RETHINKING ASYMMETRY IN SELF-SUPERVISED 3D GEOMETRY LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Self-supervised learning relies on asymmetry between what a model observes and what it must explain. Recent methods learn 3D geometry from unposed images by predicting cameras and scene structure, then rendering held-out views. To avoid trivial solutions, they separate camera and scene branches or restrict interaction between context and held-out views, sacrificing cross-view correspondence. We revisit this trade-off with geometric accuracy as the objective. When a held-out view lies between context views, the context already constrains the geometry it observes. We show that excluding such views only from the 3D representation prevents trivial solutions while allowing a shared backbone to use all views for correspondence, improving depth and pose estimation over restricted designs. We further exploit view density: sparse and dense observations depict the same scene but provide different amounts of correspondence. Aligning sparse-context features with dense-context features supplies a geometric training signal alongside rendering, without a pretrained geometric teacher. We additionally align features with a frozen vision foundation model, which brings further gains. None of these signals require geometric annotation, and the resulting model improves geometry estimation over prior self-supervised methods. We analyze each component in controlled experiments validating the effectiveness of our proposed methods.

1 INTRODUCTION

Self-supervised learning has been a main driver of progress in representation learning. Because it requires no manual labels, models can be pretrained on as much data as we can collect, and their performance continues to improve as data, model size, and compute scale (Kaplan et al., 2020). This approach has succeeded in language (Radford et al., 2018; Devlin et al., 2019), vision (Caron et al., 2021; He et al., 2022), and embodied control (Mu & Rhoda AI Team, 2026). By contrast, many 3D foundation models learn to reproduce geometric annotations first recovered by an optimizer such as COLMAP (Schönberger & Frahm, 2016), a pipeline that is inefficient and hard to scale. Recent work (Jiang et al., 2025b; Zhao et al., 2026; Huang et al., 2026b) removes this dependence by making novel-view synthesis (NVS) the training objective, predicting cameras and scene structure from raw images or a video and rendering held-out views using the model's own predictions. These methods differ in scene representation and in how they share information across views, but evaluations have focused largely on synthesis quality. In self-supervised learning, however, the pretext task is only a means, and a method is judged by what it learns. We therefore treat view synthesis as the pretext task and geometry as the outcome, and ask which design choices make geometry learning better and more scalable.

We begin with the scene representation, which determines how closely the rendering objective is tied to geometry. Latent NVS methods (Jiang et al., 2025b; Mitchel et al., 2026; Prestel et al., 2026) render held-out views from an implicit scene code, so geometry is never made explicit. They achieve strong rendering quality, but can do so without recovering accurate 3D structure, so the synthesis loss only weakly constrains geometry and camera motion. We instead use an explicit representation, 3D Gaussians (Kerbl et al., 2023) placed in physical space, so that the rendered views are closely tied to the geometry of the scene.

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

Where should the asymmetry live? A self-supervised objective needs an asymmetry between what the model sees and what it explains, or trivial solutions satisfy it. In masked image modeling, what is hidden is exactly what must be predicted, so masking the input directly defines the task. In contrast, masking a view in multi-view 3D does not create the same asymmetry. The remaining views still represent the same scene, so hiding a view does not hide the geometry. It only removes the correspondence that could have helped recover it more accurately. Existing methods place the asymmetry early. E-RayZer (Zhao et al., 2026) splits camera and geometry prediction into two backbones, and NAS3R (Huang et al., 2026b) masks view interaction between the context and held-out sets, so context views attend only among themselves. Both methods block information leakage[1], but at the cost of the rich cross-view correspondence that geometry benefits from.

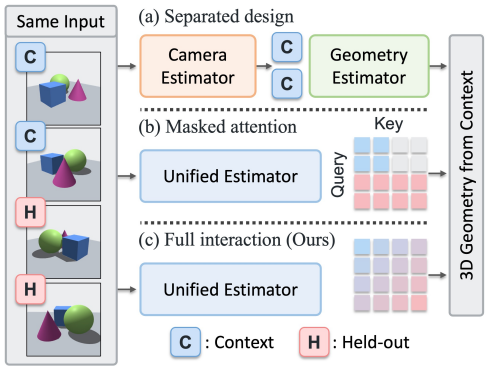


Figure 1: All designs build the 3D scene from context views (C). (a) Separate backbones and (b) masked attention restrict interaction between context and held-out views (H). (c) Our design allows full cross-view attention while excluding held-out views from the 3D representation.

We instead place the asymmetry in the 3D representation space. A shared backbone allows all views to exchange information when predicting depth and pose, preserving the correspondences needed for accurate geometry. We preserve the context and held-out split used in prior work and apply it when constructing the 3D scene, so that a held-out view is rendered only from context-view geometry. Thus, every view informs correspondence estimation, but no view reconstructs itself.

Asymmetric feature alignment. View density provides a second source of asymmetry. The same model predicts more accurate depth and pose when it sees more views of a scene, so a model given fewer views is a weaker version of itself. Such gaps within a single model have been used to bootstrap representations without any external teacher (Jiang et al., 2025a; Chefer et al., 2026). Self-Evo (Huang et al., 2026a) applies this idea to 3D by distilling a dense-context teacher into a sparse-context student, starting from a supervised model (Wang et al., 2025b; 2026b). We instead incorporate dense-to-sparse alignment directly into self-supervised training. We align features rather than predicted geometry to reduce the effect of errors in the dense-context outputs during training, and optimize alignment jointly with the rendering objective without a pretrained geometric teacher. The largest gains occur in single-view depth estimation, which the rendering objective does not supervise directly.

Semantics as a geometric prior. Both signals above are grounded in rendered pixels. The rendering loss supervises the model directly, and context alignment does so indirectly through features learned by rendering. Pixel comparison, however, says little about which pixels belong together. Such high-level structure often helps constrain geometry, especially in textureless or repetitive regions. Vision foundation models capture this kind of structure, and aligning with their features has proven effective in generative modeling (Yu et al., 2024; Wu et al., 2025) and supervised perception tasks (Yang et al., 2024; Knaebel et al., 2026). We therefore align our backbone features with those of a frozen vision foundation model (Siméoni et al., 2025; Yang et al., 2026), obtaining consistent gains in geometry estimation at modest additional cost.

We consider all these factors in a self-supervised framework for learning geometric representations and evaluate each through controlled experiments. Our contributions are:

- **Asymmetry in the 3D representation.** We show that excluding held-out views from the reconstructed scene prevents trivial solutions while a shared backbone retains cross-view correspondences for depth and pose estimation.

[1]This prevents held-out views from leaking into the reconstruction, which matters when rendering quality is the measure of interest. Geometry could also become biased toward the held-out views, but we find that richer view context in fact yields better geometric estimation (Sec. 3.2).

- **Self-alignment across view contexts.** We align sparse- and dense-context features jointly with the rendering objective, improving geometry without a pretrained geometric teacher. The largest gains occur in single-view depth estimation, which rendering does not supervise directly.
- **Semantic guidance for geometry.** Aligning intermediate features with a frozen vision foundation model adds structural guidance beyond pixel agreement, improving depth and pose estimation without geometric annotations.

2 Related Work

Self-supervised learning with novel-view synthesis. Feed-forward 3D models recover geometry directly from a set of images, but they are trained against camera and depth targets that an offline pipeline must produce first (Wang et al., 2024; 2025b; 2026a;b; Lin et al., 2025). A separate line of work has instead learned geometry from images alone. Early methods predicted depth and ego-motion from video and used warping between neighboring frames as supervision (Zhou et al., 2017; Godard et al., 2019; Bian et al., 2019; Lai et al., 2021), which ties learning to short baselines and pairwise constraints. RUST (Sajjadi et al., 2023) and RayZer (Jiang et al., 2025b) removed explicit geometry altogether, encoding a scene into a latent set (Sajjadi et al., 2022) and rendering new views with a learned decoder. Several methods extend this latent formulation to dynamic data and predictive (extrapolative) objectives (Chen et al., 2026; Mitchel et al., 2026; Prestel et al., 2026). Rendering in latent space minimizes inductive bias and lets the model learn freely from data, but it also leaves the recovered geometry only loosely tied to physical structure.

Another line of work keeps the geometry explicit, optimizing poses together with a NeRF (Mildenhall et al., 2021) or Gaussian scene (Kerbl et al., 2023) under photometric losses (Chen & Lee, 2023; Fu et al., 2023; Kang et al., 2025). Recently, E-RayZer (Zhao et al., 2026) and NAS3R (Huang et al., 2026b) scaled explicit self-supervised reconstruction to large transformers and data, yielding representations with stronger geometric grounding. To prevent shortcuts and information leakage, both restrict how information moves between the context and held-out views, E-RayZer by separating camera and geometry prediction into two backbones, NAS3R by masking view interaction in the decoder. The restriction keeps the objective from collapsing, but at the cost of rich correspondence. We instead investigate where this restriction should be placed, and find that the scene representation is a better location than the input.

Self-supervised representation alignment. Aligning intermediate features to an additional target has become a common way to improve representation learning beyond what the task loss provides. In generative modeling, REPA (Yu et al., 2024) and follow-up work (Wu et al., 2025; Shin et al., 2026) align diffusion transformer features (Peebles & Xie, 2023) with a frozen vision foundation model (Oquab et al., 2023), accelerating convergence and improving sample quality. Perception models use the same idea, aligning task features with such encoders (Yang et al., 2024; Knaebel et al., 2026) or distilling several of them into one (Ranzinger et al., 2024; Zhu et al., 2026).

On the other hand, recent work derives the target from the model itself, comparing its own representations across different inputs or layers (Jiang et al., 2025a; Chefer et al., 2026). Self-Evo (Huang et al., 2026a) brings this to 3D, distilling a dense-context teacher into a sparse-context student, though it starts from a supervised model and aligns the two branches through their predicted depth. We apply the same contrast without supervised initialization, aligning features jointly with the main rendering objective during training. We also align them with a frozen vision encoder, providing complementary internal and external signals without geometric annotations.

3 Approach

3.1 Preliminaries

Setup. Given N unposed images $\mathcal{V} = \{I_i\}_{i=1}^{N}$ of a static scene, we partition them into disjoint context and target sets, $\mathcal{C}$ and $\mathcal{T}$. A model f_θ predicts a scene representation $\mathcal{S}$ and camera poses $\hat{\pi}_i = (\hat{T}_i, \hat{K}_i)$, where $\hat{T}_i \in SE(3)$ is the extrinsics and $\hat{K}_i$ the intrinsics. A renderer g, either a learned decoder or a differentiable rasterizer, then synthesizes each target view:

$$(\mathcal{S}, \{\hat{\pi}_i\}) = f_\theta(\mathcal{V}), \quad \hat{I}_j = g(\mathcal{S}, \hat{\pi}_j), \quad \mathcal{L}_{\text{nvs}} = \sum_{j \in \mathcal{T}} \ell(\hat{I}_j, I_j). \tag{1}$$

Cameras are predicted relative to a reference view. Without metric supervision, the reconstruction is determined only up to a global scale. Estimating the pose of a target view requires observing its image, so target images enter the camera-prediction path. If a target image also contributes directly to $\mathcal{S}$, however, the model can render that view from its own representation and satisfy Eq. 1 without recovering a scene shared across views. Methods therefore impose asymmetry between the target image and the scene used to render it; they differ in where that asymmetry is applied.

Implicit vs. explicit representation. Implicit methods encode the context views into latent scene tokens Z, and g_ϕ is a learned decoder that queries Z with an encoding of the target camera, $\hat{I}_j = g_\phi(Z, \hat{\pi}_j)$. Since Z is not explicitly tied to 3D space, a low photometric loss need not reflect a consistent 3D scene, and the decoder can partly rely on interpolating appearance between nearby context views. Explicit methods instead represent the scene using explicit 3D representations, such as 3D Gaussians (3DGS). For each pixel u of view i, an output head predicts a ray depth $d_{i,u}$ along with 3DGS attributes. The Gaussian center is obtained by lifting the depth through the predicted camera,

$$\mu_{i,u} = \hat{T}_i \, d_{i,u} \, \frac{\hat{K}_i^{-1} u}{\|\hat{K}_i^{-1} u\|}, \qquad \mathcal{S} = \bigcup_{i \in \mathcal{C}} \mathcal{G}_i, \qquad \mathcal{G}_i = \bigcup_u \mathcal{G}_{i,u}, \tag{2}$$

where u is in homogeneous coordinates and $\mathcal{G}_i$ is a set of Gaussians predicted from view i, and $\mathcal{S}$ then contains Gaussians from every pixel of every context view. Rendering this scene with a differentiable rasterizer g places both depth and pose on the rendering path. A target view is then well explained only when predictions from different views agree in a shared 3D space, so photometric error can become a proxy for multi-view geometric consistency.

Where asymmetry is placed. Existing methods prevent target-view information from entering the reconstructed scene by restricting feature flow between views. *Separate branches* (Zhao et al., 2026) predict cameras from all views but reconstruct geometry from context views alone. This prevents target-view leakage, but predicts camera and geometry from separate features, without the interaction that a shared representation provides. *Masked attention* (Huang et al., 2026b) uses a single backbone with a view-level mask $A \in \{0,1\}^{N \times N}$, setting $A_{ik} = 0$ for $i \in \mathcal{C}$ and $k \in \mathcal{T}$ so that context tokens never read from targets. This keeps a shared backbone, but blocks context views from attending to targets and loses the cross-view correspondence they could provide. Both break the trivial solution by restricting what the model sees at the input.

3.2 Asymmetry in the 3D Representation

We view this framework as a means of learning geometry, with view synthesis as the training signal rather than the final goal. Previous work hides target views so that the model cannot see the geometry it is asked to render. However, when a target view lies between context views, it often observes scene regions that are already covered by the context. As illustrated in Fig. 2, the full set of captured views is shown in blue, the context subset in orange, and the held-out targets in green. The context views already span the viewpoints surrounding the targets and largely constrain the same underlying geometry. The held-out views therefore contribute primarily additional cross-view correspondences rather than new scene content. This suggests that asymmetry need not restrict which views participate in feature computation. It only needs to prevent a target view from being rendered from its own prediction. We test this against existing placements, holding the representation and backbone fixed.

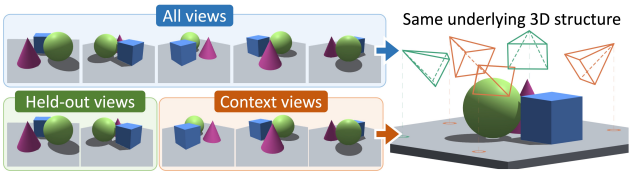


Figure 2: Held-out views add correspondence, not a new scene.

Controlled comparison. We compare three placements of asymmetry using a visual encoder (Yang et al., 2026) and a 12-layer transformer (Vaswani et al., 2017) with alternating view-wise and global attention (Wang et al., 2025b). The full-view and masked-attention variants predict cameras and Gaussians from this backbone. The separate-branch variant follows E-RayZer (Zhao et al., 2026): it predicts cameras from the shared backbone and adds a second 12-layer transformer to predict Gaussians, giving it greater capacity. We train all variants on DL3DV (Ling et al., 2024) for 100K

iterations with varying numbers of views and otherwise identical settings. At evaluation, we fix the frame span at 48 frames, so fewer views provide sparser coverage of the same scene. Full-view attention achieves the most accurate depth and pose (Fig. 3). It allows every view to contribute correspondences while excluding each target view's predicted Gaussians from the scene used to render it. Thus, the required asymmetry is imposed during 3D scene construction rather than during feature computation.

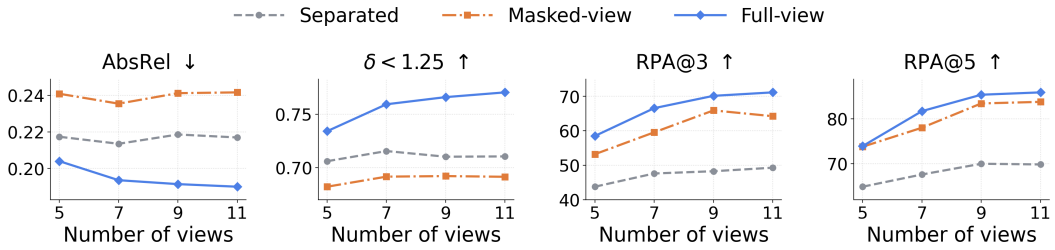


Figure 3: Effect of full interaction on depth and pose estimation. Depth is compared with MoGe-2 (Wang et al., 2025c) pseudo ground truth, and pose with dataset cameras.

3.3 Beyond Rendering Signals

Learning from context asymmetry. The preceding analysis shows that geometry estimates become less accurate as the view set becomes sparser. This gap is itself informative. Dense and sparse predictions come from the same network on the same scene, so the gap reflects only what the extra views contribute, and this asymmetry can serve as a learning signal. Self-Evo (Huang et al., 2026a) turns this gap into supervision for a pretrained 3D foundation model (Wang et al., 2025b; 2026b), using its own dense-context outputs as pseudo labels for a sparse-context copy updated by EMA.

We build on this idea, but with a crucial distinction: our model is trained entirely through self-supervision, without any 3D supervision to anchor its predictions. Without 3D supervision, depth predicted from different view sets carries no common scale, so the teacher's depth is not a reliable target. In our experiments, distilling it directly degrades geometry, even with a scale-invariant loss. Thus, we move the alignment from outputs to the features that produce them. These features encode the cross-view correspondence that dense context provides, and we match them with a cosine similarity loss, which compares their direction but not their magnitude. An EMA copy $f_{\bar{\theta}}$ processes the full set $\mathcal{V}$, and the online network f_θ processes a subset $\mathcal{V}_s \subset \mathcal{V}$. We align the features at the layers $\mathcal{K}$ that feed the output heads,

$$\mathcal{L}_{\text{align}} = -\sum_{k\in\mathcal{K}}\sum_{i\in\mathcal{V}_s} \cos\big(h_k(F_{s,i}^{(k)}),\, \text{sg}[F_i^{(k)}]\big), \qquad \cos(a,b) = \frac{a^\top b}{\|a\|\,\|b\|}, \tag{3}$$

where $F = f_{\bar{\theta}}(\mathcal{V})$ and $F_s = f_\theta(\mathcal{V}_s)$ are the dense and sparse features, h_k is a lightweight projection for layer k, $\text{sg}[\cdot]$ is the stop-gradient, and $\bar{\theta}$ is an exponential moving average of θ.

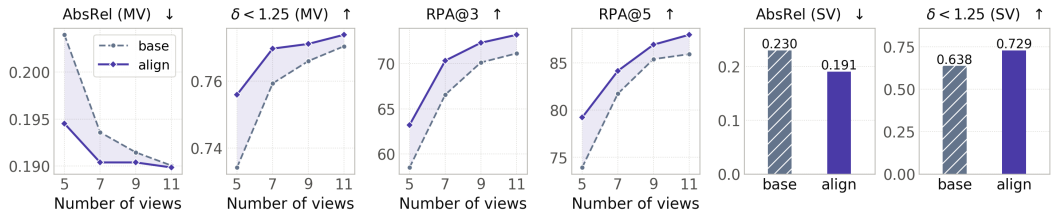


Figure 4: Effect of asymmetric feature alignment on DL3DV dataset. The left four plots evaluate multi-view depth and camera pose; the right two evaluate depth from a single input image.

We train the full-view model from Sec. 3.2 with and without context alignment, holding all other settings fixed. Alignment improves depth and pose across the evaluated view counts, with larger gains when fewer views are available (Fig. 4).

Semantic guidance. The rendering loss supervises appearance at the pixel level, so the constraints it provides are local. Textureless walls, repeated patterns, and thin structures produce photometric

error that is nearly flat with respect to depth, and the loss provides little information about object boundaries. Such distinctions are exactly what semantic features encode, we therefore align intermediate backbone features with those of a frozen vision foundation model (Siméoni et al., 2025; Yang et al., 2026). For each view, the encoder features F_i^{enc} are matched to the corresponding patch features of the frozen model Φ,

$$\mathcal{L}_{\text{sem}} = -\sum_{i \in \mathcal{V}} \cos\left(F_i^{\text{enc}}, \Phi(I_i)\right), \tag{4}$$

with a cosine similarity loss. The teacher is frozen and requires no annotation, and the alignment is applied at an intermediate layer so that the later layers remain free to specialize for geometry.

3.4 Training Objectives

Our model is trained with three signals, the rendering loss that defines the self-supervised task, the context alignment that transfers dense-view features to sparse inputs, and the semantic alignment that injects priors from a frozen foundation model. Putting everything together, we train with

$$\mathcal{L} = \mathcal{L}_{\text{nvs}} + \lambda_{\text{align}}\mathcal{L}_{\text{align}} + \lambda_{\text{sem}}\mathcal{L}_{\text{sem}}, \tag{5}$$

where $\mathcal{L}_{\text{nvs}}$ combines an L2 and a perceptual loss (Chen & Koltun, 2017), and the λ terms balance the auxiliary losses against it. We set $\lambda_{\text{align}} = 0.2$ and $\lambda_{\text{sem}} = 0.2$.

4 Experiments

Implementation details. Our model uses a pretrained Pixio (base) visual encoder (Yang et al., 2026) followed by a 12-layer transformer (Vaswani et al., 2017) with alternating view-wise and global attention and a feature dimension of 768. Images use 16×16 patches. A shared backbone predicts both cameras and Gaussians. The camera head regresses extrinsics relative to the middle input view and predicts one set of intrinsics shared across the scene. A DPT head (Ranftl et al., 2021; Wang et al., 2026a) predicts per-pixel Gaussian attributes, which are lifted to 3D with the predicted cameras. We construct the scene from context-view Gaussians only and render target views at their predicted poses using differentiable rasterization (Kerbl et al., 2023). Context alignment uses features from the four transformer layers read by the DPT head; semantic alignment uses the visual encoder output. We form sparse subsets by symmetrically dropping views on either side of the reference view. The dense branch is an EMA copy of the online network with momentum 0.999. All experiments use the same 12-layer base backbone.

Datasets. We train on RealEstate10K (RE10K) (Zhou et al., 2018) and DL3DV (Ling et al., 2024), resizing images to 256×256. ScanNetv2 (Dai et al., 2017) and NRGBD (Liu et al., 2019) serve as out-of-domain evaluations and are not used for self-supervised training.

Evaluation. We use publicly released baseline checkpoints and follow their evaluation protocols. On RE10K, we adopt the view indices of SPFSplat (Huang & Mikolajczyk, 2025), and on DL3DV, we follow E-RayZer (Zhao et al., 2026), sampling 10 views across a 96-frame span. For out-of-domain evaluation, we sample 10 views at a fixed stride, spanning 64 frames on ScanNetv2 and 256 on NRGBD. We report camera pose and depth, the two geometric quantities of interest. Pose is measured by relative pose accuracy (RPA) at several angular thresholds and AUC@30, and depth by AbsRel and δ_i after independent per-view median alignment. Since DL3DV provides no ground-truth depth, we use MoGe-2 (Wang et al., 2025c) predictions as pseudo ground truth in our analyses, and ground-truth depth on datasets that provide it. Multi-view depth is predicted from the full input set, while single-view depth is predicted from one image alone.

4.1 Results

We compare with SPFSplat and NAS3R (Huang et al., 2026b), which learn cameras and geometry from unposed images using self-supervision. All three methods train on RE10K with varying numbers of input views. SPFSplat additionally initializes its backbone from MASt3R (Leroy et al., 2024), which receives 3D supervision across multiple datasets, and takes ground-truth intrinsics as input. With 10 input views, our model achieves the highest pose accuracy at every reported threshold and competitive or better depth accuracy (Tab. 1); its advantage grows on out-of-domain data.

We further examine how each method scales with the number of input views (Fig. 5). Our model improves steadily as views are added, across every pose metric.

Method	3D prior	Params	RE10K (trained)		DL3DV		ScanNetv2				NRGBD			
			RPA@3↑	@5↑	RPA@3↑	@5↑	RPA@3↑	@5↑	AbsRel↓	$\delta < 1.25$ ↑	RPA@3↑	@5↑	AbsRel↓	$\delta < 1.25$ ↑
SPFSplat	MASt3R	616M	64.4	78.1	10.4	19.3	2.0	5.6	**0.222**	0.759	3.2	13.8	0.143	0.810
NAS3R	–	1.22B	47.5	67.2	1.6	3.5	0.1	0.5	0.371	0.607	4.2	7.4	0.214	0.630
Ours	–	183M	**65.4**	**80.0**	**14.0**	**26.4**	**2.7**	**7.1**	0.226	**0.786**	**15.8**	**30.1**	**0.105**	**0.880**

Table 1: Camera pose and depth accuracy with 10 input views, trained on RE10K dataset.

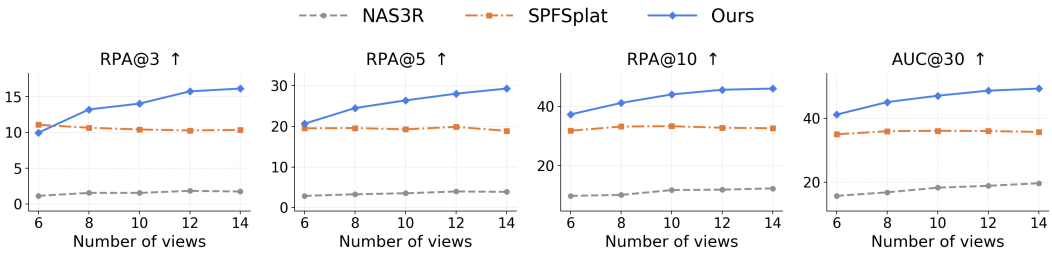


Figure 5: Camera pose accuracy on DL3DV dataset as the number of input views increases.

Against E-RayZer (Zhao et al., 2026), with both models trained on DL3DV, ours improves every reported pose and depth metric in and out of domain (Tab. 2). For depth, E-RayZer feeds only the context views to its geometry branch, whereas our backbone processes all images and only the context views are decoded by the DPT head, so both are scored on the same context views. Fig. 6 and Fig. 7 compare the recovered camera trajectories in and out of domain, respectively.

Method	Params	DL3DV (trained)		RE10K		ScanNetv2				NRGBD			
		RPA@3↑	@5↑	RPA@3↑	@5↑	RPA@3↑	@5↑	AbsRel↓	$\delta < 1.25$ ↑	RPA@3↑	@5↑	AbsRel↓	$\delta < 1.25$ ↑
E-RayZer	246M	60.4	72.2	54.5	74.2	4.9	12.3	0.167	0.820	35.8	61.7	0.095	0.907
Ours	183M	**70.2**	**79.8**	**57.1**	**75.0**	**6.3**	**16.3**	**0.126**	**0.871**	**61.0**	**80.5**	**0.082**	**0.916**

Table 2: Camera pose and depth accuracy with 10 input views, trained on DL3DV dataset.

Single-image depth. We further evaluate depth when each frame is given to the model alone (Tab. 3). The network receives only the reference camera token, so the pose head returns the identity pose and a focal length predicted from the image, which we use to convert ray depth to z-depth. This is the extreme of sparse context, where rendering provides no direct supervision and context alignment gives its largest gain. Our model outperforms E-RayZer by a large margin here. Interestingly, multi-view depth beats single-image depth on NRGBD, where our poses are accurate, but falls behind on ScanNetv2, where pose estimation is much harder. Since pose and multi-view depth come from the same cross-view reasoning, scenes where it fails degrade both, while single-image depth remains robust.

Input	Method	ScanNetv2		NRGBD	
		AbsRel↓	$\delta < 1.25$ ↑	AbsRel↓	$\delta < 1.25$ ↑
Single image	E-RayZer	0.199	0.719	0.189	0.713
	Ours	**0.098**	**0.913**	**0.108**	**0.871**
Multi-view	E-RayZer	0.167	0.820	0.095	0.907
	Ours	**0.126**	**0.871**	**0.082**	**0.916**
Ours pose AUC@30		50.6		88.4	

Table 3: Single- and multi-view depth for models trained on DL3DV dataset. In the single-image setting, each frame is fed to the model on its own.

4.2 Analyses

Unless stated otherwise, we use the full-view attention design of Sec. 3.2 and match model size and training compute across variants. We evaluate with $5, 7, 9, 11$ views sampled from a fixed 48-frame span, reporting the average across view counts unless we show each separately.

Self-supervised pretraining. We assess whether self-supervised learning improves subsequent supervised training. For supervised training (SL), we train from scratch for 200K iterations with dataset camera poses and a relative camera loss (Wang et al., 2026b). For self-supervised training (SSL), we train without geometric supervision for 100K iterations. For supervised fine-tuning (SSL+SFT), we start from SSL and receive the same supervision as SL for another 50K iterations, yielding a shorter total schedule of 150K iterations.

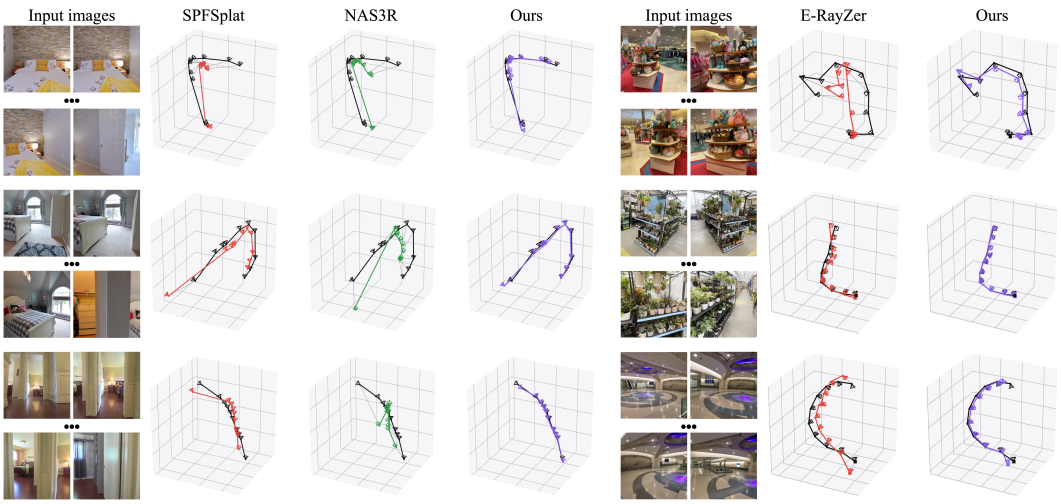


Figure 6: In-domain camera poses from 10 unposed views on RE10K (left) and DL3DV (right). Predictions are shown in color and ground truth in black, aligned by a similarity transform.

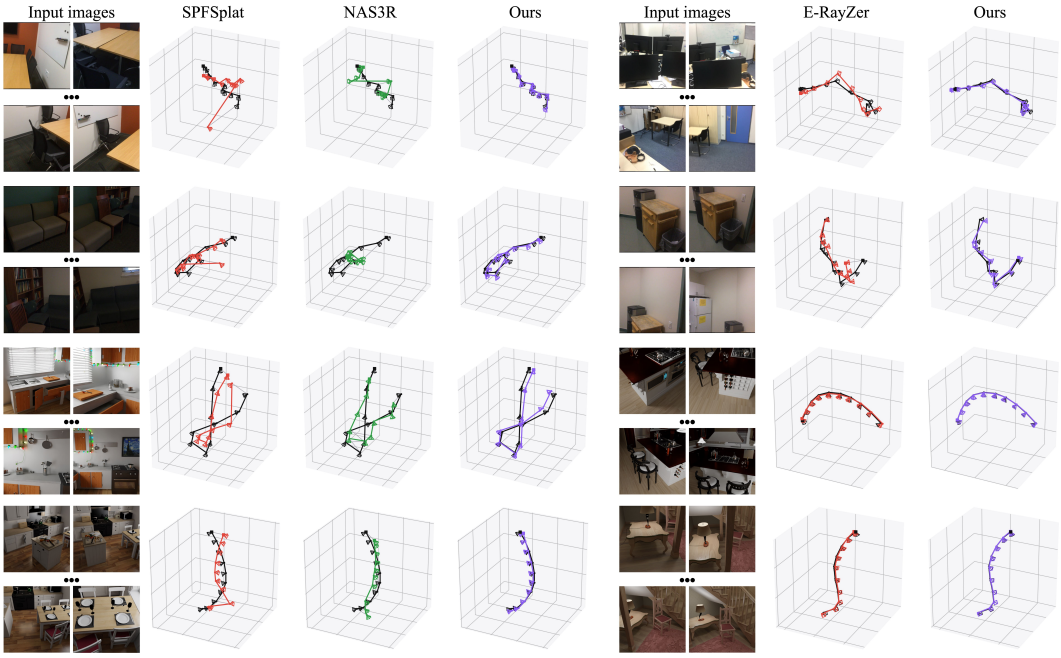


Figure 7: Out-of-domain camera poses on ScanNetv2 (top) and NRGBD (bottom), from models trained on RE10K (left) and DL3DV (right).

SSL alone achieves accuracy comparable to SL. With fine-tuning, SSL+SFT surpasses both on every reported metric (Tab. 4), despite using fewer total training iterations than SL. This result supports self-supervised learning as an effective initialization for supervised geometry estimation.

Asymmetric feature alignment. Eq. 3 aligns the intermediate features of a sparse-context model to those of a dense-context model of the same scene. We compare against depth alignment, which instead applies a scale-invariant loss between the two depth maps. The alignment loss is introduced at 40% of training and reaches full weight at 60%. We also report single-view depth, the extreme of sparse context. Tab. 5 shows that feature alignment improves every metric, and most on single-view depth (input has only one image), where the multi-view rendering loss cannot directly supervise.

Method	DL3DV (trained)				ScanNetv2				NRGBD			
	RPA@3↑	@5↑	@10↑	AUC@30↑	RPA@3↑	@5↑	@10↑	AUC@30↑	RPA@3↑	@5↑	@10↑	AUC@30↑
SSL	66.6	81.7	91.7	88.0	1.2	5.2	17.6	29.8	16.5	37.8	74.3	72.3
SL	50.8	74.6	92.5	87.2	0.7	3.2	13.5	25.2	16.3	41.2	82.7	77.2
SSL+SFT	**73.4**	**89.3**	**97.2**	**92.4**	**3.4**	**10.1**	**28.4**	**41.4**	**33.3**	**62.0**	**87.9**	**81.9**

Table 4: Camera pose accuracy of self-supervised (SSL) and supervised (SL) training, and of self-supervised pretraining followed by supervised fine-tuning (SSL+SFT).

In contrast, depth alignment does not help. Fig. 8 examines the metrics over training. Beyond standard metrics, we measure *shape*, the Pearson correlation $\rho(\log \hat{z}, \log z)$ against pseudo ground truth, which is invariant to the global scale and captures how depth is ordered across the image. Under depth alignment, shape steadily declines, so the model is getting the geometry wrong, not just the scale. Without 3D supervision, the dense-context depth is unreliable, so matching it propagates that error. Feature alignment matches representations rather than depth values, leaving depth to the rendering loss, and improves on every metric.

Method	Camera pose		Multi-view depth			Single-view depth		
	RPA@3 ↑	RPA@5 ↑	AbsRel ↓	$\delta < 1.25$ ↑	$\delta < 1.25^2$ ↑	AbsRel ↓	$\delta < 1.25$ ↑	$\delta < 1.25^2$ ↑
No alignment	66.6	81.7	0.195	0.758	0.901	0.230	0.638	0.850
Depth alignment	63.4	81.7	0.244	0.692	0.854	0.248	0.651	0.844
Feature alignment (ours)	**69.8**	**84.6**	**0.191**	**0.768**	**0.905**	**0.191**	**0.729**	**0.901**

Table 5: Context alignment analysis on DL3DV dataset.

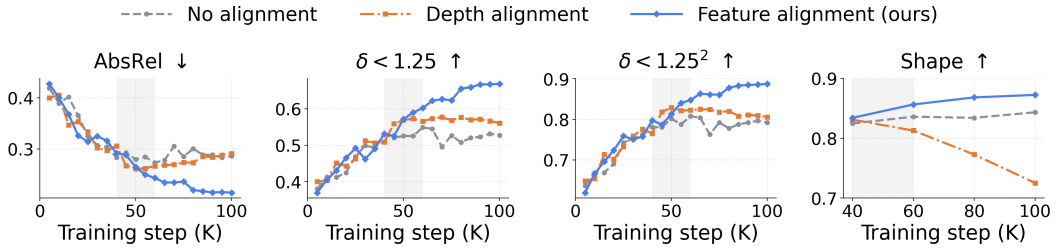


Figure 8: Single-view depth validation on DL3DV dataset over training steps. The shaded region marks the ramp-up of the alignment loss.

5 Conclusions

We study self-supervised geometry learning with rendering as the training objective and geometric accuracy as the outcome. The central design choice is where to impose asymmetry. Because overlapping views observe the same scene, withholding held-out views during feature extraction removes useful correspondences without necessarily hiding geometry. Excluding their predictions from the 3D scene prevents direct reconstruction while allowing all views to inform depth and pose. Aligning features from sparse and dense view contexts provides an additional learning signal. Together, these choices yield a simple design compatible with current 3D foundation models and improve geometric accuracy without geometric annotations.

A key opportunity is to learn from visual data beyond the reach of annotation pipelines. Our experiments focus on static scenes, where different view subsets observe the same underlying geometry. Dynamic monocular video violates this assumption because the scene changes between frames. Extending the framework to such data is an important next step. One direction uses an explicit dynamic representation, with Gaussians that move over time (Wu et al., 2024; Xu et al., 2025), or separate reconstructions at each timestep in a bullet-time formulation (Liang et al., 2024) that distinguishes object motion from camera motion. Another grounds the camera predictions of implicit models in physical geometry. These approaches offer a path to self-supervised geometry learning from the much larger supply of monocular video.

AI USE STATEMENT

Generative AI tools were used only to refine text written by the authors (grammar, readability, and style) and to assist with figure layout and LaTeX formatting. They played no role in idea generation, technical content, or experimental design. The authors reviewed all AI-assisted output and take full responsibility for the final content of this paper.

ETHICS STATEMENT

This work uses only publicly available datasets and introduces no new data. It involves no human subjects and no personally identifiable information, and we identify no ethical concerns arising from the method or its applications. The authors declare no conflicts of interest.

REPRODUCIBILITY STATEMENT

The method and objective are described in Sec. 3, and the implementation, training setup, and evaluation protocols in Sec. 4 and Sec. A.2. All datasets are publicly available, and we follow the published evaluation protocols where applicable. Source code will be released upon publication.

REFERENCES

Jiawang Bian, Zhichao Li, Naiyan Wang, Huangying Zhan, Chunhua Shen, Ming-Ming Cheng, and Ian Reid. Unsupervised scale-consistent depth and ego-motion learning from monocular video. In *Adv. Neural Inform. Process. Syst.*, 2019.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Int. Conf. Comput. Vis.*, 2021.

Hila Chefer, Patrick Esser, Dominik Lorenz, Dustin Podell, Vikash Raja, Vinh Tong, Antonio Torralba, and Robin Rombach. Self-supervised flow matching for scalable multi-modal synthesis. *arXiv preprint arXiv:2603.06507*, 2026.

Qifeng Chen and Vladlen Koltun. Photographic image synthesis with cascaded refinement networks. In *Int. Conf. Comput. Vis.*, 2017.

Xuweiyi Chen, Wentao Zhou, and Zezhou Cheng. Wildrayzer: Self-supervised large view synthesis in dynamic environments. *arXiv preprint arXiv:2601.10716*, 2026.

Yu Chen and Gim Hee Lee. Dbarf: Deep bundle-adjusting generalizable neural radiance fields. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2023.

Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017.

Tri Dao. Flashattention-2: Faster attention with better parallelism and work partitioning. In *Int. Conf. Learn. Represent.*, 2024.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, 2019.

Yang Fu, Ishan Misra, and Xiaolong Wang. Mononerf: Learning generalizable nerfs from monocular videos without camera poses. In *Int. Conf. Mach. Learn.*, 2023.

Clément Godard, Oisin Mac Aodha, Michael Firman, and Gabriel J Brostow. Digging into self-supervised monocular depth estimation. In *Int. Conf. Comput. Vis.*, 2019.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2022.

Nan Huang, Pengcheng Yu, Weijia Zeng, James M Rehg, Angjoo Kanazawa, Haiwen Feng, and Qianqian Wang. Self-improving 4d perception via self-distillation. *arXiv preprint arXiv:2604.08532*, 2026a.

Ranran Huang and Krystian Mikolajczyk. No pose at all: Self-supervised pose-free 3d gaussian splatting from sparse views. In *Int. Conf. Comput. Vis.*, 2025.

Ranran Huang, Weixun Luo, Ye Mao, and Krystian Mikolajczyk. From none to all: Self-supervised 3d reconstruction via novel view synthesis. *arXiv preprint arXiv:2603.27455*, 2026b.

Dengyang Jiang, Mengmeng Wang, Liuzhuozheng Li, Lei Zhang, Haoyu Wang, Wei Wei, Guang Dai, Yanning Zhang, and Jingdong Wang. No other representation component is needed: Diffusion transformers can provide representation guidance by themselves. *arXiv preprint arXiv:2505.02831*, 2025a.

Hanwen Jiang, Hao Tan, Peng Wang, Haian Jin, Yue Zhao, Sai Bi, Kai Zhang, Fujun Luan, Kalyan Sunkavalli, Qixing Huang, et al. Rayzer: A self-supervised large view synthesis model. In *Int. Conf. Comput. Vis.*, 2025b.

Haian 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. *arXiv preprint arXiv:2410.17242*, 2024.

Gyeongjin Kang, Jisang Yoo, Jihyeon Park, Seungtae Nam, Hyeonsoo Im, Sangheon Shin, Sangpil Kim, and Eunbyung Park. Selfsplat: Pose-free and 3d prior-free generalizable 3d gaussian splatting. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2025.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):1–14, 2023.

Karim Knaebel, Kadir Yilmaz, Daan de Geus, Alexander Hermans, David Adrian, Timm Linder, and Bastian Leibe. Dino in the room: Leveraging 2d foundation models for 3d segmentation. In *2026 International Conference on 3D Vision (3DV)*, 2026.

Zihang Lai, Sifei Liu, Alexei A Efros, and Xiaolong Wang. Video autoencoder: self-supervised disentanglement of static 3d structure and motion. In *Int. Conf. Comput. Vis.*, 2021.

Vincent Leroy, Yohann Cabon, and Jérôme Revaud. Grounding image matching in 3d with mast3r. In *Eur. Conf. Comput. Vis.*, 2024.

Hanxue Liang, Jiawei Ren, Ashkan Mirzaei, Antonio Torralba, Ziwei Liu, Igor Gilitschenski, Sanja Fidler, Cengiz Oztireli, Huan Ling, Zan Gojcic, et al. Feed-forward bullet-time reconstruction of dynamic scenes from monocular videos. *arXiv preprint arXiv:2412.03526*, 2024.

Haotong Lin, Sili Chen, Junhao Liew, Donny Y Chen, Zhenyu Li, Guang Shi, Jiashi Feng, and Bingyi Kang. Depth anything 3: Recovering the visual space from any views. *arXiv preprint arXiv:2511.10647*, 2025.

Lu Ling, Yichen Sheng, Zhi Tu, Wentian Zhao, Cheng Xin, Kun Wan, Lantao Yu, Qianyu Guo, Zixun Yu, Yawen Lu, et al. Dl3dv-10k: A large-scale scene dataset for deep learning-based 3d vision. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2024.

Chao Liu, Jinwei Gu, Kihwan Kim, Srinivasa G Narasimhan, and Jan Kautz. Neural RGB→D sensing: Depth and uncertainty from a video camera. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019.

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

Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021.

Thomas Mitchel, Hyunwoo Ryu, and Vincent Sitzmann. True self-supervised novel view synthesis is transferable. In *Int. Conf. Learn. Represent.*, 2026.

Tongzhou Mu and Rhoda AI Team. Does scaling web-video pre-training help real robots do real work? Rhoda AI Blog, 2026. URL https://www.rhoda.ai/research/scaling-web-video-pretraining.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Int. Conf. Comput. Vis.*, 2023.

Ulrich Prestel, Stefan Andreas Baumann, Nick Stracke, and Björn Ommer. Rayder: Scalable self-supervised novel view synthesis from real-world video. *arXiv preprint arXiv:2605.31535*, 2026.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. 2018.

René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *Int. Conf. Comput. Vis.*, 2021.

Mike Ranzinger, Greg Heinrich, Jan Kautz, and Pavlo Molchanov. Am-radio: Agglomerative vision foundation model reduce all domains into one. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2024.

Mehdi SM Sajjadi, Henning Meyer, Etienne Pot, Urs Bergmann, Klaus Greff, Noha Radwan, Suhani Vora, Mario Lučić, Daniel Duckworth, Alexey Dosovitskiy, et al. Scene representation transformer: Geometry-free novel view synthesis through set-latent scene representations. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2022.

Mehdi SM Sajjadi, Aravindh Mahendran, Thomas Kipf, Etienne Pot, Daniel Duckworth, Mario Lučić, and Klaus Greff. Rust: Latent neural scene representations from unposed imagery. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2023.

Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2016.

Jaeyo Shin, Jiwook Kim, and Hyunjung Shim. Representation alignment for just image transformers is not easier than you think. In *Eur. Conf. Comput. Vis.*, 2026.

Oriane Siméoni, Huy V Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, et al. Dinov3. *arXiv preprint arXiv:2508.10104*, 2025.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Adv. Neural Inform. Process. Syst.*, 2017.

Haoru Wang, Kai Ye, Minghan Qin, Yangyan Li, Wenzheng Chen, and Baoquan Chen. The less you depend, the more you learn: Synthesizing novel views from sparse, unposed images with minimal 3d knowledge. *arXiv preprint arXiv:2506.09885*, 2025a.

Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2025b.

Jianyuan Wang, Minghao Chen, Shangzhan Zhang, Nikita Karaev, Johannes Schönberger, Patrick Labatut, Piotr Bojanowski, David Novotny, Andrea Vedaldi, and Christian Rupprecht. VGGT-Ω. *arXiv preprint arXiv:2605.15195*, 2026a.

Ruicheng Wang, Sicheng Xu, Yue Dong, Yu Deng, Jianfeng Xiang, Zelong Lv, Guangzhong Sun, Xin Tong, and Jiaolong Yang. Moge-2: Accurate monocular geometry with metric scale and sharp details. In *Adv. Neural Inform. Process. Syst.*, 2025c.

Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2024.

Yifan Wang, Jianjun Zhou, Haoyi Zhu, Wenzheng Chang, Yang Zhou, Zizun Li, Junyi Chen, Jiangmiao Pang, Chunhua Shen, and Tong He. π^3: Permutation-equivariant visual geometry learning. In *Int. Conf. Learn. Represent.*, 2026b.

Ge Wu, Shen Zhang, Ruijing Shi, Shanghua Gao, Zhenyuan Chen, Lei Wang, Zhaowei Chen, Hongcheng Gao, Yao Tang, Jian Yang, Ming-Ming Cheng, and Xiang Li. Representation entanglement for generation: Training diffusion transformers is much easier than you think. In *Adv. Neural Inform. Process. Syst.*, 2025.

Guanjun Wu, Taoran Yi, Jiemin Fang, Lingxi Xie, Xiaopeng Zhang, Wei Wei, Wenyu Liu, Qi Tian, and Xinggang Wang. 4d gaussian splatting for real-time dynamic scene rendering. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2024.

Zhen Xu, Zhengqin Li, Zhao Dong, Xiaowei Zhou, Richard Newcombe, and Zhaoyang Lv. 4dgt: Learning a 4d gaussian transformer using real-world monocular videos. *arXiv preprint arXiv:2506.08015*, 2025.

Lihe Yang, Bingyi Kang, Zilong Huang, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything: Unleashing the power of large-scale unlabeled data. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2024.

Lihe Yang, Shang-Wen Li, Yang Li, Xinjie Lei, Dong Wang, Abdelrahman Mohamed, Saining Xie, Hengshuang Zhao, Kaiming He, and Hu Xu. In pursuit of pixel supervision for visual pre-training. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2026.

Yao Yao, Zixin Luo, Shiwei Li, Jingyang Zhang, Yufan Ren, Lei Zhou, Tian Fang, and Long Quan. Blendedmvs: A large-scale dataset for generalized multi-view stereo networks. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2020.

Chandan Yeshwanth, Yueh-Cheng Liu, Matthias Nießner, and Angela Dai. Scannet++: A high-fidelity dataset of 3d indoor scenes. In *Int. Conf. Comput. Vis.*, 2023.

Sihyun Yu, Sangkyung Kwak, Huiwon Jang, Jongheon Jeong, Jonathan Huang, Jinwoo Shin, and Saining Xie. Representation alignment for generation: Training diffusion transformers is easier than you think. *arXiv preprint arXiv:2410.06940*, 2024.

Qitao Zhao, Hao Tan, Qianqian Wang, Sai Bi, Kai Zhang, Kalyan Sunkavalli, Shubham Tulsiani, and Hanwen Jiang. E-rayzer: Self-supervised 3d reconstruction as spatial visual pre-training. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2026.

Tinghui Zhou, Matthew Brown, Noah Snavely, and David G Lowe. Unsupervised learning of depth and ego-motion from video. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017.

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

Chenchen Zhu, Saksham Suri, Cijo Jose, Maxime Oquab, Marc Szafraniec, Wei Wen, Yunyang Xiong, Patrick Labatut, Piotr Bojanowski, Raghuraman Krishnamoorthi, et al. Efficient universal perception encoder. *arXiv preprint arXiv:2603.22387*, 2026.

A Appendix

A.1 Design Motivation

Prior pose-free methods restrict how held-out views enter the backbone, either by excluding them from geometry prediction or by preventing context views from attending to them. This is a natural choice from the perspective of novel-view synthesis, since observing a held-out view could reveal what the model is asked to render. Our design departs from this convention while keeping the self-supervised objective intact. Our design keeps the split, and a held-out view is never used to build the scene that renders it. Held-out views instead used in the shared backbone, where they enrich cross-view correspondence. Thus, we do not use rendering metrics as the primary basis of comparison.

Our focus is geometry, with view synthesis serving as the training signal. When held-out views lie between context views, they depict a scene the context already covers and thus add correspondence rather than new content. Splitting views in image space is therefore unnecessary for the required asymmetry and instead removes correspondence the model could use. One might attribute the gain to the use of more views, but all designs receive the same images during training and differ only in how they use them. In restricted designs, held-out views do not inform the predictions for context views, from which the scene is built, whereas in our design all views inform one another. We therefore view the comparison as a question of how to recover better geometry from the same set of images, and our design as one way to make more effective use of the available views.

A.2 Additional Details

Training details. We train the main model for 100K iterations on RE10K (Zhou et al., 2018) and 150K iterations on DL3DV (Ling et al., 2024). Images are resized to 256×256 and tokenized with a patch size of 16, yielding 256 tokens per view. Each batch uses one view count from $\{5, 7, 9, 11\}$, with corresponding per-device batch sizes of $\{24, 20, 16, 12\}$ to keep the token count per step approximately constant. Following RayZer (Jiang et al., 2025b), we widen the sampling span over the first 80K iterations, progressing from shorter to longer baselines. The span grows from 24–96 to 48–192 frames on RE10K, from 24–48 to 32–54 on DL3DV, and from 24–32 to 32–48 for the analysis variants; it remains fixed thereafter. We use AdamW (Loshchilov & Hutter, 2017) with $\beta_1 = 0.9$, $\beta_2 = 0.95$, and weight decay 0.05, together with a cosine schedule and 3000 warm-up steps. The learning rate is 2×10^{-4} for the main model and 1×10^{-4} for the analysis variants. Training uses bf16 and FlashAttention-2 (Dao, 2024), with gradients clipped to a norm of 1.0.

Evaluation details. We evaluate on 5,601 RE10K scenes (Zhou et al., 2018), 140 DL3DV scenes (Ling et al., 2024), 9 NRGBD scenes (Liu et al., 2019), and 98 ScanNetv2 scenes (Dai et al., 2017). We evaluate SPFSplat and NAS3R with their official evaluation code, which applies no masking at test time. In this comparison, depth is predicted and scored on all input views for both the baselines and our model. For E-RayZer, poses are estimated from all images and depth from the context views only. Accordingly, when comparing with E-RayZer, our DPT head decodes only the context views, and depth is scored on the same context views for both methods.

Depth conversion. All depth metrics use planar z-depth. Our model and E-RayZer place each pixel-aligned Gaussian at $\mathbf{o} + r\hat{\mathbf{d}}$ ($r = d_{i,u}$ in Eq. 2), where $\mathbf{o}$ is the camera center and $\hat{\mathbf{d}}$ is the unit ray direction. Thus, r measures distance along the ray rather than z-depth. We convert it using

$$z = \frac{r}{\|\hat{K}^{-1}\mathbf{u}\|}, \tag{6}$$

where $\mathbf{u}$ denotes homogeneous pixel coordinates and $\hat{K}$ is the model's predicted intrinsic matrix. We do not use ground-truth intrinsics, so focal-length errors affect the reported depth metrics. For NAS3R, we use the per-view z-depth from its depth head. SPFSplat predicts a point map in a canonical frame; we transform it into each view using the predicted pose and read off its z coordinate. Unlike the other methods, SPFSplat receives ground-truth intrinsics as input.

Architecture. Fig. 9 illustrates the pipeline. A visual encoder tokenizes each view, and a learnable camera token is prepended to its patch tokens. Tokens from both context (C) and held-out (H) views enter a unified estimator with alternating view-wise and global attention, allowing every view to interact with the others. The resulting patch tokens feed a DPT head (Ranftl et al., 2021), which

uses features from four intermediate estimator layers to predict per-pixel depth and the remaining Gaussian attributes. Camera tokens feed a relative pose head, which regresses each pose relative to the middle input view. We unproject context-view depths using the predicted cameras to construct the Gaussian scene, then render it at the predicted held-out poses. Asymmetry is imposed at 3D representation space, thus attention remains unrestricted in the unified estimator.

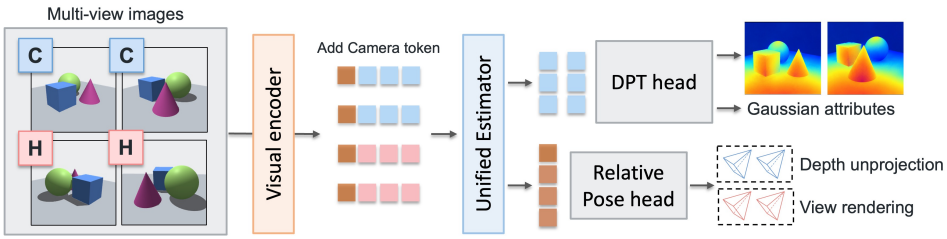


Figure 9: Overall architecture, with a shared backbone for camera and geometry prediction.

A.3 Additional Analyses

View synthesis under full-view attention. Full-view attention lets target images inform context tokens at test time, so its rendering metrics are not directly comparable. To report view synthesis without this leakage, we train with full-view attention for the first 80% of iterations and switch to masked attention for the rest, comparing against masked attention trained from scratch for the same total schedule. As shown in Fig. 11, the switched model renders consistently better, suggesting that the geometry learned under full-view attention carries over once leakage is removed.

Effect of context in the posed setting. The main experiments predict cameras jointly with geometry, so the benefit of full-view attention could come from better pose estimation rather than better geometry. To separate the two, we repeat the comparison with ground-truth cameras on DL3DV dataset. Both variants build Gaussians from context views only and differ in a single choice. FULL feeds all views, context and target, into the encoder during training, while MASK feeds only the context views. If target views offered a shortcut, FULL would learn to copy them into the rendering. If they instead supply correspondence, FULL should recover better geometry. We evaluate in two ways. In the first, both models encode only the context views at test time, so their inputs are identical. In the second, FULL encodes all views as it did during training, while MASK is unchanged. Depth is always scored on the same context views.

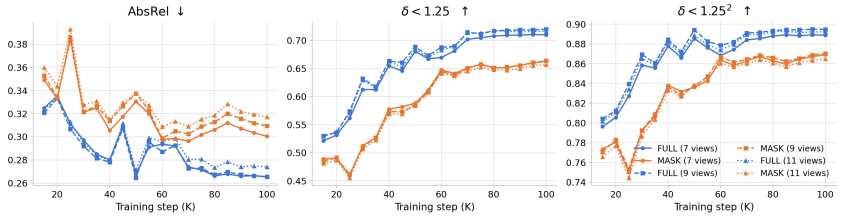


Figure 10: Depth validation over training in the posed setting.

Views	FULL					MASK				
	AbsRel ↓	$\delta < 1.25$ ↑	$\delta < 1.25^2$ ↑	PSNR ↑	LPIPS ↓	AbsRel ↓	$\delta < 1.25$ ↑	$\delta < 1.25^2$ ↑	PSNR ↑	LPIPS ↓
7 (4/3)	**0.271** (0.274)	**0.729** (0.743)	**0.887** (0.892)	20.42 (21.32)	0.365 (0.328)	0.306	0.693	0.873	**21.17**	**0.328**
9 (5/4)	**0.276** (0.281)	**0.734** (0.746)	**0.887** (0.892)	20.92 (21.74)	0.344 (0.312)	0.319	0.689	0.871	**21.68**	**0.311**
11 (6/5)	**0.276** (0.286)	**0.743** (0.748)	**0.892** (0.894)	21.27 (21.98)	0.329 (0.301)	0.319	0.688	0.872	**22.00**	**0.299**

Table 6: Effect of full-interaction in the posed setting. Views are listed as total (context/target), and depth is scored on the context views. FULL encodes only the context views at test time, matching MASK. Gray values in parentheses are from FULL encoding all views shown for reference.

More context improves geometry. With identical inputs at test time, FULL recovers better depth than MASK on every metric and at every view count (Tab. 6). Since the two models differ only in whether target views entered the encoder during training, this gain reflects how training with unconstrained interaction improves the model.

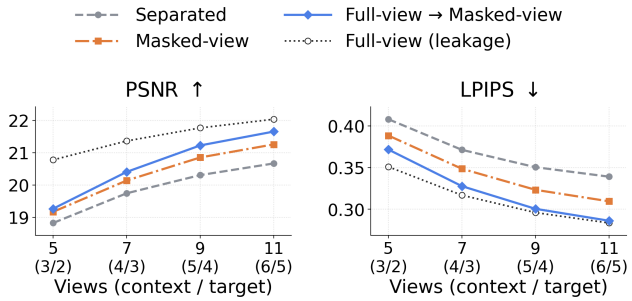


Figure 11: View synthesis evaluation. Rendering metrics on DL3DV dataset as the number of input views grows, with the context and target split shown in parentheses.

Semantic guidance. Tab. 7 isolates the effect of aligning backbone features with a frozen vision foundation model, using either Pixio (Yang et al., 2026) (Masked autoencoder) or DINOv3 (Siméoni et al., 2025) both to initialize the backbone and as the frozen model. Initializing from a pretrained encoder already matters substantially, and training from random weight lowers every metric by a wide margin. Semantic guidance adds to this for both encoders, even though the backbone starts from the same pretrained weights. The largest gain is in camera pose, particularly at stricter thresholds, and depth improves as well. The consistent gains also suggest that initialization alone does not fully retain the pretrained prior, and that keeping it as a training signal helps preserve it. Adding a projection before the alignment performs similarly, so we use the version without it. Pixio outperforms DINOv3 overall, and we use it in all other experiments.

Backbone	Method	Camera pose			Depth		NVS	
		RPA@3 ↑	RPA@5 ↑	RPA@10 ↑	AbsRel ↓	$\delta < 1.25$ ↑	PSNR ↑	LPIPS ↓
–	w/o Pretrained init	36.2	56.6	77.3	0.247	0.652	19.82	0.438
Pixio (base)	Baseline	66.6	81.7	91.7	0.195	0.758	21.48	0.312
	w/ Sem. guidance	**82.3**	**90.8**	96.0	0.185	**0.790**	**21.55**	**0.305**
	w/ Sem. guidance + Proj.	81.6	90.7	**96.2**	0.189	0.789	21.52	**0.305**
DINOv3 (base)	Baseline	46.8	64.3	80.0	0.217	0.718	20.23	0.361
	w/ Sem. guidance	62.4	79.3	91.2	**0.181**	0.778	20.40	0.361
	w/ Sem. guidance + Proj.	60.8	78.1	90.3	0.183	0.773	20.26	0.367

Table 7: Ablation of semantic guidance on DL3DV dataset. NVS metrics are reported as a reference only, since target views are processed together with the context under full-view attention.

3D probing. We test whether semantic guidance improves the learned representation by freezing each self-supervised backbone (Pixio variants) from Tab. 7 and training only its prediction heads with 3D supervision (Yao et al., 2020; Yeshwanth et al., 2023). Without pretrained initialization, performance is lower across all metrics. Semantic guidance improves on the self-supervised baseline on both datasets, with the largest gain in camera pose (Tab. 8). We evaluate on 43 BlendedMVS (Yao et al., 2020) scenes held out from the training split, and on 98 ScanNetv2 (Dai et al., 2017) scenes with the heads trained on ScanNet++ (Yeshwanth et al., 2023).

Method	BlendedMVS					ScanNetv2				
	AbsRel ↓	$\delta < 1.25$ ↑	RPA@5 ↑	RPA@15 ↑	RPA@30 ↑	AbsRel ↓	$\delta < 1.25$ ↑	RPA@5 ↑	RPA@15 ↑	RPA@30 ↑
w/o Pretrained init	0.162	0.800	4.2	16.9	26.7	0.179	0.745	0.7	9.8	28.7
Baseline	0.113	0.891	9.1	25.0	37.7	0.120	0.873	2.9	21.0	47.7
w/ Sem. guidance	**0.094**	**0.919**	**12.6**	**33.0**	**44.3**	**0.078**	**0.943**	6.2	**39.5**	**67.6**
w/ Sem. guidance + Proj.	0.096	**0.919**	11.8	28.5	41.5	0.082	0.932	**6.4**	35.7	63.0

Table 8: 3D probing with semantic guidance. Self-supervised models from Tab. 7 are used as a frozen backbone and output heads are trained with 3D supervision on each dataset. We use 10 frames per scene spanning 24 frames on BlendedMVS and 64 frames on ScanNetv2.

Comparison with neural rendering methods. We compare our explicit 3D representation with methods that render target views through a learned decoder (Sajjadi et al., 2022; Jin et al., 2024) (Tab. 9). These methods use either interpolative targets, sampled between context views (Jiang et al., 2025b; Wang et al., 2025a), or predictive targets that may lie beyond them (Mitchel et al., 2026; Prestel et al., 2026). To control for differences in data and training, we implement one variant of each within our framework. Both use the same 12-layer backbone as our model, add a 12-layer transformer decoder, and receive the same training data and compute. For the predictive variant, we follow the random-order autoregressive training design of Prestel et al. (2026). Both variants achieve better rendering quality, but neither predicts camera poses that correspond to physical camera motion. XFactor (Mitchel et al., 2026) evaluates whether learned latent poses transfer across scenes,

which tests their consistency rather than their physical accuracy. Because our goal is geometry, we evaluate predicted poses against ground truth.

Method	NVS		Camera pose				Depth	
	PSNR ↑	LPIPS ↓	RPA@3 ↑	RPA@5 ↑	RPA@10 ↑	AUC@30 ↑	AbsRel ↓	$\delta < 1.25$ ↑
Neural rendering (interpolative)	**22.28**	**0.258**	0.0	0.0	1.1	0.26	–	–
Neural rendering (predictive)	22.10	**0.258**	0.0	0.3	0.8	0.21	–	–
Ours	21.75	0.280	**85.7**	**94.1**	**98.5**	**95.12**	**0.230**	**0.775**

Table 9: Comparison with neural rendering methods on DL3DV dataset (8 views spanning 32 frames; 4 context + 4 target). The two neural rendering variants differ in whether target views are sampled between or beyond the context views during training. Our NVS metrics are grayed out, as target views are processed together with the context under full-view attention.

More qualitative results. Fig. 12 shows additional camera trajectories recovered by each method.

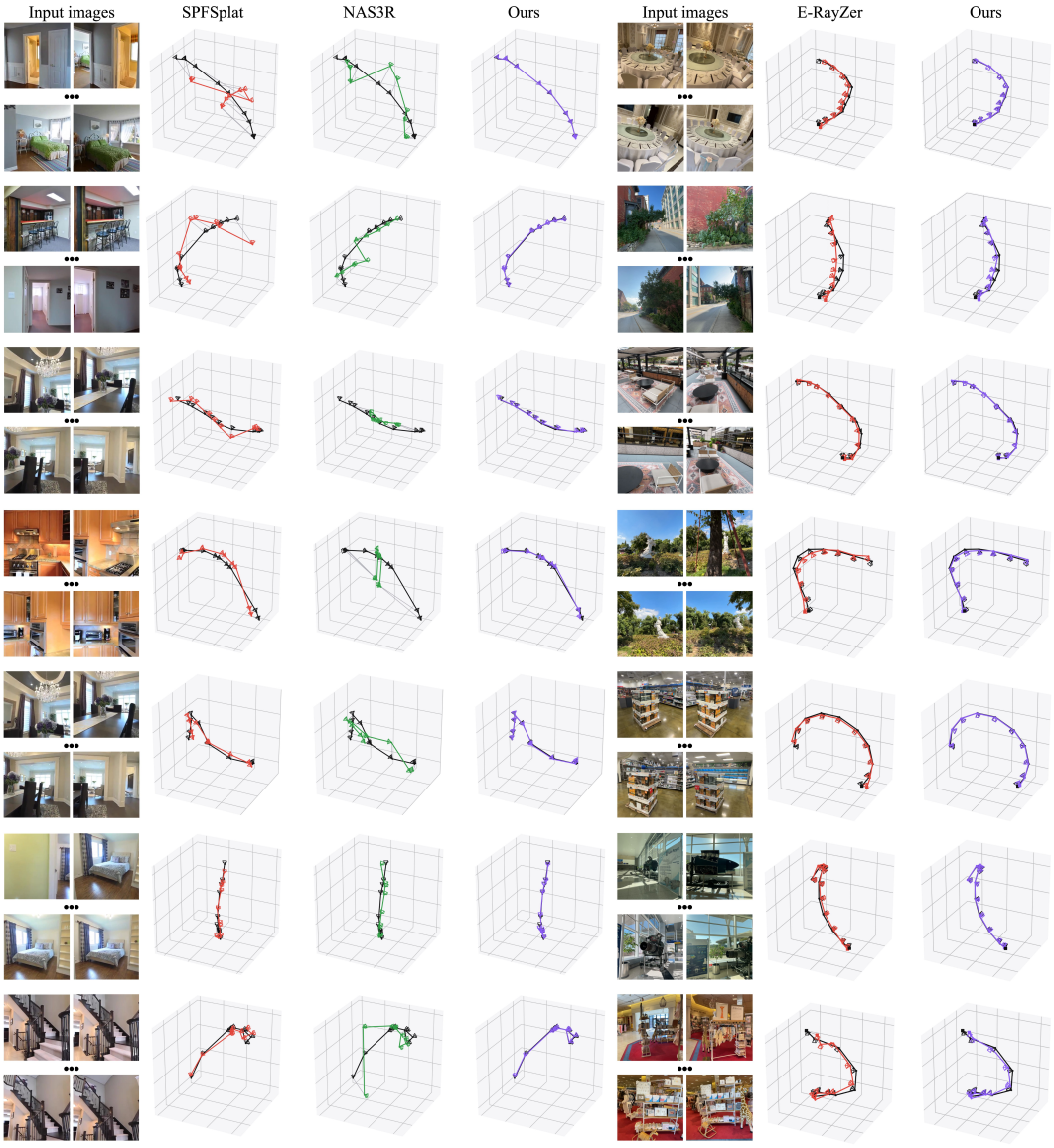


Figure 12: Additional qualitative comparison of recovered camera trajectories. In-domain camera poses from 10 unposed views on RE10K (left) and DL3DV (right). Predictions are shown in color and ground truth in black, aligned by a similarity transform.