Ours w/ Naïve Fusion								
+ MVSplat		13.76	12.69	7.71	14.44	11.19	10.63	11.74
+ DepthSplat		**14.64**	12.15	**11.00**	**16.85**	11.27	10.88	**12.80**
+ VGGT		13.30	11.69	8.68	14.08	**11.30**	10.55	11.60
+ Pi3		13.60	**12.99**	8.99	13.35	**11.30**	**10.97**	11.87

4.6 Ablation Studies

Compatibility with geometry priors. A key advantage of GeoNVS is its plug-and-play compatibility with diverse feed-forward geometry models without requiring any additional training. To validate this, we integrate GS-Adapter with four different geometry priors [9, 64, 68, 77] and evaluate using SEVA [87] as the generative backbone in Tab. 5. Both fusion variants consistently improve over the SEVA baseline regardless of the geometry prior used. However, Naïve Fusion remains sensitive to prior quality: while VGGT yields strong gains on T&T (+3.37 dB), Pi3 degrades below the baseline on DTU (14.83 vs. 17.03), as observed in Fig. 4. Adaptive Fusion substantially mitigates this sensitivity, achieving robust and consistently superior results across all priors (19.02–19.18 dB on DTU; 19.22–19.26 dB on CO3D), and enabling flexible substitution of the geometry backbone without performance degradation.

GeoNVS with CameraCtrl. To demonstrate that GS-Adapter generalizes beyond SEVA, we integrate it into CameraCtrl [23] in Tab. 6. Standalone geometry

Table 7 Model ablation study of GeoNVS. We ablate the key components of GS-Adapter, including feature refinement, feature fusion strategy, and multi-scale fusion. All experiments use VGGT [64] as the geometry prior. Here, feats abbreviates features.

Method	GS-Adapter				Mip360		DL3DV		T&T	
	Refinement		Fusion		6-view		6-view		3-view	
	$\mathcal{L}_{\text{feat}}$	GS-PE	Naïve	Adaptive	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑
Geometry prior										
VGGT [64]					15.49	0.362	17.09	0.52	23.96	0.81
Baseline										
SEVA [87]					15.38	0.339	15.84	0.45	18.85	0.61
+ Input-level injection					15.40	0.373	16.73	0.52	**23.05**	**0.74**
+ LoRA only					15.55	0.360	17.35	0.54	21.41	0.68
+ GS-Adapter (Encoder-1,2,3 feats)	✓		✓		16.36	0.378	17.80	0.55	21.67	0.69
		✓	✓		16.47	0.378	17.88	0.55	22.11	0.69
	✓	✓	✓		16.48	**0.383**	17.95	**0.56**	22.22	0.70
Multi-scale fusion										
+ GS-Adapter (Encoder-1 feat)	✓	✓	✓		16.28	0.376	17.49	0.53	21.36	0.68
+ GS-Adapter (Encoder-1,2 feats)	✓	✓	✓		16.31	0.370	17.53	0.53	21.73	0.68
+ GS-Adapter (Encoder-1,2,3 feats)	✓	✓	✓		16.48	**0.383**	17.95	**0.56**	22.22	0.70
Feature Refinement										
− RefineNet, $\mathcal{R}$				✓	16.23	0.376	17.71	0.45	21.91	0.69
Ours (Encoder-1,2,3 feats)	✓	✓		✓	**16.57**	**0.383**	**18.11**	**0.56**	22.24	0.69

priors consistently underperform the CameraCtrl baseline, suggesting that geometry alone is insufficient in sparse-view settings. GeoNVS with Naïve Fusion improves over both the baseline and the standalone priors across most settings, achieving up to $+\mathbf{1.66}$ PSNR gain on average over the baseline, suggesting that GS-Adapter provides complementary benefit even on top of a strong geometry prior and is applicable beyond a single diffusion backbone.

Analysis of model components. Table 7 validates the contribution of each design choice in GS-Adapter. Both the feature consistency loss $\mathcal{L}_{\text{feat}}$ and Gaussian positional encoding (GS-PE) contribute to performance, as removing either component leads to consistent drops across benchmarks, indicating that explicit 3D structural cues and fine-grained supervision are both necessary for recovering details lost during the lifting-and-rendering process. RefineNet consistently yields performance gains, demonstrating its independent contribution beyond 3D-GS splatting. Finally, using multi-scale features together in GeoNVS provides consistent improvements, indicating that geometry-guided features across multiple spatial resolutions are complementary.

Comparison with input-level injection of geometry. As shown in Tab. 4, input-level injection of geometry priors causes severe degradation in camera controllability and geometric consistency, performing even worse than SEVA, a purely generative baseline. This failure stems from the fact that rasterized images introduce view-dependent color noise that corrupts the structural signal rather than reinforcing it. In contrast, the proposed GS-Adapter leverages only the structural information encoded in 3D Gaussians to modulate internal diffusion features, consistently reducing camera pose errors and Chamfer Distance across all datasets. Tab. 7 further reinforces this finding: input-level injection yields synthesis quality gains only when the geometry prior is near-perfect, whereas GS-Adapter consistently improves PSNR regardless of prior quality. These results collectively demonstrate that feature-level geometry modulation is both more robust and more effective than image-level injection.

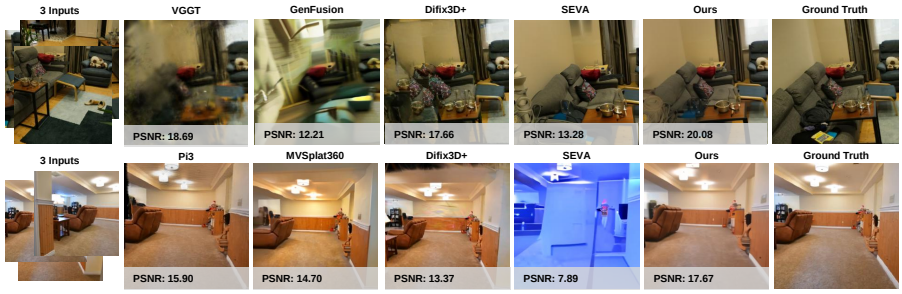


Fig. 6 Qualitative results of GeoNVS with SEVA [87].

Fig. 7 Qualitative results of GeoNVS with CameraCtrl [23]

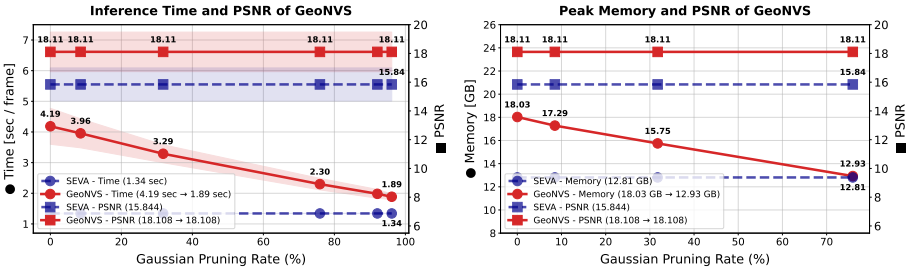


Fig. 8 Runtime and memory analysis under Gaussian pruning. Inference time, peak GPU memory, and PSNR of GeoNVS versus Gaussian pruning rate across 10 DL3DV [45] scenes. Voxel-based pruning [31] yields a 2.22× speedup at 96.2% pruning. Peak memory drops to 12.93 GB, close to SEVA [87], while PSNR is preserved.

Pose-free experiment. We extend GeoNVS to unknown pose scenarios by integrating ViPE [26], which outperforms VGGT [64] in robustness on complex scenes. Given poses estimated by ViPE from input images, we optimize 3D-GS using the VGGT point cloud, with scale aligned to ViPE poses solved via Eq. (1). As shown in the right table, GeoNVS maintains high synthesis quality without ground-truth intrinsics or extrinsics. On T&T [37], the ViPE pose setting slightly surpasses the GT setting, likely because ViPE estimates poses more robustly than the noisy COLMAP [59, 60]-derived poses.

Dataset (P)	DL3DV (6)			T&T (3)		
Metric	T_{err}	R_{err}	PSNR	T_{err}	R_{err}	PSNR
w/ GT Pose						
SEVA [87]	19.68	2.80	15.64	6.08	2.87	18.63
Ours	**13.27**	**2.01**	**17.99**	3.56	**1.52**	22.24
w/ ViPE [26] Pose						
SEVA [87]	25.11	3.54	15.38	2.73	4.87	19.24
Ours	20.64	2.98	17.87	**1.05**	2.40	**22.31**

Runtime and memory efficiency. We measure inference time, peak GPU memory, and PSNR across 10 scenes from DL3DV [45] to analyze the computational overhead relative to SEVA (Fig. 8). The primary bottleneck is the number of Gaussians from the geometry prior, which raises inference time to 4.19 sec/frame and peak memory to 18.03 GB. Applying voxel-based Gaussian pruning [31] reduces these to 1.89 sec/frame (2.22× speedup) and 12.93 GB, on par with SEVA's 12.81 GB, while maintaining a PSNR advantage over SEVA.

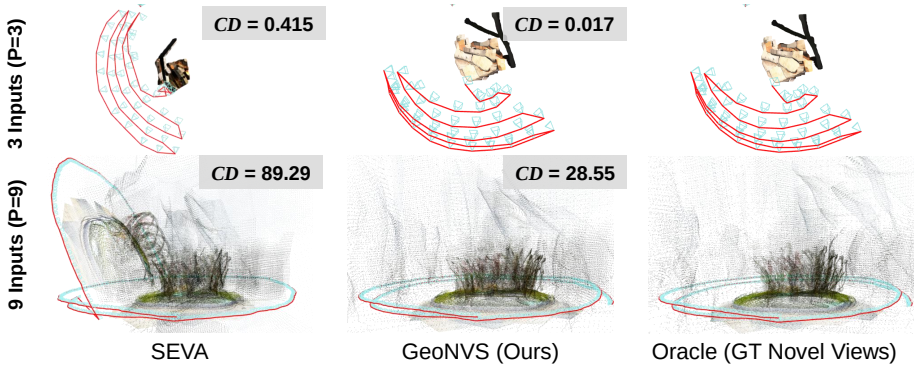


Fig. 9 Qualitative results of GeoNVS. 3D reconstruction and camera trajectory estimated from the generated video, showing the improved geometric fidelity of GeoNVS.

Fig. 10 Limitation: blurry textures. Our method preserves structure well but occasionally produces blurry textures in fine-detail regions (red boxes).

5 Conclusions

We present GeoNVS, a geometry-grounded novel-view synthesizer that enhances geometric fidelity and camera controllability using GS-Adapter. Unlike prior methods that inject geometry at the input level, GS-Adapter modulates internal diffusion features using 3D Gaussian priors in a plug-and-play manner, enabling zero-shot compatibility with any geometry model and diffusion backbone. Extensive experiments across 9 scenes and 18 settings demonstrate state-of-the-art performance, achieving 11.3% and 14.9% improvements over SEVA and CameraCtrl, with up to 2× reduction in translation error and 7× in Chamfer Distance. GeoNVS improves synthesis in non-co-visible regions by up to 2.08 dB, confirming that feature-level 3D grounding facilitates extrapolation beyond observed regions. Finally, GeoNVS supports pose-free scenarios via SfM methods [17, 26] and reduces inference overhead by 2.2× with negligible performance degradation.

Limitations and future work. While our proposed method peforms well in sparse-view and long-range trajectory NVS, its generation ability degrades in regions distant from input views. We attribute this to increasingly uncertain feature rendering in GS-Adapter as the distance from input views grows. Our method occasionally produces blurry textures due to information loss during feature lifting, which can reduce perceptual sharpness despite well-preserved structure (Fig. 10). We leave addressing these limitations to future work.

Acknowledgements. This work was supported by the InnoCORE program of the Ministry of Science and ICT (26-InnoCORE-01), and by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (RS-2026-25473963).

References

1. Bai, J., Xia, M., Fu, X., Wang, X., Mu, L., Cao, J., Liu, Z., Hu, H., Bai, X., Wan, P., et al.: Recammaster: Camera-controlled generative rendering from a single video. In: ICCV. pp. 14834–14844 (2025)
2. Barron, J.T., Mildenhall, B., Verbin, D., Srinivasan, P.P., Hedman, P.: Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In: CVPR. pp. 5470–5479 (2022)
3. Blattmann, A., Dockhorn, T., Kulal, S., Mendelevitch, D., Kilian, M., Lorenz, D., Levi, Y., English, Z., Voleti, V., Letts, A., et al.: Stable video diffusion: Scaling latent video diffusion models to large datasets. arXiv preprint arXiv:2311.15127 (2023)
4. Cao, C., Yu, C., Liu, S., Wang, F., Xue, X., Fu, Y.: Mvgenmaster: Scaling multi-view generation from any image via 3d priors enhanced diffusion model. In: CVPR. pp. 6045–6056 (2025)
5. Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., Joulin, A.: Emerging properties in self-supervised vision transformers. In: ICCV. pp. 9650–9660 (2021)
6. Chan, E.R., Nagano, K., Chan, M.A., Bergman, A.W., Park, J.J., Levy, A., Aittala, M., De Mello, S., Karras, T., Wetzstein, G.: Generative novel view synthesis with 3d-aware diffusion models. In: ICCV. pp. 4217–4229 (2023)
7. Charatan, D., Li, S., Tagliasacchi, A., Sitzmann, V.: pixelsplat: 3d gaussian splats from image pairs for scalable generalizable 3d reconstruction. In: CVPR (2024)
8. Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. IEEE transactions on pattern analysis and machine intelligence **40**(4), 834–848 (2017)
9. Chen, Y., Xu, H., Zheng, C., Zhuang, B., Pollefeys, M., Geiger, A., Cham, T.J., Cai, J.: Mvsplat: Efficient 3d gaussian splatting from sparse multi-view images. In: ECCV. pp. 370–386. Springer (2024)
10. Chen, Y., Zheng, C., Xu, H., Zhuang, B., Vedaldi, A., Cham, T.J., Cai, J.: Mvsplat360: Feed-forward 360 scene synthesis from sparse views. In: NeurIPS (2024)
11. Chung, J., Oh, J., Lee, K.M.: Depth-regularized optimization for 3d gaussian splatting in few-shot images. In: CVPR. pp. 811–820 (2024)
12. Deng, K., Liu, A., Zhu, J.Y., Ramanan, D.: Depth-supervised nerf: Fewer views and faster training for free. In: CVPR. pp. 12882–12891 (2022)
13. Duzceker, A., Galliani, S., Vogel, C., Speciale, P., Dusmanu, M., Pollefeys, M.: Deepvideomvs: Multi-view stereo on video with recurrent spatio-temporal fusion. In: CVPR. pp. 15324–15333 (2021)
14. Eftekhar, A., Sax, A., Malik, J., Zamir, A.: Omnidata: A scalable pipeline for making multi-task mid-level vision datasets from 3d scans. In: ICCV. pp. 10786–10796 (2021)
15. Fan, H., Su, H., Guibas, L.J.: A point set generation network for 3d object reconstruction from a single image. In: CVPR. pp. 605–613 (2017)
16. Fan, Z., Wen, K., Cong, W., Wang, K., Zhang, J., Ding, X., Xu, D., Ivanovic, B., Pavone, M., Pavlakos, G., et al.: Instantsplat: Sparse-view gaussian splatting in seconds. arXiv preprint arXiv:2403.20309 (2024)
17. Fu, Y., Liu, S., Kulkarni, A., Kautz, J., Efros, A.A., Wang, X.: Colmap-free 3d gaussian splatting. In: CVPR. pp. 20796–20805 (2024)
18. Gao*, R., Holynski*, A., Henzler, P., Brussee, A., Martin-Brualla, R., Srinivasan, P.P., Barron, J.T., Poole*, B.: Cat3d: Create anything in 3d with multi-view diffusion models. NeurIPS (2024)

19. Geng, D., Herrmann, C., Hur, J., Cole, F., Zhang, S., Pfaff, T., Lopez-Guevara, T., Aytar, Y., Rubinstein, M., Sun, C., et al.: Motion prompting: Controlling video generation with motion trajectories. In: CVPR. pp. 1–12 (2025)
20. Gu, X., Fan, Z., Zhu, S., Dai, Z., Tan, F., Tan, P.: Cascade cost volume for high-resolution multi-view stereo and stereo matching. In: CVPR. pp. 2495–2504 (2020)
21. Gu, Z., Yan, R., Lu, J., Li, P., Dou, Z., Si, C., Dong, Z., Liu, Q., Lin, C., Liu, Z., et al.: Diffusion as shader: 3d-aware video diffusion for versatile video generation control. In: Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers. pp. 1–12 (2025)
22. Hanson, A., Tu, A., Singla, V., Jayawardhana, M., Zwicker, M., Goldstein, T.: Pup 3d-gs: Principled uncertainty pruning for 3d gaussian splatting. In: CVPR. pp. 5949–5958 (2025)
23. He, H., Xu, Y., Guo, Y., Wetzstein, G., Dai, B., Li, H., Yang, C.: Cameractrl: Enabling camera control for video diffusion models. In: ICLR (2025)
24. He, H., Yang, C., Lin, S., Xu, Y., Wei, M., Gui, L., Zhao, Q., Wetzstein, G., Jiang, L., Li, H.: Cameractrl ii: Dynamic scene exploration via camera-controlled video diffusion models. arXiv preprint arXiv:2503.10592 (2025)
25. Hu, E.J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W., et al.: Lora: Low-rank adaptation of large language models. ICLR **1**(2), 3 (2022)
26. Huang, J., Zhou, Q., Rabeti, H., Korovko, A., Ling, H., Ren, X., Shen, T., Gao, J., Slepichev, D., Lin, C.H., et al.: Vipe: Video pose engine for 3d geometric perception. arXiv preprint arXiv:2508.10934 (2025)
27. Huang, S.Y., Choe, J., Wang, Y.C.F., Sun, C.: Openvoxel: Training-free grouping and captioning voxels for open-vocabulary 3d scene understanding. arXiv preprint arXiv:2601.09575 (2026)
28. Im, S., Jeon, H.G., Lin, S., Kweon, I.S.: Dpsnet: End-to-end deep plane sweep stereo. arXiv preprint arXiv:1905.00538 (2019)
29. Jensen, R., Dahl, A., Vogiatzis, G., Tola, E., Aanæs, H.: Large scale multi-view stereopsis evaluation. In: CVPR. pp. 406–413. IEEE (2014)
30. Ji, M., Qiu, R.Z., Zou, X., Wang, X.: Graspsplats: Efficient manipulation with 3d feature splatting. arXiv preprint arXiv:2409.02084 (2024)
31. Jiang, L., Mao, Y., Xu, L., Lu, T., Ren, K., Jin, Y., Xu, X., Yu, M., Pang, J., Zhao, F., et al.: Anysplat: Feed-forward 3d gaussian splatting from unconstrained views. arXiv preprint arXiv:2505.23716 (2025)
32. Jun-Seong, K., GeonU, K., Yu-Ji, K., Wang, Y.C.F., Choe, J., Oh, T.H.: Dr. splat: Directly referring 3d gaussian splatting via direct language embedding registration. In: CVPR (2025)
33. Keetha, N., Müller, N., Schönberger, J., Porzi, L., Zhang, Y., Fischer, T., Knapitsch, A., Zauss, D., Weber, E., Antunes, N., Luiten, J., Lopez-Antequera, M., Bulò, S.R., Richardt, C., Ramanan, D., Scherer, S., Kontschieder, P.: MapAnything: Universal feed-forward metric 3D reconstruction. In: International Conference on 3D Vision (3DV). IEEE (2026)
34. Kerbl, B., Kopanas, G., Leimkühler, T., Drettakis, G.: 3d gaussian splatting for real-time radiance field rendering. ACM Transactions on Graphics **42**(4) (July 2023), https://repo-sam.inria.fr/fungraph/3d-gaussian-splatting/
35. Kerr, J., Kim, C.M., Goldberg, K., Kanazawa, A., Tancik, M.: Lerf: Language embedded radiance fields. In: ICCV. pp. 19729–19739 (2023)
36. Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W.Y., et al.: Segment anything. In: ICCV. pp. 4015–4026 (2023)
37. Knapitsch, A., Park, J., Zhou, Q.Y., Koltun, V.: Tanks and temples: Benchmarking large-scale scene reconstruction. ACM Transactions on Graphics **36**(4) (2017)

38. Lee, J., Park, C., Choe, J., Wang, Y.C.F., Kautz, J., Cho, M., Choy, C.: Mosaic3d: Foundation dataset and model for open-vocabulary 3d segmentation. In: CVPR. pp. 14089–14101 (2025)
39. Li, L., Zhang, Z., Li, Y., Xu, J., Hu, W., Li, X., Cheng, W., Gu, J., Xue, T., Shan, Y.: Nvcomposer: Boosting generative novel view synthesis with multiple sparse and unposed images. In: CVPR. pp. 777–787 (2025)
40. Li, M., Yang, T., Kuang, H., Wu, J., Wang, Z., Xiao, X., Chen, C.: Controlnet++: Improving conditional controls with efficient consistency feedback: Project page: liming-ai. github. io/controlnet_plus_plus. In: ECCV. pp. 129–147. Springer (2024)
41. Li, W., Zhao, Y., Qin, M., Liu, Y., Cai, Y., Gan, C., Pfister, H.: Langsplatv2: High-dimensional 3d language gaussian splatting with 450+ fps. Advances in Neural Information Processing Systems (2025), https://arxiv.org/abs/2507.07136
42. Liang, S., Wang, S., Li, K., Niemeyer, M., Gasperini, S., Lensch, H., Navab, N., Tombari, F.: Supergseg: Open-vocabulary 3d segmentation with structured super-gaussians. arXiv preprint arXiv:2412.10231 (2024)
43. Lin, H., Chen, S., Liew, J., Chen, D.Y., Li, Z., Shi, G., Feng, J., Kang, B.: Depth anything 3: Recovering the visual space from any views. arXiv preprint arXiv:2511.10647 (2025)
44. Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: CVPR. pp. 2117–2125 (2017)
45. Ling, L., Sheng, Y., Tu, Z., Zhao, W., Xin, C., Wan, K., Yu, L., Guo, Q., Yu, Z., Lu, Y., et al.: Dl3dv-10k: A large-scale scene dataset for deep learning-based 3d vision. In: CVPR. pp. 22160–22169 (2024)
46. Liu, D., Li, S., Liu, Y., Li, Z., Wang, K., Li, X., Qin, Q., Liu, Y., Xin, Y., Li, Z., et al.: Lumina-video: Efficient and flexible video generation with multi-scale next-dit. arXiv preprint arXiv:2502.06782 (2025)
47. Liu, X., Zhou, C., Huang, S.: 3dgs-enhancer: Enhancing unbounded 3d gaussian splatting with view-consistent 2d diffusion priors. NeurIPS **37**, 133305–133327 (2024)
48. Marrie, J., Ménégaux, R., Arbel, M., Larlus, D., Mairal, J.: Ludvig: Learning-free uplifting of 2d visual features to gaussian splatting scenes. In: ICCV. pp. 7440–7450 (2025)
49. Meng, C., He, Y., Song, Y., Song, J., Wu, J., Zhu, J.Y., Ermon, S.: Sdedit: Guided image synthesis and editing with stochastic differential equations. arXiv preprint arXiv:2108.01073 (2021)
50. Mildenhall, B., Srinivasan, P.P., Ortiz-Cayon, R., Kalantari, N.K., Ramamoorthi, R., Ng, R., Kar, A.: Local light field fusion: Practical view synthesis with prescriptive sampling guidelines. ACM Transactions on Graphics (ToG) **38**(4), 1–14 (2019)
51. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. Communications of the ACM **65**(1), 99–106 (2021)
52. Oquab, M., Darcet, T., Moutakanni, T., Vo, H., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., et al.: Dinov2: Learning robust visual features without supervision. arXiv preprint arXiv:2304.07193 (2023)
53. Qin, M., Li, W., Zhou, J., Wang, H., Pfister, H.: Langsplat: 3d language gaussian splatting. In: CVPR. pp. 20051–20060 (2024)
54. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: ICML. pp. 8748–8763. PmLR (2021)

55. Ranftl, R., Bochkovskiy, A., Koltun, V.: Vision transformers for dense prediction. In: ICCV. pp. 12179–12188 (2021)
56. Reizenstein, J., Shapovalov, R., Henzler, P., Sbordone, L., Labatut, P., Novotny, D.: Common objects in 3d: Large-scale learning and evaluation of real-life 3d category reconstruction. In: CVPR. pp. 10901–10911 (2021)
57. Ren, X., Shen, T., Huang, J., Ling, H., Lu, Y., Nimier-David, M., Müller, T., Keller, A., Fidler, S., Gao, J.: Gen3c: 3d-informed world-consistent video generation with precise camera control. In: CVPR. pp. 6121–6132 (2025)
58. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: CVPR. pp. 10684–10695 (2022)
59. Schönberger, J.L., Frahm, J.M.: Structure-from-motion revisited. In: CVPR (2016)
60. Schönberger, J.L., Zheng, E., Pollefeys, M., Frahm, J.M.: Pixelwise view selection for unstructured multi-view stereo. In: ECCV (2016)
61. Seo, S., Chang, Y., Kwak, N.: Flipnerf: Flipped reflection rays for few-shot novel view synthesis. In: ICCV. pp. 22883–22893 (2023)
62. Tang, S., Ye, W., Ye, P., Lin, W., Zhou, Y., Chen, T., Ouyang, W.: Hisplat: Hierarchical 3d gaussian splatting for generalizable sparse-view reconstruction. In: ICLR (2025)
63. Wang, G., Chen, Z., Loy, C.C., Liu, Z.: Sparsenerf: Distilling depth ranking for few-shot novel view synthesis. In: ICCV. pp. 9065–9076 (2023)
64. Wang, J., Chen, M., Karaev, N., Vedaldi, A., Rupprecht, C., Novotny, D.: Vggt: Visual geometry grounded transformer. In: CVPR. pp. 5294–5306 (2025)
65. Wang, J., Sun, K., Cheng, T., Jiang, B., Deng, C., Zhao, Y., Liu, D., Mu, Y., Tan, M., Wang, X., et al.: Deep high-resolution representation learning for visual recognition. IEEE transactions on pattern analysis and machine intelligence **43**(10), 3349–3364 (2020)
66. Wang, S., Zhang, S., Millerdurai, C., Westermann, R., Stricker, D., Pagani, A.: Inpaint360gs: Efficient object-aware 3d inpainting via gaussian splatting for 360 {\deg} scenes. arXiv preprint arXiv:2511.06457 (2025)
67. Wang, W., Xie, E., Li, X., Fan, D.P., Song, K., Liang, D., Lu, T., Luo, P., Shao, L.: Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In: ICCV. pp. 568–578 (2021)
68. Wang, Y., Zhou, J., Zhu, H., Chang, W., Zhou, Y., Li, Z., Chen, J., Pang, J., Shen, C., He, T.: π^3: Scalable permutation-equivariant visual geometry learning (2025), https://arxiv.org/abs/2507.13347
69. Wang, Z., Yuan, Z., Wang, X., Li, Y., Chen, T., Xia, M., Luo, P., Shan, Y.: Motionctrl: A unified and flexible motion controller for video generation. In: SIGGRAPH. pp. 1–11 (2024)
70. Wu, H., Wu, D., He, T., Guo, J., Ye, Y., Duan, Y., Bian, J.: Geometry forcing: Marrying video diffusion and 3d representation for consistent world modeling. arXiv preprint arXiv:2507.07982 (2025)
71. Wu, J.Z., Zhang, Y., Turki, H., Ren, X., Gao, J., Shou, M.Z., Fidler, S., Gojcic, Z., Ling, H.: Difix3d+: Improving 3d reconstructions with single-step diffusion models. In: CVPR. pp. 26024–26035 (2025)
72. Wu, R., Mildenhall, B., Henzler, P., Park, K., Gao, R., Watson, D., Srinivasan, P.P., Verbin, D., Barron, J.T., Poole, B., et al.: Reconfusion: 3d reconstruction with diffusion priors. In: CVPR. pp. 21551–21561 (2024)
73. Wu, S., Xu, C., Huang, B., Geiger, A., Chen, A.: Genfusion: Closing the loop between reconstruction and generation via videos. In: CVPR. pp. 6078–6088 (2025)
74. Wu, T., Zhang, J., Fu, X., Wang, Y., Ren, J., Pan, L., Wu, W., Yang, L., Wang, J., Qian, C., et al.: Omniobject3d: Large-vocabulary 3d object dataset for realistic perception, reconstruction and generation. In: CVPR. pp. 803–814 (2023)

75. Wu, W., Li, Z., Gu, Y., Zhao, R., He, Y., Zhang, D.J., Shou, M.Z., Li, Y., Gao, T., Zhang, D.: Draganything: Motion control for anything using entity representation. In: ECCV. pp. 331–348. Springer (2024)
76. Xia, H., Fu, Y., Liu, S., Wang, X.: Rgbd objects in the wild: Scaling real-world 3d object learning from rgb-d videos. In: CVPR. pp. 22378–22389 (2024)
77. Xu, H., Peng, S., Wang, F., Blum, H., Barath, D., Geiger, A., Pollefeys, M.: Depthsplat: Connecting gaussian splatting and depth. In: CVPR. pp. 16453–16463 (2025)
78. Yang, L., Kang, B., Huang, Z., Xu, X., Feng, J., Zhao, H.: Depth anything: Unleashing the power of large-scale unlabeled data. In: CVPR. pp. 10371–10381 (2024)
79. Yao, Y., Luo, Z., Li, S., Fang, T., Quan, L.: Mvsnet: Depth inference for unstructured multi-view stereo. ECCV (2018)
80. Yin, S., Wu, C., Liang, J., Shi, J., Li, H., Ming, G., Duan, N.: Dragnuwa: Fine-grained control in video generation by integrating text, image, and trajectory. arXiv preprint arXiv:2308.08089 (2023)
81. Yin, W., Zhang, C., Chen, H., Cai, Z., Yu, G., Wang, K., Chen, X., Shen, C.: Metric3d: Towards zero-shot metric 3d prediction from a single image. In: CVPR. pp. 9043–9053 (2023)
82. Yin, X., Zhang, Q., Chang, J., Feng, Y., Fan, Q., Yang, X., Pun, C.M., Zhang, H., Cun, X.: Gsfixer: Improving 3d gaussian splatting with reference-guided video diffusion priors. arXiv preprint arXiv:2508.09667 (2025)
83. Yu, W., Xing, J., Yuan, L., Hu, W., Li, X., Huang, Z., Gao, X., Wong, T.T., Shan, Y., Tian, Y.: Viewcrafter: Taming video diffusion models for high-fidelity novel view synthesis. arXiv preprint arXiv:2409.02048 (2024)
84. Yu, Z., Peng, S., Niemeyer, M., Sattler, T., Geiger, A.: Monosdf: Exploring monocular geometric cues for neural implicit surface reconstruction. NeurIPS **35**, 25018–25032 (2022)
85. Zhang, L., Rao, A., Agrawala, M.: Adding conditional control to text-to-image diffusion models. In: ICCV. pp. 3836–3847 (2023)
86. Zhang, Z., Liao, J., Li, M., Dai, Z., Qiu, B., Zhu, S., Qin, L., Wang, W.: Tora: Trajectory-oriented diffusion transformer for video generation. In: CVPR. pp. 2063–2073 (2025)
87. Zhou, J., Gao, H., Voleti, V., Vasishta, A., Yao, C.H., Boss, M., Torr, P., Rupprecht, C., Jampani, V.: Stable virtual camera: Generative view synthesis with diffusion models. In: ICCV. pp. 12405–12414 (2025)
88. Zhou, S., Chang, H., Jiang, S., Fan, Z., Zhu, Z., Xu, D., Chari, P., You, S., Wang, Z., Kadambi, A.: Feature 3dgs: Supercharging 3d gaussian splatting to enable distilled feature fields. In: CVPR. pp. 21676–21685 (2024)
89. Zhou, T., Tucker, R., Flynn, J., Fyffe, G., Snavely, N.: Stereo magnification: Learning view synthesis using multiplane images. arXiv preprint arXiv:1805.09817 (2018)
90. Zhu, Z., Fan, Z., Jiang, Y., Wang, Z.: Fsgs: Real-time few-shot view synthesis using gaussian splatting. In: ECCV. pp. 145–163. Springer (2024)