

# From Rays to Projections: Better Inputs for Feed-Forward View Synthesis

Zirui Wu<sup>1</sup> Zeren Jiang<sup>2</sup> Martin R. Oswald<sup>3</sup> Jie Song<sup>1,4</sup>

<sup>1</sup>HKUST (GZ) <sup>2</sup>University of Oxford <sup>3</sup>University of Amsterdam <sup>4</sup>HKUST

<https://wuzirui.github.io/pvsm-web>

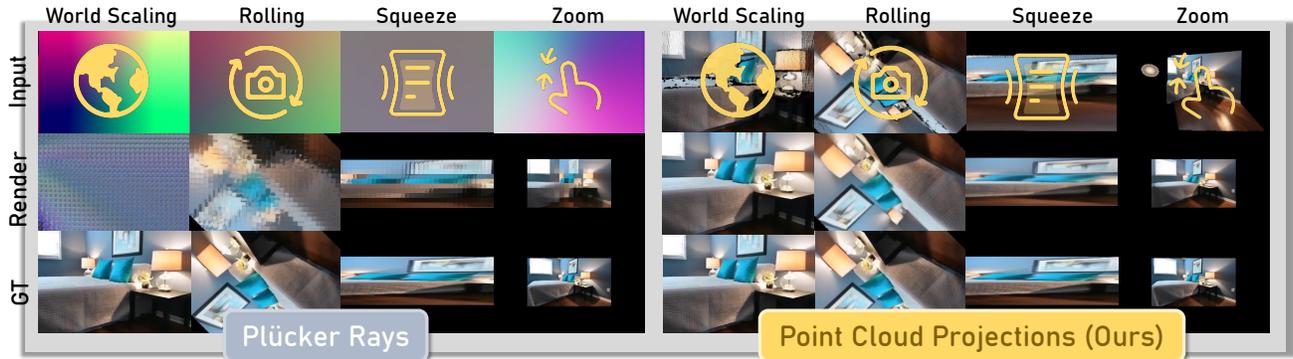


Figure 1. **Projective conditioning enables robust novel view synthesis.** We investigate what camera pose encoding best conditions a view synthesis model. Compared with the commonly used absolute-valued Plücker ray conditioning of prior works [9, 11], the projection-based signal encodes scene-camera configuration through a target-view projective cue. Under various geometric transformations, it remains robust, whereas the absolute conditioning signal fails because transformations are not smooth in Plücker ray space.

## Abstract

*Feed-forward view synthesis models predict a novel view in a single pass with minimal 3D inductive bias. Existing works encode cameras as Plücker ray maps, which tie predictions to the arbitrary world coordinate gauge and make them sensitive to small camera transformations, thereby undermining geometric consistency. In this paper, we ask what inputs best condition a model for robust and consistent view synthesis. We propose projective conditioning, which replaces raw camera parameters with a target-view projective cue that provides a stable 2D input. This reframes the task from a brittle geometric regression problem in ray space to a well-conditioned target-view image-to-image translation problem. Additionally, we introduce a masked autoencoding pretraining strategy tailored to this cue, enabling the use of large-scale uncalibrated data for pretraining. Our method shows improved fidelity and stronger cross-view consistency compared to ray-conditioned baselines on our view-consistency benchmark. It also achieves state-of-the-art quality on standard novel view synthesis benchmarks.*

## 1. Introduction

Synthesizing realistic novel views from a set of captured context images is a long-standing goal in computer vision

and graphics. Recent feed-forward models [3, 11, 26, 42] leverage data-driven priors to directly render novel views in a single forward pass, by conditioning the model on a few context views and the target camera frustum.

In particular, the recent large view-synthesis models (LVSMs) [9, 11, 30] employ vision transformers (ViTs) [5] and encode camera parameters using Plücker ray embeddings [23] as their input space representation. While this interface is convenient, it can also introduce brittleness. The absolute coordinate representation makes the model sensitive, such that even minor adjustments to the camera can lead to significant shifts in the 6D ray space, despite only small visual changes. This sensitivity can lead to inputs straying from the model’s training distribution, which in turn degrades 3D consistency, as illustrated in Fig. 1. Basic transformations, such as zooming and squeezing, can be combined to create complex camera movements. For example, simultaneously adjusting the zoom while moving the camera produces the dolly zoom effect (see Suppl. video). However, existing models struggle to robustly handle these fundamental operations, resulting in artifacts that violate geometric consistency.

In this work, we investigate the input representations for feed-forward novel view synthesis and pose the question: *What inputs best condition a model for robust and consistent view synthesis?* Instead of directly encoding camera param-

eters, we condition the model on a *target-view point cloud projection image*. This image is generated by first extracting depth maps from context views using off-the-shelf perception models [13] and then rasterizing the unprojected point cloud into the target camera view. This design delegates camera handling to a deterministic geometry engine, allowing the model to operate within a stable 2D image domain.

The advantages of this representation are twofold: *First*, minor adjustments to the camera lead to correspondingly small and localized changes in the point cloud image, facilitating more precise camera control and enhancing robustness to various camera transformations. This approach reframes novel view synthesis as an image-to-image mapping conditioned on coherent visual cues, ultimately resulting in stronger 3D consistency across different viewpoints. *Second*, since projective conditioning operates on 2D buffers, it can be seamlessly combined with Masked Auto-Encoding (MAE)-style pretraining [8, 33]. We leverage the observation that the task of reconstructing a randomly masked target image structurally mirrors our view synthesis goal and propose a self-supervised pretraining strategy, which allows the model to learn a powerful cross-view completion prior.

To evaluate rendering consistency, we establish a view consistency benchmark using common camera transformations in Fig. 1, applied to the target view. Our experiments reveal that, with identical backbones and training schedules, projective conditioning exhibits enhanced fidelity and stronger cross-view consistency compared to ray-based conditioning. Thanks to the rapid advancements in recent geometric foundation models [13, 18, 31, 32], we can now extract high-quality 3D annotations from uncalibrated images. To facilitate reproducible training and evaluation at scale, we curate a refined derivative of RealEstate10K [48] with dense depth and refined camera parameters from MapAnything [13], which we use throughout the projective-conditioning experiments and consistency benchmark.

Our contributions can be summarized as follows:

- We analyze the instability of Plücker ray conditioning and show that target-view projective conditioning is a substantially more stable conditioning interface for deterministic feed-forward view synthesis. This reframes view synthesis as an image-to-image mapping, enhancing camera control and robustness to various transformations. Extensive experiments demonstrate that our model outperforms ray-conditioned baselines in both view consistency and novel view synthesis quality. Code, data, and models will be made publicly available.
- We introduce a MAE pretraining stage that leverages the 2D nature of projective conditioning to learn scene completion priors in a self-supervised manner, thereby reducing dependence on expensive 3D annotations.
- We contribute a refined RealEstate10K annotation set for reproducible projective-conditioning experiments, along

with a view consistency benchmark for rigorous evaluation of robustness.

## 2. Related Works

**Neural Rendering.** Novel view synthesis has long been a central task in 3D computer vision. Neural representations such as neural radiance fields (NeRFs) [1, 17, 22, 37] and 3D Gaussian Splatting [2, 20, 34, 36, 39, 40, 47] have achieved impressive results with volumetric scene parameterizations. Their main limitation is the need for per-scene optimization, which leads to high computation cost and weak generalization beyond the training scene.

**Feed-forward View Synthesis.** To alleviate this cost, feed-forward alternatives have gained traction. pixelNeRF [42] conditions NeRFs on context views without to predict novel views in a single forward pass instead of optimizing a scene-specific model. Follow-up work on feed-forward 3D Gaussians [3, 4, 10, 18, 38] regresses pixel-aligned Gaussian primitives from context views and splats them into the target camera. More recent methods move toward implicit networks that avoid explicit 3D structures. LVSM [11] and its extensions [9, 14, 30] directly map camera ray embeddings to target RGBs, demonstrating strong scalability. Our method instead operates entirely in the 2D image domain, predicting the target view from a projective point-cloud cue rendered in the target camera, without explicit pose inputs or assumptions about an underlying 3D representation. Earlier systems such as *LookinGood* [21] and *Infinite Nature* [16], as well as recent geometry-guided video generation works [7, 24, 35, 43, 44], also make use of projective or reprojected cues. Our work brings this design into deterministic feed-forward novel view synthesis and evaluates it directly against Plücker-ray conditioning under matched backbones and training schedules.

**Self-supervised Learning.** Self-supervised learning (SSL) [8, 25, 28, 33] is an appealing way to exploit large-scale unlabeled data. MAE [8] masks a large portion of image patches and trains the model to reconstruct them, yielding 2D priors that transfer well to downstream tasks. In 3D vision, where dense geometry is often scarce or noisy, SSL is especially valuable. CroCo [33] extends masked modeling to multi-view images, and RayZer [9] and Less3D [30] apply SSL to feed-forward view synthesis by mapping uncalibrated images to a latent SE(3) manifold and predicting target views from context images and latent cameras. This reframes novel view synthesis as an SSL problem and scales well, but also introduces two limitations: (1) the model can indirectly access the ground-truth target view through the latent-camera construction, which risks information leakage; and (2) the latent manifold is not aligned with a physical coordinate system, making precise viewpoint control difficult in practice. In contrast, we treat

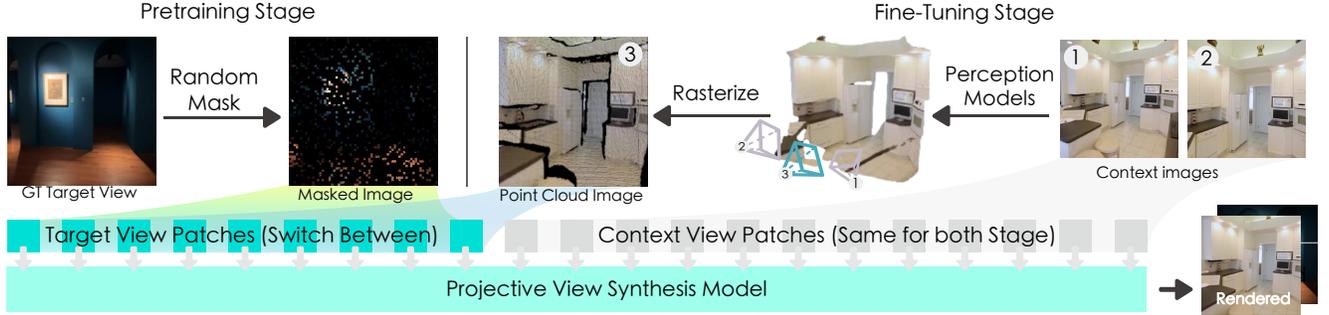


Figure 2. **An overview of our proposed two-stage training pipeline.** 1. **Pretraining:** This stage is self-supervised with the model conditioned on a set of context views and a randomly masked version of the target view itself (Masked Image). Its objective is to reconstruct the complete, original Ground Truth (GT) Target View. 2. **Fine-Tuning:** The context views are first unprojected into a unified 3D point cloud with extracted depth from perception models [13], which is then rasterized from the perspective of the target camera’s frustum to create a point cloud projection image that provides geometric cues. The model is then fine-tuned to generate the final target image.

the projected point-cloud cue as an explicit structural signal, pre-training with a MAE-style objective on DL3DV [15] and then fine-tuning the network to map this cue directly to the target RGB view.

### 3. Analysis of Ray Conditioning Instability

We begin by reviewing the core pipeline of the Large View Synthesis Model (LVSM) [11], which serves as the foundation for our analysis. LVSM formulates novel view synthesis as a single-pass, conditional image regression task using a Vision Transformer (ViT) architecture [5]. Given  $N^c$  context views  $\{\mathcal{I}_i^c\}_{i=1}^{N^c}$  with known camera parameters  $\{\mathcal{C}_i^c\}_{i=1}^{N^c}$ , and a target camera pose  $\mathcal{C}^t$ , the model predicts the target image  $\mathcal{I}^t$  via a mapping  $\hat{\mathcal{I}}^t = \mathcal{M}(\{\{\mathcal{I}_i^c, \mathcal{C}_i^c\}_{i=1}^{N^c}, \mathcal{C}^t\})$ . Central to this approach is its use of Plücker coordinates.

#### 3.1. Preliminaries: Large View Synthesis Models

**Plücker Ray Representation.** LVSM first converts each camera pose  $\mathcal{C} = (K, [R|t])$  (including the intrinsics  $K$  and extrinsics  $[R|t]$ ) into a per-pixel 6D Plücker coordinate. The viewing ray of a pixel is defined by the ray origin  $\mathbf{o}$  and normalized direction vector  $\mathbf{d}$ . The Plücker representation is then constructed as  $\mathbf{L} = (\mathbf{m}, \mathbf{d})$ , where  $\mathbf{m} = \mathbf{o} \times \mathbf{d}$  is the moment vector. All valid Plücker coordinates form the Klein quadric  $\mathcal{K}$ , which is the embedding of the Grassmannian manifold  $G(1, 3)$  into  $\mathbb{P}^5$  [6].

**Pipeline.** Both context images and their associated Plücker maps are split into non-overlapping  $p \times p$  patches and linearly embedded into tokens. The target view ray map is also split into patches and linearly embedded into tokens similarly. All tokens are concatenated and processed by a decoder-only Vision Transformer [5, 29], whose output tokens corresponding to the target view are decoded into RGB patches. While effective in practice, this approach encodes all geometric structure through absolute Plücker coordinates. As shown below, this induces instability under

coordinate-gauge changes.

#### 3.2. The Global SE(3) Invariance

A fundamental property any view-synthesis model should satisfy is the invariance to the choice of world coordinate gauge. Formally, the rendered image  $\mathcal{I}^t$  should remain unchanged if the entire scene geometry  $\mathcal{G}$  and all cameras  $\{\mathcal{C}_i\}$  are all transformed by a global transformation  $g \in \text{SE}(3)$ :

$$\mathcal{M}(\{\{\mathcal{I}_i^c, g \cdot \mathcal{C}_i^c\}_{i=1}^{N^c}, g \cdot \mathcal{C}^t\}) = \mathcal{M}(\{\{\mathcal{I}_i^c, \mathcal{C}_i^c\}_{i=1}^{N^c}, \mathcal{C}^t\}) \quad (1)$$

When expressed in Plücker space, the action  $\rho(g)$  corresponding to the transformation  $g = (R, \mathbf{t})$  applies as:

$$(\mathbf{m}', \mathbf{d}') = \rho(g)(\mathbf{m}, \mathbf{d}) = (R\mathbf{m} + [\mathbf{t}]_{\times} R\mathbf{d}, R\mathbf{d}). \quad (2)$$

This shows that even a uniform world-space transformation can cause non-uniform, spatially varying perturbations in Plücker coordinates. Each pixel-level ray token changes differently depending on its location, making the representation highly sensitive to the arbitrary chosen world gauge. In practice, identical relative camera-scene configurations can yield drastically different Plücker distributions if expressed in different coordinate frames.

As visualized in Fig. 3, a small random SE(3) perturbation can cause severe degradation in Plücker-conditioned models. This reveals that the network tends to overfit to dataset-specific coordinate gauges, leading to a noticeable train–test gap. Moreover, the absolute world reference frame carries no meaningful information for the rendering problem, so learning the invariance (e.g., by data augmentation) only wastes model capacity and training resources.

#### 3.3. Fine-grained Invariance

Beyond global rigid motions, rendering should remain invariant to broader transformations that preserve the underlying ray–scene relationships. For a given scene  $\mathcal{G}$  and ray  $\mathbf{r}$ , the

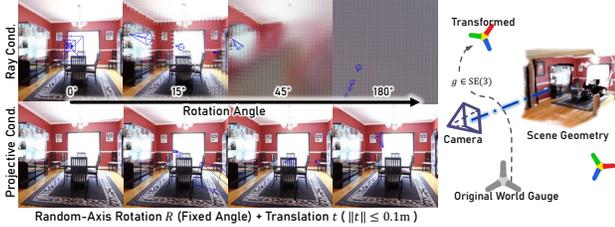


Figure 3. Under a random global  $SE(3)$  transformation, including both translation and rotation of the world coordinate system, ray-conditioned models [11] produce degenerate results while projective conditioning remains robust.

pixel color determined by their intersection should be unaffected by how the world coordinates, focals, or image lattice are parameterized.

To examine this property, we extend the analysis from global  $SE(3)$  motions to include world rescaling, focal-length variation, and image-plane rotation, which are common instances of  $Sim(3)$  and image-space resampling. Analogous to Eq. (2), these transformations act non-uniformly in Plücker space, yielding heterogeneous and non-local perturbations across tokens. Empirically, these perturbations translate to large performance drops as shown in Fig. 4. This further reinforces that Plücker rays encode an over-parameterized, gauge-dependent representation ill-suited for consistent rendering.

#### 4. Projective View Synthesis Model

To circumvent the instabilities of ray conditioning, we propose the Projective View Synthesis Model (PVSM, Fig. 2). Our key idea is to delegate the complex geometric transformations to a deterministic rasterization engine, thereby reframing the problem from a challenging geometric regression task to a consistent image-to-image translation task.

**Projective Conditioning.** We augment the context views with their corresponding depth maps  $\{\mathcal{D}_i^c\}$ , which can be readily obtained from off-the-shelf models [13, 31]. We then warp the context images into the target view by first unprojecting their pixels into a unified 3D point cloud and then rasterizing this point cloud from the target view:

$$\mathcal{I}^{c \rightarrow t} = \text{Rast}(\{\text{UnProj}(\mathcal{D}_i^c, \mathcal{I}_i^c, \mathcal{C}_i^c)\}, \mathcal{C}^t), \quad (3)$$

where  $\text{UnProj}$  is the standard un-projection operator, and  $\text{Rast}$  is a point cloud rasterizer<sup>1</sup>.

This reframes novel view synthesis as an image-to-image task. The resulting point cloud projection image  $\mathcal{I}^{c \rightarrow t}$  (Fig. 2 Top-right) provides a direct and coherent visual cue about the *visible* scene geometry from the target viewpoint, while leaving disoccluded regions as holes that must be completed

<sup>1</sup>We use the `gsplat` [41] rasterizer due to its excellent compatibility with PyTorch tensors and batchified rendering APIs. Each 3D point is rendered as a 3D Gaussian splat [2] with preset, fixed parameters.

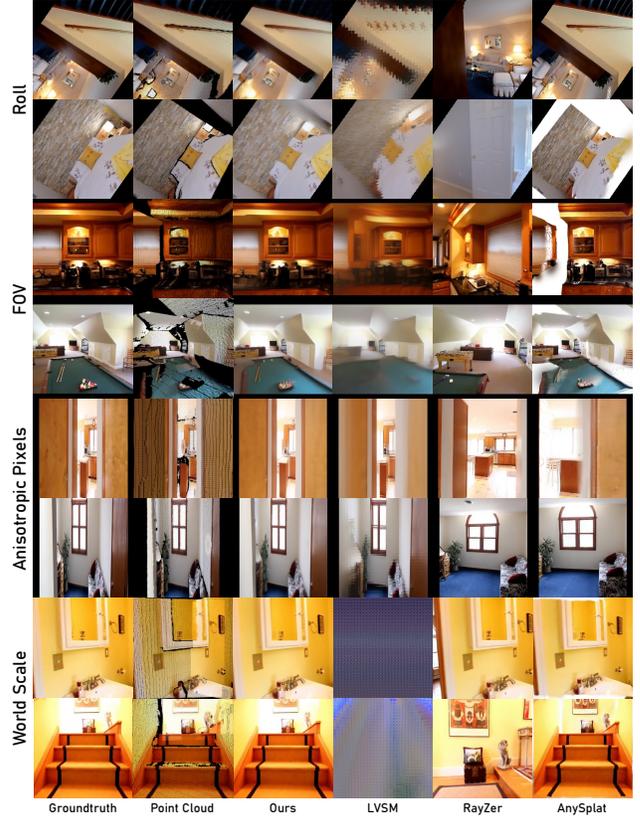


Figure 4. **Qualitative Results on our Consistency Benchmark.** Our method produces more geometrically consistent results. LVSM struggles to maintain geometric consistency, while RayZer and AnySplat fail to retrieve accurate camera parameters.

by the network. Consequently, small changes in the target camera pose  $\mathcal{C}^t$  lead to smooth and localized changes in the input projection image. This grants the model inherent robustness to various camera transformations (e.g., changes in focal length, aspect ratio, or extrapolated extrinsics).

**Architecture.** Following previous works [11], our model employs a decoder-only ViT as its backbone. The input token sequence is constructed from three sources: 1) patch-wise tokens from the context images  $\mathcal{I}_i^c$ , 2) patch-wise tokens from the generated point cloud projection image  $\mathcal{I}^{c \rightarrow t}$ , and 3) features from the pretrained DINOv3 model [25]  $\mathbf{f}^{\text{dino}}$  of the context views, which we find empirically enhances structural consistency (Sec. 5.3).

Formally, we tokenize each context and target view patches separately with linear embedding layers:

$$\mathbf{x}_{ij}^c = \text{Linear}_c(\mathcal{I}_{ij}^c) \quad \mathbf{x}_j^t = \text{Linear}_p(\mathcal{I}_j^{c \rightarrow t}), \quad (4)$$

where  $\mathcal{I}_j$  denotes the  $j$ -th patch of  $\mathcal{I}$ . All tokens are then concatenated sequentially into  $\{\mathbf{z}_i^0\} = [\{\mathbf{x}_{ij}^c\}, \{\mathbf{f}_{ij}^{\text{dino}}\}, \{\mathbf{x}_j^t\}]$  and processed through:

$$\{\mathbf{z}_i^l\} = \text{TransformerLayer}^l(\{\mathbf{z}_i^{l-1}\}). \quad (5)$$

The output tokens from the last layer that correspond to the target view are decoded into RGB patches using a linear layer followed by a sigmoid activation, resulting in the final rendered image  $\hat{\mathcal{I}}_j^t = \sigma(\text{Linear}_o(\mathbf{z}_j^t))$ .

A unique challenge with projective conditioning is that the image  $\mathcal{I}^{c \rightarrow t}$  often contains empty regions. After patchification, these empty patches can lead to identical input tokens, creating ambiguity for the permutation-invariant self-attention mechanism. To resolve this issue, we apply Rotary Positional Embeddings (RoPE) [27] to all tokens, ensuring that each token is assigned unique positional information.

#### 4.1. Quotient-Space Interpretation

Here we provide a quotient-space interpretation of why projective conditioning guarantees the invariance as described in Sec. 3.2. Let  $q = \text{Rast} \circ \text{UnProj}$  denote the *projective conditioning operator*, which converts the full configuration space  $\mathcal{X}$  into a point cloud image from the target viewpoint:

$$q : \mathcal{X} = (\{\mathcal{I}_i^c, \mathcal{D}_i^c, \mathcal{C}_i^c\}_i \times \{\mathcal{C}^t\}) \rightarrow \text{Im}(q). \quad (6)$$

Given that for any 3D point  $\mathbf{X} \in \mathbb{P}^3$  and camera projection matrix  $\mathbf{P}$ , the projective image relationship holds when simultaneously apply the transformation  $T$  to both:

$$\mathbf{P}' \mathbf{X}' \sim (\mathbf{P}T^{-1})(T\mathbf{X}) = \mathbf{P}\mathbf{X}, \quad (7)$$

which means that the transformation  $T$  does not change the relationship between the 3D point and the camera. Therefore, the composition operator  $q$  produces outputs that depend only on the relative arrangement among cameras and geometry. Consequently,  $q(\mathcal{X})$  provides an invariant representation of the quotient space  $\mathcal{X}/\text{SE}(3)$ , meaning that all configurations that differ only by a global coordinate gauge are map to the same element in  $\text{Im}(q)$ .

In practice, the network learns a mapping  $h : \text{Im}(q) \rightarrow \mathcal{I}^t$ , and the overall function can be expressed as  $f = h \circ q$ . This formulation ensures *gauge-free conditioning by design*: the model never needs to infer invariance to global transformations from data, since such invariance is already encoded in the projective operator itself.

#### 4.2. Mask Auto-Encoding Pretraining

Acquiring large-scale, calibrated RGB-D datasets for training is often resource-intensive. To mitigate this dependency and leverage abundant uncalibrated image and video data, we propose a self-supervised pretraining strategy.

Our key observation is that the projected sparse point-cloud projection image that we use visually resembles a randomly masked target image (Fig. 5). This motivates a pretext task inspired by masked image modeling [8, 33]. During pretraining, we corrupt the ground-truth target image  $\mathcal{I}^t$  to obtain  $\mathcal{I}^{t*}$ : first by randomly removing a portion of patches, then sparsifying some of the remaining ones by

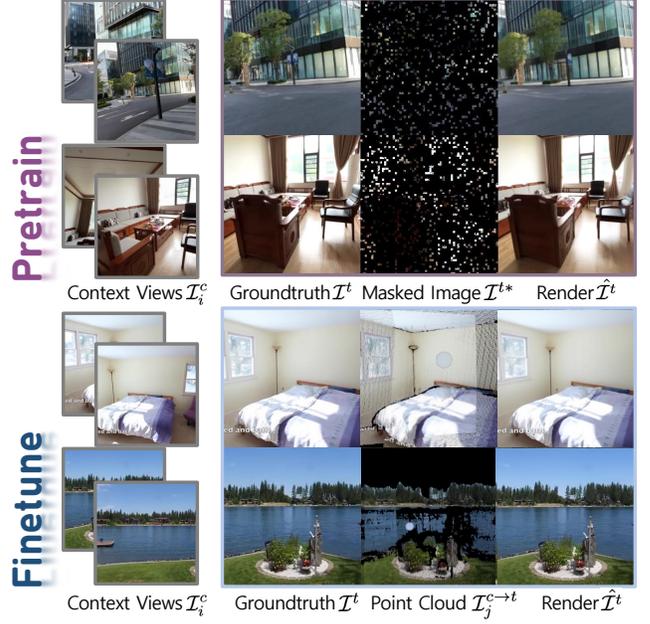


Figure 5. **Our training stages.** *Pretraining (top)*: The model reconstructs a target view from a randomly masked version of itself, conditioned on context views, using uncalibrated image data. *Fine-tuning (bottom)*: The model is then fine-tuned to reconstruct the target view from a point-cloud projection image obtained by warping the context views into the target camera frustum.

dropping pixels, and finally applying a random affine color transform to mimic exposure or camera-setting changes.

The corrupted image  $\mathcal{I}^{t*}$ , together with the context views  $\mathcal{I}_i^c$ , is fed into our rendering ViT, which is trained to reconstruct the original target image. This objective teaches the model robust cross-view image completion prior. After this self-supervised stage, we fine-tune the model on the projective conditioning task for a shorter schedule, yielding better performance and improved data efficiency.

**Optimization.** Following prior work [11, 45], we train our model with perceptual losses:

$$\mathcal{L} = \text{MSE}(\mathcal{I}^t, \hat{\mathcal{I}}^t) + \lambda \cdot \text{Perceptual}(\mathcal{I}^t, \hat{\mathcal{I}}^t), \quad (8)$$

where the perceptual loss [12] encourages the preservation of high-level features, and  $\lambda$  is a balancing hyperparameter.

## 5. Experiments

**Implementation Details.** We follow most architectural design choices outlined in LVSM [11]. Specifically, our model is a vision transformer with 12 or 24 multi-head self-attention layers. We use a patch size of  $p = 8$  and latent dimension  $d_{\text{model}} = 768$  with 64 dimensions per head. Our model is pretrained for 100,000 steps with an AdamW [19] optimizer. The learning rate is cosine scheduled with a peak value =  $10^{-3}$  and 3000 warm-up steps at the beginning. During the fine-tuning stage, the model continues from the

Model	Anisotropic Pixel			World Scale			FOV			Roll		
	SSIM $\uparrow$	LPIPS $\downarrow$	PSNR (M) $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$	PSNR (M) $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$	PSNR (M) $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$	PSNR (M) $\uparrow$
AnySplat [10]	0.634	0.376	10.48	0.616	0.346	18.06	0.808	0.163	15.37	0.536	0.446	14.14
WorldMirror [18]	0.578	0.380	11.71	0.742	0.250	22.07	0.855	0.122	18.30	0.552	0.379	16.50
LVSM [11]	0.725	0.235	19.58	0.318	0.633	14.56	0.813	0.119	18.67	0.588	0.454	19.54
RayZer* [9]	0.578	0.380	11.71	0.428	0.483	14.06	0.730	0.223	13.52	0.337	0.631	10.34
Ours	0.763	0.215	19.66	0.812	0.169	25.43	0.877	0.104	20.88	0.629	0.411	17.53
LVSM + aug.	0.721	0.252	20.09	0.344	0.666	13.57	0.870	0.104	21.26	0.588	0.442	19.79
Ours + aug.	<b>0.784</b>	<b>0.191</b>	<b>20.33</b>	<b>0.823</b>	<b>0.155</b>	<b>25.78</b>	<b>0.890</b>	<b>0.090</b>	<b>21.55</b>	<b>0.702</b>	<b>0.303</b>	<b>20.04</b>

Table 1. **Quantitative results on our consistency benchmark.** We evaluate model robustness against four types of camera transformations. Projective conditioning yields more consistent results than ray-based [9, 11] view synthesis models and feed-forward 3D Gaussian baselines [10, 18]. \*We use the 24-view checkpoint from RayZer [9]; see Sec. 5.1 for details. “+ aug.” denotes models fine-tuned with additional camera augmentations for 500 extra steps.

Model	Small Overlap			Medium Overlap			Large Overlap			Total		
	PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$	PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$	PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$	PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$
PixelNeRF [42]	19.27	0.536	0.568	20.38	0.559	0.540	20.94	0.581	0.517	20.26	0.560	0.540
PixelSplat [3]	21.22	0.752	0.225	23.61	0.821	0.162	26.18	0.879	0.115	23.76	0.821	0.164
MVSplat [4]	20.67	0.730	0.238	23.97	0.819	0.165	27.32	<b>0.889</b>	0.112	24.12	0.817	0.168
NoPoSplat [38]	21.58	0.750	0.231	23.67	0.808	0.177	25.84	0.854	0.133	23.78	0.807	0.178
AnySplat [10]	15.07	0.581	0.411	17.08	0.613	0.350	19.58	0.654	0.281	17.30	0.617	0.345
Hunyuan-WorldMirror [18]	19.03	0.677	0.313	21.40	0.741	0.250	23.95	0.796	0.196	21.55	0.741	0.250
LVSM (12 layers) [11]	21.58	0.721	0.251	24.49	0.796	0.180	27.38	0.858	0.127	24.60	0.795	0.182
RayZer* [9]	20.70	0.669	0.278	22.82	0.737	0.222	24.75	0.789	0.184	22.85	0.735	0.225
Ours (12 layers)	23.64	0.789	0.188	25.60	0.833	0.147	27.43	0.867	0.116	25.64	0.832	0.148
LVSM (24 layers) [11]	22.71	0.765	0.202	25.60	0.830	0.149	28.58	0.887	0.108	25.74	0.830	0.150
Ours (24 layers)	<b>24.98</b>	<b>0.807</b>	<b>0.171</b>	<b>26.88</b>	<b>0.852</b>	<b>0.132</b>	<b>28.60</b>	<b>0.889</b>	<b>0.101</b>	<b>26.90</b>	<b>0.851</b>	<b>0.133</b>

Table 2. **Quantitative evaluation results on the RealEstate10K [48] dataset.** We follow the benchmark splits from NoPoSplat [38]. \*We use the 24 view checkpoint from RayZer [9], see Sec. 5.1 for details.

pretrained state and utilizes a new warm-up cosine learning rate schedule, with a peak value of  $4 \times 10^{-4}$ .

**Compute.** Our 12-layer model is pretrained and fine-tuned on 4 NVIDIA 3090 GPUs with a batch size of 6 per GPU for approximately 120 hours. In contrast, the 24-layer model is trained on NVIDIA H100 GPUs for around 1560 GPU-hours, which is  $\sim 7$  times lower than the training time of our baseline model [11].

## 5.1. Consistency Benchmark

**Benchmark Construction.** To rigorously evaluate the 3D consistency and robustness of view synthesis models under shifts in camera parameter distributions, we construct a new benchmark based on the NoPoSplat [38] benchmark. For each sequence, we apply four types of out-of-distribution transformations to the target cameras while keeping the context views unchanged, simulating real-world usage:

- **Anisotropic Pixel:** Modify intrinsics to change the pixel aspect ratio within  $[0.1, 10]$ .
- **World Scale:** Scale the world coordinate system by a constant factor, adjusting camera positions and depth accordingly while keeping ground-truth images unchanged.
- **FOV:** Simulate zooming by resizing the target image and updating the focal length in the intrinsics.
- **Roll:** Apply an in-plane roll rotation to the target camera. Because these transformations may introduce empty regions in the target frame, PSNR is computed only over valid pixels (“PSNR (M)”), while SSIM and LPIPS [46] are evaluated on the full image.

**Comparisons.** As shown in Tab. 1, our method consistently outperforms all baselines across all transformations. The largest gain appears in the *World Scale* test, where our model reaches 25.43 PSNR versus LVSM’s 14.56, underscoring the brittleness of Plücker ray conditioning to scale changes. Our method also excels in the *Anisotropic Pixel* and *FOV* settings, achieving higher SSIM and lower LPIPS, reflecting better structural and perceptual fidelity. With an additional 500 steps of camera augmentations (“+ aug.”), our model adapts quickly, whereas “LVSM + aug.” shows only limited improvement in short time since these simple transformations can cause significant distribution shift in the Plücker ray space.

Qualitative comparisons (Fig. 4) reinforce these findings. LVSM produces pronounced jagged or grid-like artifacts under roll, FOV, and aspect-ratio variations, and collapses entirely under large world-scale changes. Gaussian-based methods such as AnySplat often generate distortions and holes, likely from inaccurate intrinsics or geometry estimation, while RayZer frequently fails to render the specified viewpoint due to limitations in its Camera Estimator. In contrast, our approach remains stable and geometrically coherent across all transformations. We attribute the remaining gap to the fact that feed-forward Gaussians expose geometry errors directly through a fixed rasterizer, whereas our model learns to correct its imperfections during 2D decoding.

**Evaluating RayZer.** RayZer [9] only open-sourced the checkpoint trained for the setting with 16 context views and 8 target views (realEstate52K.pt). Directly using the pro-



Figure 6. **Qualitative comparisons on the RealEstate10K [48] dataset.** Our model consistently generates more plausible and geometrically consistent results. We highlight two common failure modes where **Orange boxes** indicate rendering artifacts like blurriness and ghosting, while **blue boxes** suggest geometric errors where models fail to preserve scene structure or render from the correct viewpoint.

vided checkpoint in our evaluation setting leads to degraded results because their model hardcodes the view index into the positional embedding. To ensure a fair comparison, we therefore appended repeated views to the token list with fabricated view indices.

	<i>Seen</i> PSNR	<i>Unseen</i> PSNR	Full
AnySplat [10]	16.22	11.86	15.07
Hunyuan-WorldMirror [18]	19.65	16.91	19.03
LVSM [11]	22.14	19.39	21.58
Ours	<b>24.30</b>	<b>21.29</b>	<b>23.64</b>
Point Cloud Image (reference)	15.81	N/A	12.54

Table 3. **Seen vs. unseen analysis on RealEstate10K-Small.** We split the target view into *seen* and *unseen* regions based on the projected point cloud image as a visibility mask. Our method captures the view dependent effects in *seen* regions and hallucinates more plausible content in *unseen* regions.

## 5.2. Sparse View Novel View Synthesis

We compare the quality of sparse view novel view synthesis against several baselines, including: 1) NeRF-based method: pixelNeRF [42]; 2) Feed-forward 3D Gaussians: PixelSplat [3], MVSplat [4], NoPoSplat [38], AnySplat [10], and Hunyuan-WorldMirror [18]; 3) View synthesis models: LVSM [11] and RayZer [9]. As shown in Tab. 2, our model consistently surpasses all the baseline methods. Against the strongest 12-layer LVSM, our 12-layer variant delivers small improvements at large overlaps and increasingly larger

gains as overlap decreases. Scaling to 24 layers further boosts performance. Compared to the full 24-layer LVSM, our model matches the performances at large overlaps and clearly outperforms it at medium and small overlap setting, indicating stronger robustness to viewpoint variation and (dis)occlusions.

Qualitatively, as illustrated in Fig. 6, our method produces more plausible and geometrically consistent results. Gaussian-based methods, such as AnySplat and WorldMirror, exhibit noticeable rendering artifacts, including blurriness and ghosting, particularly when synthesizing views with large viewpoint changes. Other view synthesis models, like RayZer and LVSM, struggle with geometric consistency and view controllability, often resulting in distorted scenes or failing to render the correct structures.

**Seen vs. Unseen Breakdown.** Tab. 3 refines the Small-overlap split by separating *seen* and *unseen* target pixels. We classify a target pixel as *seen* if it is covered by the projected point cloud image; the remaining pixels are classified as *unseen*. Our method achieves 24.30dB PSNR on *seen* regions and 21.29dB PSNR on *unseen* regions, exceeding LVSM by +2.16dB and +1.90dB PSNR, respectively. This analysis demonstrates that our approach generalizes beyond mere warping, maintaining the largest margin on *unseen* pixels where synthesis must hallucinate newly revealed content. For reference, directly rendering the projected point-cloud image

#	Model	Pretrained on	for (steps)	Finetune on	for (steps)	PSNR (Large)	PSNR (Medium)	PSNR (Small)	Total
1	LVSM	None		RealEstate10K	100K	27.38	24.49	21.58	24.60
2	Ours	None		RealEstate10K	100K	26.96	25.11	23.06	25.13
3	Ours	RealEstate10K	75K	RealEstate10K	25K	26.28	24.78	23.02	24.78
4	Ours	DL3DV	100K	RealEstate10K	200	23.36	22.37	21.03	22.32
5	Ours	DL3DV	100K	RealEstate10K	<b>50K</b>	<b>27.43</b>	<b>25.60</b>	<b>23.64</b>	<b>25.64</b>

Table 4. **Ablation studies** on pretraining. Rows 1–3 are compute-matched at 100K total steps, while row 5 uses a longer schedule. Pretraining on DL3DV [15] improves initialization and data efficiency on NoPoSplat [38].

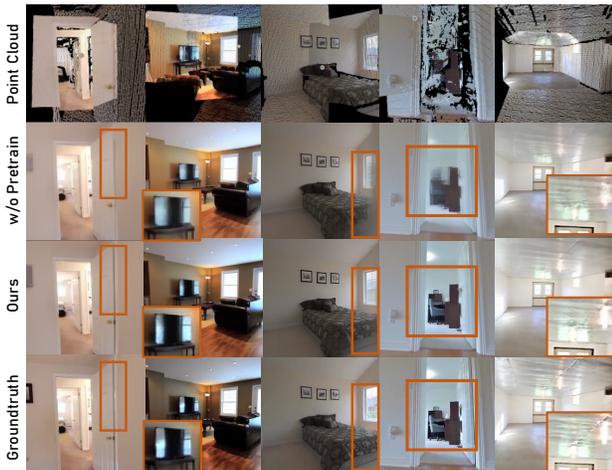


Figure 7. **Qualitative ablation on pretraining.** Rendering artifacts are highlighted or zoomed in with orange boxes.

is significantly inferior, underscoring that the improvements stem from learned synthesis rather than simply copying geometry.

Time (ms)	WorldMirror	LVSM	RayZer*	Ours-12	Ours-24
Processing	74	2.8	18	1.1	2.5
Rendering	26	52	56	28	56

Table 5. **Runtime Analysis.** Measured on NVIDIA 3090 GPUs.

**Runtime Analysis.** We compare the runtime of our model against several baselines in Tab. 5. The analysis is divided into two phases: ‘Processing Time’, which measures the duration required to construct the 3D scene representation from input views, and ‘Rendering Time’, which measures the time taken to render a single novel view. Our 12-layer model runs in real-time with a significantly lower processing time than the 3D Gaussian-based baseline [18] while our full 24-layer model matches rendering speed of our baseline view synthesis models [9, 11].

### 5.3. Ablation Studies

We conduct a series of ablation studies to validate the effectiveness of our key design choices.

**Pretraining.** We investigate the impact of pretraining in Tab. 4. We first establish a baseline by training our model from scratch (row 2). Even without pretraining, our model

Pretrain	DINO Prior	Projection	PSNR	SSIM	LPIPS
×	×	×	24.60	0.795	0.182
×	×	✓	25.20	0.811	0.177
×	✓	✓	25.13	0.816	0.163
✓	✓	✓	<b>25.64</b>	<b>0.832</b>	<b>0.148</b>

Table 6. **Ablation studies** on pretraining, DINO priors, and projective conditioning on DL3DV [15] and RealEstate10K [48].

achieves a PSNR of 25.13, already surpassing the baseline method in row 1. To further enhance performance, we explore pretraining with various datasets and different numbers of finetuning steps. While pretraining and finetuning on the same dataset serves as a useful reference, the true breakthrough comes from leveraging a larger, more diverse dataset. By pretraining on the large-scale DL3DV dataset, we provide our model with a powerful and generalizable prior for 3D scene understanding. With just 200 steps of finetuning (row 4), the model rapidly develops foundational rendering capabilities, achieving a PSNR of 22.32. Building upon this strong foundation, a more extensive finetuning schedule of 50K steps (row 5) enables the model to fully adapt its robust prior to the specifics of the target domain.

**Other Design Choices.** Tab. 6 ablates the contributions of several design choices. For DINOv3 prior, we adopt the DINOv3-ViT-B model in our experiments. Starting with a model that lacks pretraining, DINO prior, and projective conditioning (which corresponds to our baseline [11], row 1), the addition of projection (row 2) yields consistent improvements (25.20 / 0.811 / 0.177). Incorporating DINO feature priors on top of projection (row 3) further enhances perceptual and structural quality, with only a minor change in PSNR. The combination of pretraining, DINO, and projection (row 4) achieves the best overall results.

## 6. Conclusion

We show that Plücker ray conditioning is brittle under coordinate-gauge changes and simple camera transformations, whereas target-view projective conditioning provides a stronger interface for deterministic feed-forward novel view synthesis. Combined with the proposed masked auto-encoding pretraining strategy, it achieves state-of-the-art performance on standard benchmarks and our out-of-distribution consistency benchmark. Remaining failure modes stem from imperfect geometry and dynamic scenes.

## 7. Acknowledgment

We sincerely thank Haozhe Lou, Jiajun Jiang for the suggestions on theoretical analysis; Haoxuan Wang for the assistance and valuable feedbacks on figure designs; Handi Yin, Haonan He, Bonan Liu, Jianteng Chen, Runyi Yang, Liyi Luo, Xuanchao Peng, Yiming Zhu, Shibo Wang and Xinyuan Xu for fruitful discussions and proofreading.

## References

- [1] Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. Zip-NeRF: Anti-Aliased Grid-Based Neural Radiance Fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 19697–19705. IEEE, 2023. 2
- [2] Kerbl Bernhard, Kopanas Georgios, Leimkühler Thomas, and Drettakis George. 3D Gaussian Splatting for Real-Time Radiance Field Rendering. *ACM Transactions on Graphics*, 2023. 2, 4
- [3] David Charatan, Sizhe Lester Li, Andrea Tagliasacchi, and Vincent Sitzmann. pixelSplat: 3D Gaussian Splats from Image Pairs for Scalable Generalizable 3D Reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2024. 1, 2, 6, 7
- [4] Yuedong Chen, Haofei Xu, Chuanxia Zheng, Bohan Zhuang, Marc Pollefeys, Andreas Geiger, Tat-Jen Cham, and Jianfei Cai. MVsplat: Efficient 3D Gaussian Splatting from Sparse Multi-View Images. arXiv, 2403.14627, 2024. 2, 6, 7
- [5] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale, arXiv, 2010.11929, 2021. 1, 3
- [6] Phillip Griffiths and Joseph Harris. *Principles of algebraic geometry*. John Wiley & Sons, 2014. 3
- [7] Zekai Gu, Rui Yan, Jiahao Lu, Peng Li, Zhiyang Dou, Chenyang Si, Zhen Dong, Qifeng Liu, Cheng Lin, Ziwei Liu, Wenping Wang, and Yuan Liu. Diffusion as shader: 3d-aware video diffusion for versatile video generation control. *arXiv preprint arXiv:2501.03847*, 2025. 2
- [8] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked Autoencoders Are Scalable Vision Learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2021. 2, 5
- [9] Hanwen Jiang, Hao Tan, Peng Wang, Haiyan Jin, Yue Zhao, Sai Bi, Kai Zhang, Fujun Luan, Kalyan Sunkavalli, Qixing Huang, and Georgios Pavlakos. RayZer: A Self-supervised Large View Synthesis Model, arXiv, 2505.00702, 2025. 1, 2, 6, 7, 8
- [10] Lihan Jiang, Yucheng Mao, Linning Xu, Tao Lu, Kerui Ren, Yichen Jin, Xudong Xu, Mulin Yu, Jiangmiao Pang, Feng Zhao, Dahua Lin, and Bo Dai. AnySplat: Feed-forward 3D Gaussian Splatting from Unconstrained Views, arXiv, 2505.23716, 2025. 2, 6, 7
- [11] Haiyan Jin, Hanwen Jiang, Hao Tan, Kai Zhang, Sai Bi, Tianyuan Zhang, Fujun Luan, Noah Snavely, and Zexiang Xu. LVSM: A Large View Synthesis Model with Minimal 3D Inductive Bias. In *The Thirteenth International Conference on Learning Representations*, 2025. 1, 2, 3, 4, 5, 6, 7, 8
- [12] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision*, pages 694–711. Springer, 2016. 5
- [13] Nikhil Keetha, Norman Müller, Johannes Schönberger, Lorenzo Porzi, Yuchen Zhang, Tobias Fischer, Arno Knapitsch, Duncan Zauss, Ethan Weber, Nelson Antunes, Jonathon Luiten, Manuel Lopez-Antequera, Samuel Rota Bulò, Christian Richardt, Deva Ramanan, Sebastian Scherer, and Peter Kotschieder. MapAnything: Universal Feed-Forward Metric 3D Reconstruction, arXiv, 2509.13414, 2025. 2, 3, 4
- [14] Ruilong Li, Brent Yi, Junchen Liu, Hang Gao, Yi Ma, and Angjoo Kanazawa. Cameras as Relative Positional Encoding, arXiv, 2507.10496, 2025. 2
- [15] Lu Ling, Yichen Sheng, Zhi Tu, Wentian Zhao, Cheng Xin, Kun Wan, Lantao Yu, Qianyu Guo, Zixun Yu, Yawen Lu, et al. D13dv-10k: A large-scale scene dataset for deep learning-based 3d vision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22160–22169, 2024. 3, 8
- [16] Andrew Liu, Ameesh Makadia, Richard Tucker, Noah Snavely, Varun Jampani, and Angjoo Kanazawa. Infinite nature: Perpetual view generation of natural scenes from a single image. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 14438–14447, 2021. 2
- [17] Junchen Liu, Wenbo Hu, Zhuo Yang, Jianteng Chen, Guoliang Wang, Xiaoxue Chen, Yantong Cai, Huan-ang Gao, and Hao Zhao. Rip-nerf: Anti-aliasing radiance fields with ripmap-encoded platonic solids. In *SIGGRAPH'24 Conference Proceedings*, 2024. 2
- [18] Yifan Liu, Zhiyuan Min, Zhenwei Wang, Junta Wu, Tengfei Wang, Yixuan Yuan, Yawei Luo, and Chunchao Guo. World-Mirror: Universal 3D World Reconstruction with Any-Prior Prompting, arXiv, 2510.10726, 2025. 2, 6, 7, 8
- [19] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*, 2019. 5
- [20] Haozhe Lou, Yurong Liu, Yike Pan, Yiran Geng, Jianteng Chen, Wenlong Ma, Chenglong Li, Lin Wang, Hengzhen Feng, Lu Shi, Liyi Luo, and Yongliang Shi. Robo-gs: A physics consistent spatial-temporal model for robotic arm with hybrid representation. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 15379–15386, 2025. 2
- [21] Ricardo Martin-Brualla, Rohit Pandey, Shuoran Yang, Pavel Pidlypenskiy, Jonathan Taylor, Julien P. C. Valentin, Sameh Khamis, Philip Davidson, Anastasia Tkach, Peter Lincoln, Adarsh Kowdle, Christoph Rhemann, Dan B. Goldman, Cem Keskin, Steven M. Seitz, Shahram Izadi, and Sean Ryan Fanello. Lookingood: Enhancing performance capture with

- real-time neural re-rendering. *ACM Trans. Graph.*, 37(6):255, 2018. 2
- [22] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. In *Computer Vision – ECCV 2020*, pages 405–421, Cham, 2020. Springer International Publishing. 2
- [23] Julius Plucker. XVII. On a new geometry of space. *Philosophical Transactions of the Royal Society of London*, 155: 725–791, 1997. 1
- [24] Xuanchi Ren, Tianchang Shen, Jiahui Huang, Huan Ling, Yifan Lu, Merlin Nimier-David, Thomas Müller, Alexander Keller, Sanja Fidler, and Jun Gao. Gen3c: 3d-informed world-consistent video generation with precise camera control. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025. 2
- [25] Oriane Siméoni, Huy V. Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, Francisco Massa, Daniel Haziza, Luca Wehrstedt, Jianyuan Wang, Timothée Darcet, Théo Moutakanni, Leonel Sentana, Claire Roberts, Andrea Vedaldi, Jamie Tolan, John Brandt, Camille Couprie, Julien Mairal, Hervé Jégou, Patrick Labatut, and Piotr Bojanowski. Dinov3, arXiv, 2508.10104, 2025. 2, 4
- [26] Brandon Smart, Chuanxia Zheng, Iro Laina, and Victor Adrian Prisacariu. Splatt3R: Zero-shot Gaussian Splatting from Uncalibrated Image Pairs, arXiv, 2408.13912, 2024. 1
- [27] Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. RoFormer: Enhanced Transformer with Rotary Position Embedding, arXiv, 2104.09864, 2023. 5
- [28] Beiwen Tian, Liyi Luo, Hao Zhao, and Guyue Zhou. VIBUS: Data-efficient 3D scene parsing with Viewpoint Bottleneck and Uncertainty-Spectrum modeling. *ISPRS Journal of Photogrammetry and Remote Sensing*, 194:302–318, 2022. 2
- [29] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention Is All You Need, arXiv, 1706.03762, 2017. 3
- [30] Haoru Wang, Kai Ye, Yangyan Li, Wenzheng Chen, and Baoquan Chen. The Less You Depend, The More You Learn: Synthesizing Novel Views from Sparse, Unposed Images without Any 3D Knowledge, arXiv, 2506.09885, 2025. 1, 2
- [31] Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. VGGT: Visual Geometry Grounded Transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025. 2, 4
- [32] 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 (CVPR)*. IEEE, 2023. 2
- [33] Philippe Weinzaepfel, Vincent Leroy, Thomas Lucas, Romain Brégier, Yohann Cabon, Vaibhav Arora, Leonid Antsfeld, Boris Chidlovskii, Gabriela Csurka, and Jérôme Revaud. CroCo: Self-Supervised Pre-training for 3D Vision Tasks by Cross-View Completion, arXiv, 2210.10716, 2023. 2, 5
- [34] Zirui Wu, Jianteng Chen, Laijian Li, Shaoteng Wu, Zhikai Zhu, Kang Xu, Martin R. Oswald, and Jie Song. 3D Gaussian Inverse Rendering with Approximated Global Illumination, arXiv, 2504.01358, 2025. 2
- [35] Yunzhi Yan, Zhen Xu, Haotong Lin, Haian Jin, Haoyu Guo, Yida Wang, Kun Zhan, Xianpeng Lang, Hujun Bao, Xiaowei Zhou, and Sida Peng. Streetcrafter: Street view synthesis with controllable video diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. 2
- [36] Runyi Yang, Zhenxin Zhu, Zhou Jiang, Baijun Ye, Xiaoxue Chen, Yifei Zhang, Yuantao Chen, Jian Zhao, and Hao Zhao. Spectrally pruned gaussian fields with neural compensation, arXiv, 2405.00676, 2024. 2
- [37] Baijun Ye, Caiyun Liu, Xiaoyu Ye, Yuantao Chen, Yuhai Wang, Zike Yan, Yongliang Shi, Hao Zhao, and Guyue Zhou. Blending distributed nerfs with tri-stage robust pose optimization. *arXiv preprint arXiv:2405.02880*, 2024. 2
- [38] Botao Ye, Sifei Liu, Haofei Xu, Xueting Li, Marc Pollefeys, Ming-Hsuan Yang, and Songyou Peng. No Pose, No Problem: Surprisingly Simple 3D Gaussian Splats from Sparse Unposed Images, arXiv, 2410.24207, 2024. 2, 6, 7, 8
- [39] Baijun Ye, Minghui Qin, Saining Zhang, Moonjun Gong, Shaoting Zhu, Zebang Shen, Luan Zhang, Lu Zhang, Hao Zhao, and Hang Zhao. Gs-occ3d: Scaling vision-only occupancy reconstruction with gaussian splatting. *arXiv preprint arXiv:2507.19451*, 2025. 2
- [40] Chongjie Ye, Yinyu Nie, Jiahao Chang, Yuantao Chen, Yihao Zhi, and Xiaoguang Han. GauStudio: A Modular Framework for 3D Gaussian Splatting and Beyond. arXiv, 2403.19632, 2024. 2
- [41] Vickie Ye, Ruilong Li, Justin Kerr, Matias Turkulainen, Brent Yi, Zhuoyang Pan, Otto Seiskari, Jianbo Ye, Jeffrey Hu, Matthew Tancik, and Angjoo Kanazawa. Gsplat: An Open-Source Library for Gaussian Splatting. arXiv, 2409.06765, 2024. 4
- [42] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelNeRF: Neural Radiance Fields from One or Few Images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 1, 2, 6, 7
- [43] Mark YU, Wenbo Hu, Jinbo Xing, and Ying Shan. TrajectoryCrafter: Redirecting Camera Trajectory for Monocular Videos via Diffusion Models, arXiv, 2503.05638, 2025. 2
- [44] Wangbo Yu, Jinbo Xing, Li Yuan, Wenbo Hu, Xiaoyu Li, Zhipeng Huang, Xiangjun Gao, Tien-Tsin Wong, Ying Shan, and Yonghong Tian. ViewCrafter: Taming Video Diffusion Models for High-fidelity Novel View Synthesis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–18, 2025. 2
- [45] Kai Zhang, Sai Bi, Hao Tan, Yuanbo Xiangli, Nanxuan Zhao, Kalyan Sunkavalli, and Zexiang Xu. GS-LRM: Large Reconstruction Model for 3D Gaussian Splatting. arXiv, 2404.19702, 2024. 5
- [46] 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/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 586–595. IEEE, 2018. 6
- [47] Yuhang Zheng, Xiangyu Chen, Yupeng Zheng, Songen Gu,

Runyi Yang, Bu Jin, Pengfei Li, Chengliang Zhong, Zeng-mao Wang, Lina Liu, Chao Yang, Dawei Wang, Zhen Chen, Xiaoxiao Long, and Meiqing Wang. Gaussiangrasper: 3d language gaussian splatting for open-vocabulary robotic grasping. *IEEE Robotics and Automation Letters*, 9(9):7827–7834, 2024. [2](#)

- [48] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: Learning view synthesis using multiplane images. *ACM Trans. Graph. (Proc. SIGGRAPH)*, 37, 2018. [2](#), [6](#), [7](#), [8](#)