Less Decoder is More Encoder: Geometric Representation Learning from Novel View Synthesis

Keerthi Kaashyap **Dennis Anthony** **Akshay Krishnan** **Nhi Ngoc Nguyen**
Jeremy Collins **James Hays** **Shreyas Kousik** **Animesh Garg**
Georgia Institute of Technology

Abstract

This paper examines the role of Novel View Synthesis (NVS) in geometric representation learning. In principle, NVS should reason about 3D scene structure, thereby enabling transferable multi-view geometric representations. Yet, existing encoder-based NVS methods yield poor representations. This is not because of a lack of supervisory signal, but rather due to inconspicuous architectural choices: *spatially expressive decoders* that dilute representational capabilities of the scene encoder, and *low-level pixel-space targets* that hinder feature learning. We present SNAP, a self-supervised encoder-decoder transformer that addresses both through a pose-conditioned local decoder and a latent-space reconstruction objective. SNAP is task agnostic, and we show that it is competitive with special-purpose geometry-supervised methods. SNAP also performs competitively against self-supervised representations across five tasks: visual localization, pose estimation, point correspondence, depth estimation, and robot manipulation. Remarkably, SNAP's patch features exhibit emergent viewpoint invariance that approaches heavily supervised models despite lower compute and data budgets. Under camera shifts where standard 2D representations collapse, SNAP degrades more gracefully, revealing that restricting decoder expressivity actively prevents the suppression of transferable geometric structure. https://snap-nvs.github.io

1 Introduction

Geometric understanding from images is a foundation of computer vision. Methods like DUSt3R Wang et al. [2024], MASt3R Leroy et al. [2024], and VGGT Wang et al. [2025a] demonstrate that transformers can produce rich, transferable geometric representations, when trained on large-scale annotated 3D data. This supervision requirement is a fundamental bottleneck since dense 3D ground truth is expensive to collect at scale.

Novel View Synthesis (NVS) offers a compelling alternative Jin et al. [2025], Zhao et al. [2026], Jiang et al. [2025], Kim et al. [2026]. These models must reason about 3D scene structure to predict unseen viewpoints, providing a rich geometric supervision signal only from posed images, which are abundantly available across videos, robotics platforms, and structure-from-motion pipelines. However, representations produced from feed-forward NVS methods transfer poorly to downstream geometric tasks. We argue this poor transfer is not a signal failure, but rather an architectural failure, driven by two identifiable and independently addressable choices that dilute signal.

First, spatially expressive decoders tend to absorb the representational load of the task. When a decoder has sufficient capacity to reconstruct a target view from any encoder output, the encoder is not pressured to maintain structured, locally discriminative features. While the reconstruction task is solved, the latent representations are not geometrically grounded. **Second**, pixel-space reconstruction targets require modeling high-frequency aliasing statistics that are weakly related to 3D structure and semantics. Consequently, the reconstruction loss produces gradients that encourage the representation

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

encoder to preserve fine-grained, image-specific statistics that are not directly informative about the underlying 3D structure or semantics.

We present SNAP, which addresses both failure axes with two targeted interventions. First, a *pose-conditioned local decoder* built on PROPE Li et al. [2025a] restricts the decoder's receptive field via an explicit BlockNN mask applied to queried-view self-attention. This constraint forces the geometric structure to reside within the encoder's scene representation rather than being an effect of the generative capacity of the decoder. Second, SNAP operates directly on frozen DINOv3 ViT-B patch features to provide a richer supervision signal than RGB reconstruction. This structured grounding is jointly aware of appearance, geometry, and semantics and produces better representations.

Our main contributions are:

1. **SNAP**, a self-supervised multi-view spatial reasoning backbone that builds upon frozen 2D semantic latent spaces with an asymmetric NVS prediction objective, and latent feature targets to learn geometrically transferable representations without dense 3D supervision.
2. **Mechanistic analyses** isolating the independent impact of decoder expressivity and prediction space on geometric transfer, via receptive field sweeps and prediction space landscape experiments.
3. **Empirical evaluation** across *five tasks* matching or exceeding self-supervised baselines and competitive with geometry-supervised methods that require large-scale annotated 3D data.

2 Related Work

SNAP sits at the intersection of self-supervised representation learning, novel view synthesis, and geometric foundation models. Detailed discussions are provided in Appendix A.

Self-Supervised Representation Learning. Masked Image Modeling (MIM) has evolved from pixel-reconstruction (MAE He et al. [2022], SimMIM Xie et al. [2022]) toward latent-space prediction (I-JEPA Assran et al. [2023], V-JEPA Bardes et al. [2024]). In multi-view regimes, CroCo Weinzaepfel et al. [2022] and Muskie Li et al. [2025b] leverage cross-view completion to learn implicit 3D priors. SNAP extends this trajectory by utilizing frozen DINOv3 Siméoni et al. [2025] features as opposed to pixel space targets used in Muskie Li et al. [2025b], which act as semantically structured latent distillation targets to provide a cleaner supervision signal for optimization.

Novel View Synthesis (NVS). Feed-forward NVS was established by architectures such as SRT Sajjadi et al. [2022] and RUST Sajjadi et al. [2023], recently scaling to large-view models such as LVSM Jin et al. [2025], RayZer Jiang et al. [2025], with contemporaneous works LagerNVS Szymanowicz et al. [2026] and SVSM Kim et al. [2026] continuing to extend the paradigm. However, most of these variants utilize expressive decoders that absorb the representational load of the task, leading to ungrounded latents. By constraining the decoder's receptive field via a BlockNN mask, adopting PROPE Li et al. [2025a], and targeting a latent feature space, SNAP ensures its representations prioritize 3D structural awareness over reconstruction statistics.

Geometry-Supervised Models. Recent foundation models, including DUSt3R Wang et al. [2024], MASt3R Leroy et al. [2024], VGGT Wang et al. [2025a], and D4RT Zhang et al. [2026], achieve state-of-the-art performance but rely on massive 3D-annotated datasets. Similarly, π^3 Wang et al. [2026] and Depth Anything 3 Lin et al. [2026] provide robust spatial priors via explicit 3D supervision. Conversely, SNAP approaches the geometric transferability of these models while remaining self-supervised and training on significantly less data and compute.

3 Method

SNAP operates in two stages, as illustrated in Figure 1. First, a pose-free multi-view encoder aggregates frozen DINOv3 patch features from N reference views into scene tokens $\mathbf{Z}$ via full cross-view self-attention. Notably, no camera parameters are provided at this stage. Second, a pose-conditioned patch decoder then queries $\mathbf{Z}$ using PROPE-rotated cross-attention to synthesize the target view's feature map. The target and context camera intrinsics and extrinsics are fed into PROPE at this stage. Two design choices jointly force geometric structure into the encoder: a BlockNN mask restricting the decoder's spatial receptive field (Sec. 3.3), and a latent reconstruction objective against frozen DINOv3 teacher features rather than RGB pixels (Sec. 3.4). We train SNAP on scenes from RealEstate10K Zhou et al. [2018], DL3DV Ling et al. [2024], and Co3Dv2 Reizenstein et al. [2021]; comprehensive implementation and architectural details are deferred to Appendix B

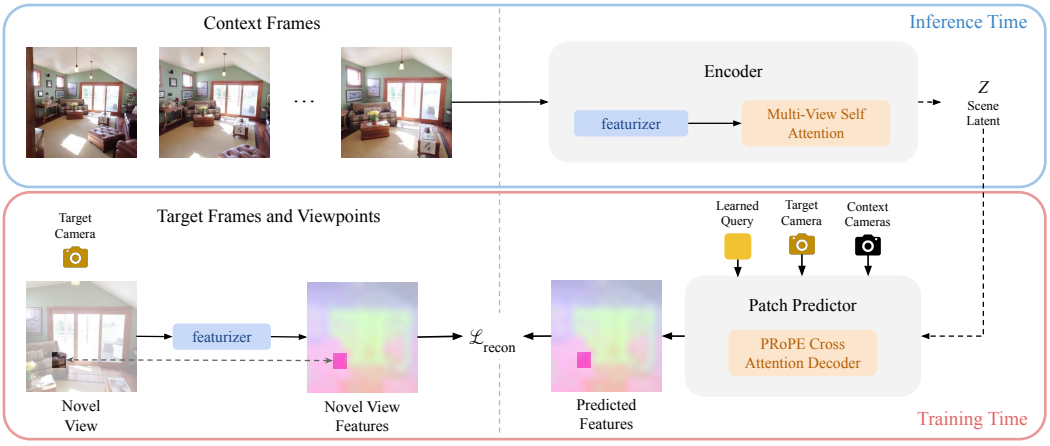

Figure 1: Training and inference pathways. Top (Inference): A pose-free multi-view encoder processes reference frames via a frozen DINOv3 featurizer (φ) and cross-view self-attention to construct an implicit 3D scene latent Z. **Bottom (Training):** A pose-conditioned local decoder queries Z using PRoPE to predict the feature map of a novel target view. Supervision ($\mathcal{L}_{\text{recon}}$) is applied in a semantic latent space against a frozen DINOv3 teacher, completely bypassing pixel-level rendering and forcing geometric structure into the encoder.

3.1 Notation

We use $\{\mathbf{I}_k, T_k, \mathbf{K}_k\}_{k=1}^N$ to denote N reference views, where $\mathbf{I}_k \in \mathbb{R}^{3\times H\times W}$ is an RGB image, $T_k \in \mathrm{SE}(3)$ its camera-to-world pose with respect to T_1 (with T_1 set as the identity) and $\mathbf{K}_k \in \mathbb{R}^{3\times 3}$ its intrinsic matrix. During inference, a set of target poses $\{T_j^*\}$ and intrinsics $\{\mathbf{K}_j^*\}$ are provided for one or more target views. $L = (H/p) \times (W/p)$ denotes the number of $p\times p$ patches per image, φ a frozen ViT featurizer (DINOv3 Siméoni et al. [2025]) with output dimension d, and the projection matrix $\mathbf{P}_k = \hat{\mathbf{K}}_k \cdot T_k^{-1}$ maps world-space 3D coordinates to image-space patch coordinates with normalized intrinsics $\hat{\mathbf{K}}_k$.

3.2 Pose-Free Multi-View Encoder as a Spatial Backbone

SNAP's pose-free multi-view encoder aggregates reference features $\mathbf{F}_k = \varphi(\mathbf{I}_k) \in \mathbb{R}^{L\times d}$ into unified scene tokens $\mathbf{Z} = \mathcal{E}_{\text{RoPE2D}}(\{F_k\}) \in \mathbb{R}^{NL\times E}$. Each reference image $\mathbf{I}_k$ is patchified by the frozen DINOv3 encoder into $\mathbf{F}_k \in \mathbb{R}^{L\times d}$, then projected to the transformer working dimension E via a learned linear layer to yield $X_k \in \mathbb{R}^{L\times E}$. RoPE2D Heo et al. [2024] is applied, to queries and keys within self-attention layers, encoding the intra-image 2D patch grid position (i, j). The projected multi-view sequence $[X_1; \ldots; X_N] \in \mathbb{R}^{NL\times E}$ is processed by a multi-view transformer with L_e cross-view self-attention blocks. In full cross-view self-attention, each patch attends to all patches across all reference views. The output $\mathbf{Z}$ is a pose-free scene representation with implicit geometry.

3.3 Pose-Conditioned Patch Decoder as a Feature Predictor

For the second stage of SNAP, a pose-conditioned patch decoder predicts the target feature map:

$$\hat{\mathbf{F}} = \mathcal{D}_{\text{PRoPE}}(\mathbf{Z}; \{\mathbf{P}_j^*\}, \{\mathbf{P}_k\}) \in \mathbb{R}^{L\times d}. \tag{1}$$

Our decoder $\mathcal{D}$ is a latent novel-view predictor, which predicts the patchified latent features $\mathbf{F}^*$ from the camera parameters of the target and context views. We employ PRoPE Li et al. [2025a] for pose-conditioning in the decoder, which consists of alternating blocks of PRoPE-rotated cross-attention against $\mathbf{Z}$ and block-masked self-attention among target queries which is motivated in Sec. 4.1. A single learned feature query $\mathbf{q} \in \mathbb{R}^{1\times E}$ is broadcast into a sequence of L identical target patch queries $\mathbf{Q} \in \mathbb{R}^{L\times E}$. Spatial symmetry is broken entirely by PRoPE, which differentiates the learned queries across viewpoints using the target intrinsics $\mathbf{K}_j^*$ and pose T_j^*. During cross-attention, the target queries $\mathbf{Q}$ serve as q, while the encoder's scene tokens $\mathbf{Z}$ provide k and v. PRoPE constructs projection matrices $\mathbf{P}_k = \hat{\mathbf{K}}_k \cdot T_k^{-1}$ and rotates queries and keys such that:

$$\mathrm{Attn}(q,\, k,\, v) = \mathrm{softmax}\left(\frac{(\mathbf{P}_j^{*\top} q)\,(\mathbf{P}_k^{-1} k)^\top}{\sqrt{E/H}}\right) v, \tag{2}$$

where H is the number of attention heads and $\mathbf{P}_j^* = \hat{\mathbf{K}}_j^* \cdot T_j^{*-1}$ is the projection matrix for target view j. A token pair scores high attention when their world-space rays are co-aligned emulating a soft epipolar search.

Following cross-attention, lateral communication among target queries is restricted by a non-overlapping BlockNN mask. Every patch is assigned to an exclusive spatial block of size RF×RF. Self-attention is only permitted between queries falling within the same block, explicitly bottlenecking the spatial receptive field of the decoder. As demonstrated in Sec. 4.1, isolating spatial communication forces geometric reasoning into the encoder; therefore, in our optimal SNAP architecture, we strictly enforce RF $= 1$, reducing target-query self-attention to a pointwise residual MLP.

3.4 Latent Reconstruction Objective

Training minimizes

$$\mathcal{L}_{\text{recon}} = \underbrace{\frac{1}{Ld}\sum_{p,d}\left|\hat{\mathbf{F}}_{p,d} - \mathbf{F}^*_{p,d}\right|}_{\mathcal{L}_{\text{feat}}} + \underbrace{\frac{1}{2}\left(\|\nabla_x\hat{\mathbf{F}} - \nabla_x\mathbf{F}^*\|_1 + \|\nabla_y\hat{\mathbf{F}} - \nabla_y\mathbf{F}^*\|_1\right)}_{\mathcal{L}_{\text{grad}}}, \tag{3}$$

where ∇_x, ∇_y are first-order spatial finite-differences over the 2D token grid. We apply an L_1 penalty rather than cosine similarity to the normalized features to promote sharper, sparsity-inducing gradients at structural boundaries. Both $\hat{\mathbf{F}}$ and the ground-truth target map $\mathbf{F}^* = \varphi(\mathbf{I}^*)$ are ℓ_2-normalised onto the unit hypersphere before supervision.

$\mathcal{L}_{\text{grad}}$ is not auxiliary: without it, $\mathcal{D}$ can satisfy $\mathcal{L}_{\text{feat}}$ with spatially-uniform predictions that average out local structure, collapsing the geometric signal the encoder is required to encode. Penalising finite-difference residuals enforces spatial fidelity precisely at object boundaries and texture edges, where patch-wise ℓ_1 is least discriminative.

By routing supervision through DINOv3 features rather than RGB, we replace a weak, low-level training signal with more structured guidance. Unlike raw pixels, which contain nuisance variation and little inherent bias toward semantics, DINOv3 features reside on a structured manifold and provide dense, spatially localized embeddings. Section 4.2 motivates this choice and quantifies its effect via a systematic sweep over DINOv3 layers.

3.5 Multitask Evaluation of Representation

We probe the geometric content of frozen $\mathbf{Z}$ on five tasks spanning correspondence, localization, and pose estimation. In every case $\mathcal{E}$ is held frozen; only a lightweight task-specific head is trained.

(T1) Point Correspondence – Given views A and B, a point correspondence head $\texttt{PC}(\mathbf{Z}_A, \mathbf{Z}_B, \mathbf{c}_A)$ predicts the normalized coordinate $\hat{\mathbf{c}}_B \in [0,1]^2$ in view B matching query $\mathbf{c}_A \in [0,1]^2$ in view A. Both views are encoded under identity poses; pixel-unit intrinsics (f_x, f_y, c_x, c_y) preserve field-of-view. Ground-truth correspondences are supplied at training time by the frozen RoMa v2 matcher Edstedt et al. [2026]; pairs with fewer than $\tau_{\min}$ confident matches are discarded. The objective is $\mathcal{L}_{\text{corr}} = \|\hat{\mathbf{c}}_B - \mathbf{c}_B^*\|_1$. Evaluation: $\text{pct@}\{5, 10\}$ px in pixels and Average Pixel Error.

(T2) Visual Localization. – Given N reference views with known poses and a single unposed query image, a visual localization head $\texttt{VL}(\mathbf{Z}_{\text{ref}}, \mathbf{Z}_{\text{tgt}})$ regresses the absolute target pose as a 6D rotation Zhou et al. [2019] and 3D translation, all tokens extracted under identity poses. Let $\mathcal{H}_\beta(a,b) = \text{SmoothL1}(a,b,\beta)$; the loss is

$$\mathcal{L}_{\text{loc}} = \mathcal{H}_{0.05}(R_{\text{pred}}, R^*) + 0.5\mathcal{H}_{0.05}(t_{\text{pred}}, t^*). \tag{4}$$

Evaluation: AUC at $\{20°, 30°\}$, where joint AUC integrates $\min(\text{recall}_R, \text{recall}_t)$ normalized to the threshold range.

(T3) Multi-View Pose Estimation. – Given N images in an *unknown* coordinate frame, a pose estimation sequence head $\texttt{PES}(\mathbf{Z})$ uses N learned view queries to simultaneously regress all poses in a single forward pass. Per-view validity indicators $m_k \in \{0,1\}$ mask padded entries:

$$\mathcal{L}_{\text{seq}} = \frac{1}{\sum_k m_k}\sum_{k=1}^{N} m_k\big(\mathcal{H}_{0.05}(R_k, R_k^*) + 0.5\mathcal{H}_{0.05}(t_k, t_k^*)\big). \tag{5}$$

Metrics are identical to T2, marginalized over all valid views.

(T4) Pointmap and Depth Estimation. – To test dense 3D scene structure, a DPT Ranftl et al. [2021] readout head predicts relative depth for each patch token. Evaluated on Hypersim Roberts et al. [2021] and 7Scenes Shotton et al. [2013] using L_1 relative depth error (AbsRel) and δ_1, the percentage of pixels satisfying $\max(\hat{d}/d, d/\hat{d}) < 1.25$.

(T5) Robot Manipulation. – To evaluate viewpoint robustness in embodied settings, frozen encoder tokens serve as the visual state for an imitation learning policy. Evaluated on MimicGen Mandlekar et al. [2023] under systematic camera shifts (0.0 to 1.0 radians), measuring task success rate.

4 Analysis: Decoder Expressivity and Prediction Target

Our objective is to learn scene representations that transfer across geometric tasks, not simply to optimize novel-view reconstruction. These goals can diverge: an encoder-decoder NVS model Jin et al. [2025], Szymanowicz et al. [2026] may reconstruct target views accurately while placing little transferable spatial structure in the multi-view encoder. We identify two key design factors that influence whether the learned multi-view representation encodes transferable spatial structure: decoder expressivity and the prediction target.

4.1 Axis 1: Decoder Expressivity Degrades Geometric Transfer

Decoder expressivity. In encoder-decoder NVS models, the decoder's capacity dictates the balance between its own learned 2D priors and the encoded conditioning signal. A highly expressive decoder acts as a strong unconditional prior, utilizing local spatial context to reconstruct or smooth images independently. While this improves raw synthesis quality, it creates an optimization shortcut: the strong prior weakens the model's reliance on the conditioning signal, relieving the encoder of the need to capture precise, multi-view geometry. Consequently, the downstream utility of the learned representation degrades. In transformer-based architectures, the strength of this prior is primarily controlled by the number of decoder layers and the receptive field of target-view self-attention.

Prior NVS models commonly use highly expressive decoders with full self-attention in the reconstruction of the target-view Jin et al. [2025], Szymanowicz et al. [2026]. We study the influence of decoder expressivity by varying the receptive field of these self-attention blocks.

Experiment To isolate this factor, we train SNAP variants with varying decoder receptive fields, RF $\in \{1\times1, 2\times2, 4\times4, 8\times8, 16\times16\}$. We control the receptive field using a BlockNN mask, which restricts the decoder's self-attention to non-overlapping $k\times k$ spatial windows. Expanding the mask increases the amount of target-view spatial context the decoder can aggregate, while leaving the decoder architecture otherwise unchanged. At the strictest setting, RF $= 1\times1$, spatial communication between target queries is eliminated, reducing the decoder to a patchwise residual MLP. At the broadest setting, RF $= 16\times16$, the window spans the full feature map, giving the decoder full self-attention in target-view, as in Jin et al. [2025], Szymanowicz et al. [2026].

Figure 2 demonstrates increasing decoder receptive field improves the reconstruction objective but degrades downstream geometric transfer. As RF grows, the decoder is able to use more target-view spatial context, leading to lower feature reconstruction error. However, the frozen encoder representation becomes less useful for geometry: point correspondence error, visual localization, and pose estimation all degrade. This inverse relationship suggests that reconstruction quality alone is a poor proxy for representation quality in encoder-decoder NVS models. The qualitative results in Figure 3 show the same trend. With RF $= 1\times1$, similarity maps are sharply localized around the corresponding point and remain stable across viewpoints, while with complete self-attention (RF $= 16\times16$), the maps do not always track the ground truth correspondence. These results demonstrate limiting target-view spatial communication prevents the decoder from absorbing reconstruction priors, encouraging transferable spatial structure into the multi-view encoder, motivating our patch decoder.

4.2 Axis 2: Prediction Space Determines Representation Quality

The prediction target determines what information the model is pressured to encode. Recent NVS models typically supervise reconstruction in pixel space Jin et al. [2025], Szymanowicz et al. [2026], which can emphasize low-level appearance, texture, and illumination rather than transferable geometry. We instead use latent feature prediction, where earlier ViT layers remain closer to RGB while deeper layers yield high-level structural abstractions that are robust to non-geometric artifacts.

We sweep both the DINOv3 layer that is used as the featurized input and the layer used as the prediction target, asking which feature spaces are optimal for novel-view prediction, both at the input and at the target. We evaluate the resulting encoders on RE10K and DL3DV (val subset),

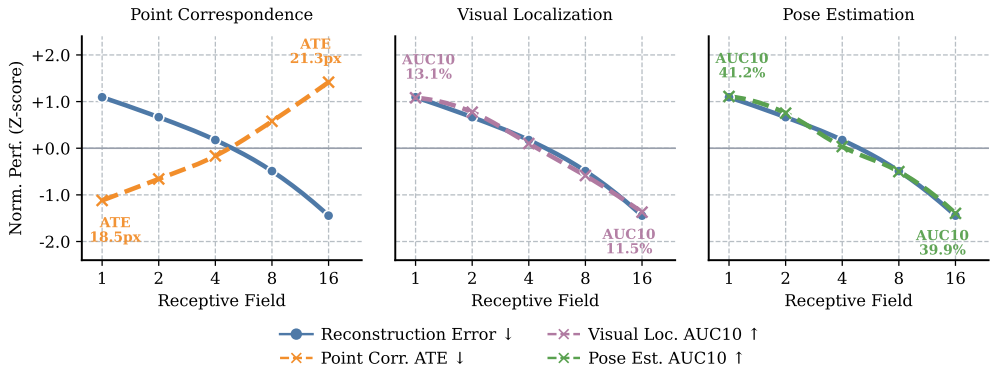

Figure 2: Effect of Receptive field across three geometric tasks on RE10K. Normalized performance (Z-score) vs. RF on a $\log_2$ scale. Blue circles: reconstruction error (lower is better); dashed colored lines: downstream task performance (higher is better). **Left:** Point correspondence reveals the critical crossover: as RF expands, reconstruction error falls while correspondence error *rises*. The decoder's improving reconstruction comes entirely at the cost of encoder spatial structure. **Center & Right:** Visual localization and pose estimation performance degrade monotonically as RF increases, implying that as decoder expressivity increases, downstream representational performance decreases.

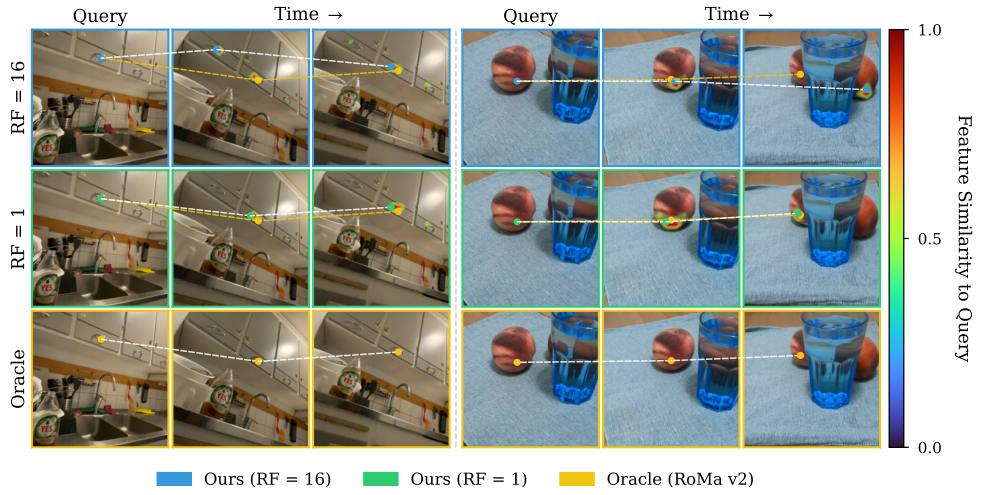

Figure 3: Receptive field qualitative comparison. Each row tracks a query point (colored dot) and its feature similarity map across time. While a large receptive field (RF $= 16\times16$) yields more scattered similarity maps, restricting the receptive field to RF $= 1\times1$ captures richer and more precise geometric correspondences albeit slightly more diffuse heatmaps, tracking with the Oracle across both indoor (left) and object (right) scenes.

reporting feature reconstruction error, nearest-neighbor point correspondence (PCK@0.1), patch retrieval (MRR), and intrinsic dimensionality using the TwoNN estimator Facco et al. [2017].

Figure 4 shows that the easiest reconstruction targets do not necessarily produce the best spatial representations. Lower-layer targets are closer to RGB and are easier to predict, with the best reconstruction error obtained by the (input: 9, target: 2) pairing. However, downstream geometry improves when predicting higher-layer DINOv3 features, suggesting that *more semantic latent targets provide a better supervisory signal for transferable spatial structure*. The sweep also reveals a trade-off in the input feature space: input layer 4 gives the best in-distribution PCK and MRR, improving over layer 11 by roughly 5.8% and 5.2%, respectively, but its representation has $\sim 53\%$ higher intrinsic dimensionality. We therefore use layer 11 as input, accepting a small loss in in-distribution accuracy for a more compressed and invariant multi-view representation. A comprehensive analysis of these trade-offs across all layers and metrics is provided in Appendix G.

5 Experiments

We evaluate SNAP across five tasks using frozen encoder features with lightweight task-specific decoder heads. This protocol isolates the intrinsic spatial structure learned during pretraining, testing out-of-the-box generalization without confounding effects of task-specific fine-tuning data.

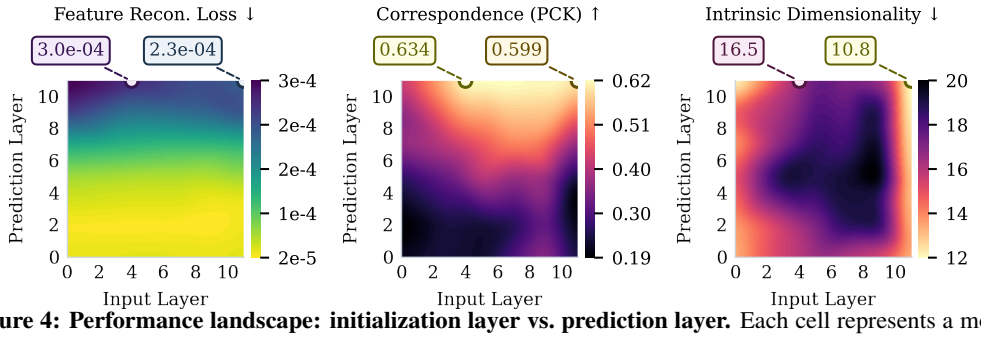

Figure 4: Performance landscape: initialization layer vs. prediction layer. Each cell represents a model where the encoder outputs from the "Input Layer" and the decoder predicts features from the "Prediction Layer" of a 12-layer DINOv3 VIT-B (layers 0-11). Color encodes performance across the evaluated metrics, scaled such that yellow always indicates better performance. For downstream geometric tasks like correspondence (middle), predicting mid-to-high-layer targets generally yields the strongest relative performance. Conversely, predicting raw RGB or early-layer targets (bottom rows) consistently degrades these downstream geometric metrics. Notably, downstream representational capacity does not perfectly align with intrinsic dimensionality (right) or feature reconstruction (left) which is the validation cosine similarity; while RGB targets yield poor geometric features, they maintain a lower intrinsic dimensionality than several deep-layer prediction configurations.

Baselines. We compare to different kinds of methods: **feed-forward NVS models** without any intermediate geometry prediction (LVSM Jin et al. [2025], LagerNVS Szymanowicz et al. [2026], RayZer Jiang et al. [2025]), **feed-forward 3DGS** prediction models (E-RayZer Zhao et al. [2026]), and feed-forward **models trained for geometry prediction** itself (VGGT Wang et al. [2025a]). We also compare to recent work on learning **multi-view representations using masked modeling** methods (Muskie Li et al. [2025b]). This spans a wide range of feed-forward multi-view representations.

To evaluate domain transfer, we split our protocol: point correspondence and relative pose are evaluated on the validation splits of our pretraining datasets (RealEstate10K Zhou et al. [2018], DL3DV Ling et al. [2024]), while depth estimation and robot manipulation are evaluated on domains unseen by the vision backbone (Hypersim, 7Scenes, MimicGen) to evaluate out-of-domain generalization.

For perceiver-style NVS methods whose latents are not spatially organized (LVSM, RayZer), we cross-attend their latents to a shared $H \times W$ spatial grid and compute all downstream features from the resulting spatial grid tokens, ensuring a fair comparison with spatially indexed encoder methods.

Importantly, despite employing full cross-view attention, SNAP maintains a significantly smaller memory footprint than alternative models (Appendix E). By operating on 256×256 images with a patch size of 16, our encoder processes a sequence length of only $256N + 8$ tokens (including registers). In contrast, VGGT processes 518×518 images at a patch size of 14, resulting in $1369N + 5$ tokens; it would require more than 12 views for our model to remotely approach the attention cost of these baselines. Combined with Grouped Query Attention (GQA) configured with 16 heads and 8 groups, SNAP alleviates KV costs, operating efficiently even under limited compute budgets.

5.1 Point Correspondence

Point correspondence tests spatial discrimination at the pixel level: a match is correct only if the feature at one pixel is more similar to its true correspondent than to all other image pixels. Ground-truth correspondences for evaluation are supplied by a state-of-the-art dense matcher (RoMa v2 Edstedt et al. [2026]) acting as a correspondence oracle. As Table 1 shows, SNAP outperforms self-supervised baselines on RealEstate10K across all pixel thresholds. Remarkably, SNAP achieves this higher sub-patch precision despite processing inputs at half the resolution of VGGT (256×256 vs. 518×518), underscoring the spatial sharpness of our learned representations. On DL3DV, where scene complexity is higher, SNAP performs competitively with VGGT despite utilizing strictly no dense 3D supervision, approaching its performance within fractional margins across all distance thresholds. NVS baselines (ERayZer, LVSM, RayZer) consistently underperform, confirming that high reconstruction fidelity does not imply spatial feature discriminability.

5.2 Visual Localization

Visual localization requires matching a query image against a reference map and estimating absolute camera pose, thereby combining per-patch discriminability with global geometric consistency under

Table 1: Point correspondence. Pixel-level matching precision (PCK) and average pixel error (APE). Top results are highlighted in **bold**. SNAP outperforms baselines on RE10K and remains competitive on DL3DV and Hypersim without explicit 3D supervision.

	RealEstate10K			DL3DV			Hypersim		
	PCK ↑		APE ↓	PCK ↑		APE ↓	PCK ↑		APE ↓
Method	@5px	@10px		@5px	@10px		@5px	@10px	
LVSM	5.51	14.76	41.04	3.53	11.05	45.75	5.62	13.41	42.15
RayZer	3.81	13.53	39.63	2.73	13.53	47.03	4.21	11.21	44.82
ERayZer	4.59	12.47	39.67	3.55	11.26	45.30	4.31	14.89	40.54
LagerNVS	9.61	27.36	26.54	3.81	11.31	42.41	8.94	24.30	31.02
Muskie	12.09	34.43	20.90	3.58	11.39	41.78	12.71	34.15	21.15
VGGT	12.85	32.78	21.79	4.15	**13.65**	39.34	14.92	**35.38**	**19.14**
SNAP	**15.48**	**36.29**	**18.41**	**4.20**	13.33	**39.26**	**15.01**	34.87	19.82

Table 2: Visual localization and relative pose estimation performance. Metrics include Rotation Accuracy at 5° (R_5), Translation Accuracy at 5° (T_5), and AUC at 20° (A_{20}) and 30° (A_{30}). Top results are highlighted in **bold**. SNAP demonstrates superior geometric precision among self-supervised and NVS methods.

	Visual Localization (AUC / Acc ↑)								Relative Pose Estimation (AUC / Acc ↑)							
	RE10K				DL3DV				RE10K				DL3DV			
Method	R_5	T_5	A_{20}	A_{30}	R_5	T_5	A_{20}	A_{30}	R_5	T_5	A_{20}	A_{30}	R_5	T_5	A_{20}	A_{30}
LVSM	69.87	8.48	20.62	29.39	20.39	3.55	6.02	11.82	70.31	38.98	43.03	46.97	54.41	32.37	27.65	30.18
RayZer	56.30	7.29	13.74	20.10	22.85	2.92	5.13	10.01	78.32	**42.89**	53.00	59.60	58.32	34.91	32.11	36.10
ERayZer	59.09	6.65	15.47	23.53	24.04	3.05	5.99	12.03	77.82	40.45	46.63	54.07	50.82	31.29	23.56	27.32
LagerNVS	58.65	9.42	17.07	25.01	22.81	7.35	5.79	12.05	70.81	30.74	35.67	42.17	57.81	36.91	34.96	44.13
Muskie	79.65	11.50	23.25	32.56	23.02	8.54	6.90	13.85	80.26	40.09	44.63	53.07	65.36	38.38	47.54	54.64
VGGT	74.31	9.58	25.01	34.70	22.98	**11.83**	**8.86**	**17.36**	**82.15**	42.36	**59.36**	**68.74**	**74.25**	**44.48**	**59.32**	**67.94**
SNAP	**80.32**	**15.47**	**26.41**	**34.72**	**24.62**	8.59	7.58	15.91	78.81	41.42	53.07	62.30	65.31	39.91	49.31	57.43

large viewpoint change. We evaluate by training a lightweight regression head on frozen encoder features to directly output a 6D rotation matrix and 3D translation. As Table 2 shows, SNAP establishes a new frontier among self-supervised models on most metrics, confirming that features from restricted receptive field generalize beyond local correspondence to global pose inference.

5.3 Relative Pose Estimation

Relative pose estimation uses a direct regression head over the full encoder latent, including register tokens, with early-stage cross-view fusion. On RealEstate10K (Table 2), SNAP achieves high rotation accuracy among NVS-based methods and is competitive with other self-supervised and NVS methods on translation accuracy. The result confirms that SNAP's encoder retains global geometric reasoning simultaneously with the local discriminative structure demonstrated in Table 1 — the local decoder bottleneck does not degrade large-scale geometric inference.

5.4 Robot Manipulation Under Viewpoint Shift

Beyond static benchmarks, we evaluate whether SNAP can transfer effectively to online decision-making. We compare SNAP against a representative suite of baselines, including semantic backbones (DINOv3, ResNet), geometry-supervised encoders (VGGT), and special-purpose pipelines (Adapt3R Wilcox et al. [2025]) on robotic manipulation under out-of-distribution viewpoint shifts. We evaluate on three MimicGen tasks under camera shifts of 0.0 to 1.0 radians using frozen encoder backbones. Despite not being trained on any robotics-related data, SNAP is competitive across the board.

As Figure 5 shows, semantic encoders (DINOv3, ResNet) collapse rapidly beyond 0.2 radians. Arguably, 2D contrastive training suppresses the cross-view spatial variations which the policy requires to infer 3D configuration. Conversely, SNAP's multi-view latent reconstruction objective induces strong viewpoint invariance out-of-the-box. Notably, SNAP outperforms its own frozen DINOv3 teacher, generally outperforms task-specific baseline Adapt3R, and tracks closely with the fully geometry-supervised VGGT baseline across all tasks and shift magnitudes.

5.5 Relative Depth Estimation

Relative depth estimation requires predicting scale-invariant depth for each image patch, a direct test of whether an encoder has internalized 3D scene structure. We train a depth head on Hypersim and evaluate performance on 7Scenes and Hypersim validation split. Notably, while LagerNVS

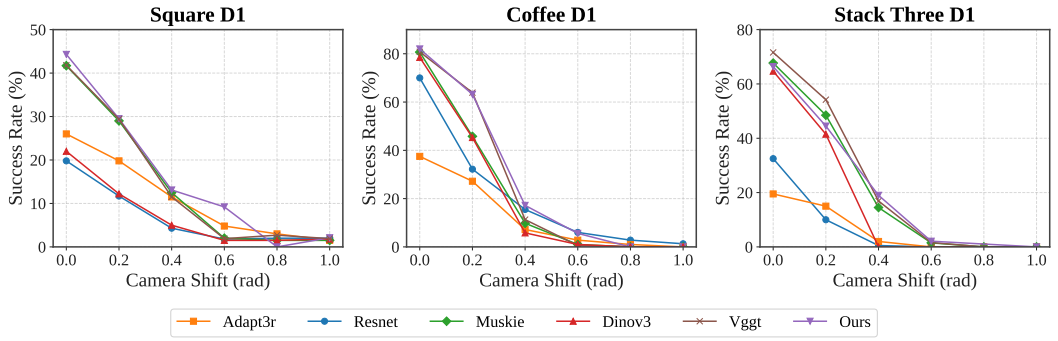

Figure 5: Emergent Viewpoint Invariance in Robot Manipulation. Success rate across three MimicGen tasks evaluated under severe camera shifts (0.0 to 1.0 radians). While standard 2D semantic priors (DINOv3, ResNet He et al. [2016]) collapse rapidly under viewpoint perturbation, SNAP (Ours) maintains highly robust spatial representations, tracking closely with the geometry-supervised VGGT ceiling and systematically outperforming methods explicitly designed for spatial robustness (Adapt3R).

was pre-trained on Hypersim and operates at a 518 resolution, LVSM and SNAP backbones are not pre-trained on Hypersim. Furthermore, LVSM utilizes a smaller $p = 8$ patch size, providing a higher-resolution latent grid than SNAP. As Table 3 shows, SNAP outperforms NVS-based encoders. Using explicit 3D supervision (VGGT) performs better in this task. While exposure to Hypersim during pre-training yields high synthetic δ_1 for LagerNVS, SNAP demonstrates strong generalization on the real-world 7-Scenes dataset. We supply qualitative results in Appendix F.

Table 3: Relative depth estimation. We evaluate relative depth prediction on a validation-split of Hypersim, and 7Scenes, a real-world held-out dataset. We report AbsRel and δ_1, the percentage of pixels satisfying $\max(\hat{d}/d, d/\hat{d}) < 1.25$. Reporting for VGGT comes from the off-the-shelf depth estimation head.

	Hypersim		7Scenes	
Method	AbsRel ↓	δ_1 ↑	AbsRel ↓	δ_1 ↑
VGGT	*0.2080*	*94.6036*	*1.9138*	*70.0987*
LVSM	2.0648	38.5086	1.9474	32.5691
LagerNVS	0.7441	**73.1069**	2.3858	49.2679
ERayZer	1.1321	50.6075	1.3320	35.5672
SNAP (ours)	**0.7093**	64.6815	**0.9257**	**52.0218**

6 Conclusion

While our current pretraining relies on posed video sequences and static, textured scenes, and our findings are bound by a fixed compute and data budget that has yet to reveal the full scaling laws of the architecture, SNAP demonstrates that feed-forward novel view synthesis (NVS) is a legitimate powerhouse for self-supervised geometric pretraining. Extending this paradigm to dynamic video remains a critical next step in establishing this approach as a foundation model. More broadly, our results show that NVS's rich geometric signal is most available when two architectural pathologies are corrected: decoders expressive enough to absorb spatial reasoning that should belong to the encoder, and pixel-space targets that contaminate gradients with low-level statistics.

Restricting decoder receptive field via a BlockNN mask biases spatial reasoning back into the scene tokens; replacing pixel targets with last-layer DINOv3 features further redirects optimization toward a structured, geometry-privileged manifold. The resulting backbone rivals geometry-supervised models across five tasks without dense 3D annotations, and transfers robustly to robot manipulation, where viewpoint shifts break standard 2D semantic models. Moreover Appendix D suggests that these improvements are not restricted to a single teacher, improving correspondence among DINOv3, SAM, and MAE alike. Most strikingly, SNAP demonstrates NVS's geometric signal is best leveraged by restricting decoder capacity, not by optimizing reconstruction, underscoring that representation quality does not always correspond with reconstruction fidelity.

Funding Transparency Statement

Funding. The computational experiments in this work were conducted with resources provided by the Partnership for an Advanced Computing Environment (PACE) at the Georgia Institute of Technology. Dennis Anthony was supported by the Georgia Tech Research Institute (GTRI) HIVES IRAD Project I15RD.HI.02.11.

Competing Interests. The authors declare no competing financial interests relevant to this work.

References

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

Vincent Leroy, Yohann Cabon, and Jérôme Revaud. Grounding image matching in 3d with mast3r. In *European conference on computer vision*, pages 71–91. Springer, 2024.

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*, 2025a.

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. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=QQBPWtvtcn.

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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7525–7535, June 2026.

Hanwen Jiang, Hao Tan, Peng Wang, Haian Jin, Yue Zhao, Sai Bi, Kai Zhang, Fujun Luan, Kalyan Sunkavalli, Qixing Huang, and Georgios Pavlakos. Rayzer: A self-supervised large view synthesis model. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4918–4929, October 2025.

Evan Kim, Hyunwoo Ryu, Thomas W. Mitchel, and Vincent Sitzmann. Scaling view synthesis transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026.

Ruilong Li, Brent Yi, Junchen Liu, Hang Gao, Yi Ma, and Angjoo Kanazawa. Cameras as relative positional encoding. In *Advances in Neural Information Processing Systems*, 2025a.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

Zhenda Xie, Zheng Zhang, Yue Cao, Yutong Lin, Jianfeng Bao, Zhuliang Yao, Qi Dai, and Han Hu. SimMIM: A simple framework for masked image modeling. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. Self-supervised learning from images with a joint-embedding predictive architecture. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

Adrien Bardes, Quentin Garrido, Jean Ponce, Michael Rabbat, Yann LeCun, Mahmoud Assran, and Nicolas Ballas. V-JEPA: Latent video prediction for visual representation learning. *arXiv preprint arXiv:2404.08471*, 2024.

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. *Advances in Neural Information Processing Systems*, 35:3502–3516, 2022.

Wenyu Li, Sidun Liu, Peng Qiao, Yong Dou, and Tongrui Hu. Muskie: Multi-view masked image modeling for 3d vision pre-training, 2025b. URL https://arxiv.org/abs/2511.18115.

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, 2025. URL https://arxiv.org/abs/2508.10104.

Mehdi S. M. Sajjadi, Henning Meyer, Etienne Pot, Urs Bergmann, Klaus Greff, Noha Radwan, Suhani Vora, Mario Lučić, Daniel Duckworth, Alexey Dosovitskiy, Jakob Uszkoreit, Thomas Funkhouser, and Andrea Tagliasacchi. Scene representation transformer: Geometry-free novel view synthesis through set-latent scene representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6229–6238, June 2022.

Mehdi S. M. Sajjadi, Aravindh Mahendran, Thomas Kipf, Etienne Pot, Daniel Duckworth, Mario Lučić, and Klaus Greff. Rust: Latent neural scene representations from unposed imagery. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 17297–17306, June 2023.

Stanislaw Szymanowicz, Minghao Chen, Jianyuan Wang, Christian Rupprecht, and Andrea Vedaldi. Lagernvs: Latent geometry for fully neural real-time novel view synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15443–15453, June 2026.

Chuhan Zhang, Guillaume Le Moing, Skanda Koppula, Ignacio Rocco, Liliane Momeni, Junyu Xie, Shuyang Sun, Rahul Sukthankar, Joëlle K. Barral, Raia Hadsell, Zoubin Ghahramani, Andrew Zisserman, Junlin Zhang, and Mehdi S. M. Sajjadi. Efficiently reconstructing dynamic scenes one d4rt at a time. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7382–7392, June 2026.

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 C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust, editors, *International Conference on Learning Representations*, volume 2026, pages 10481–10497, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/11a09e0aaa74867c6b0719c639fc09f8-Paper-Conference.pdf.

Haotong Lin, Sili Chen, Jun Hao Liew, Donny Y. Chen, Zhenyu Li, Yang Zhao, Sida Peng, Hengkai Guo, Xiaowei Zhou, Guang Shi, Jiashi Feng, and Bingyi Kang. Depth Anything 3: Recovering the visual space from any views. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=yirunib8l8.

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.

Lu Ling, Yichen Sheng, Zhi Tu, Wentian Zhao, Cheng Xin, Kun Wan, Lantao Yu, Qianyu Guo, Zixun Yu, Yawen Lu, et al. Dl3dv-10k: A large-scale scene dataset for deep learning-based 3d vision. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 22160–22169. IEEE, 2024.

Jeremy Reizenstein, Roman Shapovalov, Philipp Henzler, Luca Sbordone, Patrick Labatut, and David Novotny. Common objects in 3d: Large-scale learning and evaluation of real-life 3d category reconstruction. In *International Conference on Computer Vision*, 2021.

Byeongho Heo, Song Park, Dongyoon Han, and Sangdoo Yun. Rotary position embedding for vision transformer. In *European Conference on Computer Vision*, pages 289–305. Springer, 2024.

Johan Edstedt, David Nordström, Yushan Zhang, Georg Bökman, Jonathan Astermark, Viktor Larsson, Anders Heyden, Fredrik Kahl, Mårten Wadenbäck, and Michael Felsberg. RoMa v2: Harder Better Faster Denser Feature Matching. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2026. URL https://eccv.ecva.net/virtual/2026/poster/3967.

Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. On the continuity of rotation representations in neural networks. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 12159–12168. IEEE, 2021.

Mike Roberts, Jason Ramapuram, Anurag Ranjan, Atulit Kumar, Miguel Angel Bautista, Nathan Paczan, Russ Webb, and Joshua M. Susskind. Hypersim: A photorealistic synthetic dataset for holistic indoor scene understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10912–10922, October 2021.

Jamie Shotton, Ben Glocker, Christopher Zach, Shahram Izadi, Antonio Criminisi, and Andrew Fitzgibbon. Scene coordinate regression forests for camera relocalization in RGB-D images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2930–2937, 2013.

Ajay Mandlekar, Soroush Nasiriany, Bowen Wen, Iretiayo Akinola, Yashraj Narang, Linxi Fan, Yuke Zhu, and Dieter Fox. MimicGen: A data generation system for scalable robot learning using human demonstrations. In *Conference on Robot Learning (CoRL)*, 2023.

Elena Facco, Maria d'Errico, Alex Rodriguez, and Alessandro Laio. Estimating the intrinsic dimension of datasets by a minimal neighborhood information. *Scientific Reports*, 7(1), September 2017. ISSN 2045-2322. doi: 10.1038/s41598-017-11873-y. URL http://dx.doi.org/10.1038/s41598-017-11873-y.

Albert Wilcox, Mohamed Ghanem, Masoud Moghani, Pierre Barroso, Benjamin Joffe, and Animesh Garg. Adapt3r: Adaptive 3d scene representation for domain transfer in imitation learning. In Joseph Lim, Shuran Song, and Hae-Won Park, editors, *Proceedings of The 9th Conference on Robot Learning*, volume 305 of *Proceedings of Machine Learning Research*, pages 1486–1514. PMLR, 27–30 Sep 2025. URL https://proceedings.mlr.press/v305/wilcox25a.html.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

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 *ECCV*, 2020.

Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Trans. Graph.*, 41(4):102:1–102:15, July 2022. doi: 10.1145/3528223.3530127. URL https://doi.org/10.1145/3528223.3530127.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4), July 2023. URL https://repo-sam.inria.fr/fungraph/3d-gaussian-splatting/.

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.

Qianqian Wang, Zhicheng Wang, Kyle Genova, Pratul P. Srinivasan, Howard Zhou, Jonathan T. Barron, Ricardo Martin-Brualla, Noah Snavely, and Thomas Funkhouser. Ibrnet: Learning multi-view image-based rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

Anpei Chen, Zexiang Xu, Fuqiang Zhao, Xiaoshuai Zhang, Fanbo Xiang, Jingyi Yu, and Hao Su. Mvsnerf: Fast generalizable radiance field reconstruction from multi-view stereo. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 14124–14133, 2021.

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)*, pages 19457–19467, 2024.

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. In *European Conference on Computer Vision (ECCV)*, 2024.

Stanislaw Szymanowicz, Christian Rupprecht, and Andrea Vedaldi. Splatter image: Ultra-fast single-view 3d reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10208–10217, 2024.

Thomas Mitchel, Hyunwoo Ryu, and Vincent Sitzmann. True self-supervised novel view synthesis is transferable. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=aJJppqAm6r.

Qianqian Wang, Yifei Zhang, Aleksander Holynski, Alexei A. Efros, and Angjoo Kanazawa. Continuous 3D perception model with persistent state. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025b.

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 Kontschieder. MapAnything: Universal feed-forward metric 3D reconstruction. In *International Conference on 3D Vision (3DV)*. IEEE, 2026.

Zhongxiao Cong, Qitao Zhao, Minsik Jeon, and Shubham Tulsiani. Flow3r: Factored flow prediction for scalable visual geometry learning. In *CVPR*, 2026.

David Nordström, Johan Edstedt, Fredrik Kahl, and Georg Bökman. Mum: Multi-view masked image modeling for 3d vision. *arXiv preprint arXiv:2511.17309*, 2025.

Mohamed El Banani, Amit Raj, Kevis-Kokitsi Maninis, Abhishek Kar, Yuanzhen Li, Michael Rubinstein, Deqing Sun, Leonidas Guibas, Justin Johnson, and Varun Jampani. Probing the 3d awareness of visual foundation models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 21795–21806, 2024.

Yue Chen, Xingyu Chen, Anpei Chen, Gerard Pons-Moll, and Yuliang Xiu. Feat2gs: Probing visual foundation models with gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

Zixuan Huang, Xiang Li, Zhaoyang Lv, and James M. Rehg. How much 3d do video foundation models encode? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 384–394, June 2026.

Joshua Ainslie, James Lee-Thorp, Michiel De Jong, Yury Zemlyanskiy, Federico Lebrón, and Sumit Sanghai. Gqa: Training generalized multi-query transformer models from multi-head checkpoints. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pages 4895–4901, 2023.

Ali Abouzeid, Malak Mansour, Qinbo Sun, Zezhou Sun, and Dezhen Song. Geoaware-vla: Implicit geometry aware vision-language-action model. *arXiv preprint arXiv:2509.14117*, 2025.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=St1giarCHLP.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollar, and Ross Girshick. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4015–4026, October 2023.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. IEEE, 2009.

Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

Lorenzo Mur-Labadia, Matthew Muckley, Amir Bar, Mido Assran, Koustuv Sinha, Mike Rabbat, Yann LeCun, Nicolas Ballas, and Adrien Bardes. V-JEPA 2.1: Unlocking dense features in video self-supervised learning, 2026. URL https://arxiv.org/abs/2603.14482.

Supplementary Material

A Related Work

A.1 Feed-forward novel-view synthesis

Novel-view synthesis has traditionally been addressed through per-scene optimization of explicit or implicit 3D representations, including neural radiance fields and 3D Gaussian splatting Mildenhall et al. [2020], Müller et al. [2022], Kerbl et al. [2023]. These methods produce high-quality renderings, but require optimizing a new scene representation at test time. A parallel line of work instead relies on reconstruction priors learned from data, using feed-forward models for generalizable view synthesis. Early examples condition radiance fields or image-based rendering networks on sparse input views Yu et al. [2021], Wang et al. [2021], Chen et al. [2021], while recent methods directly predict feed-forward 3D Gaussian scene representations Charatan et al. [2024], Chen et al. [2024], Szymanowicz et al. [2024]. More recently, large transformer-based NVS models have moved further away from 3D representations, synthesizing target views directly from learned multi-view representations or scene tokens Jin et al. [2025], Jiang et al. [2025], Zhao et al. [2026], Mitchel et al. [2026], Szymanowicz et al. [2026], Kim et al. [2026]. These models are typically evaluated by image reconstruction quality, but their learned intermediate representations have also been proposed as general-purpose multi-view scene representations. Our work investigates aspects of NVS pretraining to improve the learned representation for downstream tasks, rather than only for reconstructing novel views.

A.2 Multi-view representation learning

Beyond novel-view synthesis, several works learn multi-view representations explicitly for geometric reasoning. CroCo introduced cross-view completion as a self-supervised pretraining objective for two-view 3D vision Weinzaepfel et al. [2022]. This representation was later used as the foundation for DUSt3R and MASt3R, which used pointmap prediction to solve many diverse tasks Wang et al. [2024], Leroy et al. [2024]. Recent multi-view generalist models such as VGGT, CUT3R, π^3, MapAnything, D4RT, and Flow3R further extend this direction, predicting camera parameters, depth, pointmaps, tracks, or dynamic scene structure from arbitrary image collections Wang et al. [2025a,b, 2026], Keetha et al. [2026], Zhang et al. [2026], Cong et al. [2026]. Many of the strongest geometric foundation models rely on explicit 3D supervision, such as depth, pointmaps, camera poses, or tracks, which limits the scale and diversity of usable training data. Self-supervised alternatives such as Muskie and MuM Li et al. [2025b], Nordström et al. [2025] revisit masked modeling in the multi-view setting. Our work is complementary: rather than using masked reconstruction as the pretext task, we study novel-view prediction as a scalable self-supervised signal, and identify how decoder expressivity and prediction target determine whether NVS produces transferable spatial representations.

A.3 3D awareness in video and multi-view models

Recent work studies whether pretrained 2D, video, and multi-view foundation models encode useful 3D structure. El Banani et al. [2024] probes frozen 2D visual foundation models for single-view geometry and cross-view surface consistency, finding that some self-supervised image features encode non-trivial 3D information. Feat2GS Chen et al. [2025] further probes visual foundation models by reading out 3D Gaussian attributes from frozen features, using novel-view synthesis to evaluate geometry and texture awareness. VidFM3D Huang et al. [2026] shows that video foundation model features also contain 3D-aware information extractable with lightweight geometry heads. Our work asks a complementary question: what makes NVS pretraining produce transferable spatial representations? Rather than only probing for 3D information after pretraining, we isolate two design factors that shape the learned encoder representation: decoder expressivity and the prediction target. We also evaluate our representation on a range of geometric and robotics tasks.

Among these, the architectures closest to ours are SVSM Kim et al. [2026], LagerNVS Szymanowicz et al. [2026], and D4RT Zhang et al. [2026], all concurrent works with distinct objectives. SVSM and LagerNVS both follow prior work in pairing the scene representation with an expressive decoder, which is the main architectural difference from our approach; as we show in Sec. 4.1, this design choice likely limits how well their learned representations transfer to downstream multi-view tasks. Within this pair, SVSM focuses on the data scale required for pretraining, while LagerNVS instead focuses on distilling priors from geometry-pretrained models such as VGGT. D4RT, in contrast,

adopts a lightweight cross-attention decoder similar to ours, but trains it with supervised geometry targets, whereas we rely on a purely self-supervised NVS objective.

B Implementation Details

B.1 Pretraining Setup

Datasets and Splits. We train SNAP on a large-scale mixture of multiview video datasets totaling approximately 117,000 sequences. This training set consists of the RealEstate10K training partition ($\sim$67,000 sequences), DL3DV-10K ($\sim$10,000 sequences), and Co3Dv2 ($\sim$40,000 sequences). For the DL3DV and Co3Dv2 datasets, we hold out 20% of the available sequences for validation, and we use the preprovided test split for validation on RE10K. During training, we randomly sample up to $N = 6$ reference views and one target view per sequence. All images are center-cropped and resized to 256×256 pixels. We use 16 layers of self attention for the multi-view encoder with embedding dimension of 1024 and 8 blocks of alternating PROPE Li et al. [2025a] cross-attention and masked self attention. We leverage grouped-query attention Ainslie et al. [2023] operating with 16 query heads and 8 KV-groups.

Optimization. Training was conducted for 40,000 steps using an effective batch size of 256 across multiple classes of GPU hardware using Distributed Data Parallel (DDP). We employ the AdamW optimizer with a peak learning rate of 1×10^{-4} and a weight decay of 0.02. The learning rate follows a linear warmup for the first 2,000 steps, followed by a cosine annealing schedule that decays toward zero. We apply gradient clipping with a maximum norm of 0.5 to maintain training stability.

Exponential Moving Average (EMA). To stabilize the learned representations, we maintain an Exponential Moving Average of all trainable parameters with a decay rate of 0.999. The EMA scope encompasses every component of the architecture, including the pose-free multi-view encoder and the pose-conditioned local decoder, which serves as the latent feature reconstruction predictor. All empirical evaluations and downstream geometric probes reported in the main text utilize these EMA weights.

B.2 Point Correspondence

Label Generation. Training labels are generated using image pairs from the train splits of RealEstate 10K, DL3DV-10K, and Hypersim. We employ two distinct strategies depending on the dataset:

- **RE10K and DL3DV:** We derive pseudo-ground-truth correspondences from a frozen RoMa v2 matcher. To ensure high-quality supervision, we filter for image pairs containing at least $\tau_{min} = 8$ confident matches.
- **Hypersim:** We compute exact ground-truth correspondences via cross-projection using provided depth maps and camera poses.

Architecture. The point correspondence head employs a three-stage mechanism: (1) symmetric cross-attention layers that map the encoder's latent tokens to a canonical 16×16 spatial grid; (2) a source-view cross-attention block to localize the discriminative feature at query coordinate c_A; and (3) a matching stage that computes angular similarity between the query and target-view features via L_2 normalized dot products.

Specifically, the decode stage (1) consists of 3 cross-attention layers, and the localization stage (2) consists of 3 cross-attention layers. In the final matching stage (3), independent linear projections are applied to the queries and keys to prevent gradient interference. The L_2 normalized dot product is scaled by a learnable, log-parameterized temperature scalar (initialized to 1.0) before calculating the similarity heatmap. A differentiable spatial expectation over the resulting similarity heatmap produces the final predicted coordinate.

Optimization. The correspondence head is trained for 20,000 steps with an effective batch size of 128. We utilize the AdamW optimizer with a peak learning rate of 2×10^{-4}, a 2000-step linear warmup, and a cosine annealing schedule. As specified in our evaluation protocol, all views are encoded under identity camera-to-world poses with pixel-unit intrinsics to isolate the encoder's intrinsic geometric awareness from external pose signals. Evaluation occurs on the held-out validation splits on all datasets.

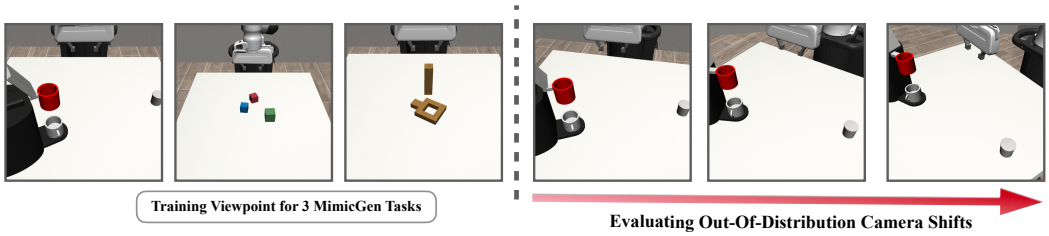

Figure 6: Training and evaluation setting for robotics tasks: **Left**: Three MimicGen simulation task we train our downstream policies on. **Right**: To test the model's emergent viewpoint invariance, we evaluate its performance on out-of-distribution camera shifts ranging from 0.2 to 1.0 radians, none of which were encountered during training.

B.3 Visual Localization

Architecture. The visual localization head is designed to regress an absolute target pose by localizing an unposed query image against a reference multi-view context. The target pose is parameterized using 9 parameters: a continuous 6D rotation representation and a 3D translation vector.

The regression head utilizes alternating transformer layers of self-attention (processing the target sequence) and cross-attention (reading from the reference context). A learned CLS token is prepended to the target image sequence, and 1D RoPE is applied to encode sequence positions. After processing the tokens through the transformer blocks, the refined CLS token is extracted and passed through a final multi-layer perceptron to output the 9 pose parameters. To ensure stable early training, the final linear layer's biases are initialized to represent an identity transformation.

Optimization. We train this head using the RealEstate 10K and DL3DV train splits, and evaluation is conducted on their respective held-out validation splits. Optimization uses a peak learning rate of 2×10^{-5} with a cosine annealing schedule and 2,500 steps of linear warmup. The network minimizes a 9D pose regression loss using a SmoothL1 penalty on both the predicted rotation R_{pred} and translation t_{pred}.

B.4 Pose Estimation

Architecture. The relative pose estimation head predicts the pose sequence for all N input views simultaneously in a single forward pass. Similar to the visual localization head, this module uses transformer layers alternating between self-attention and cross-attention. It queries the multi-view context using N learned view queries (one corresponding to each view) equipped with 1D RoPE. The output tokens are processed by a regression MLP (Linear $\rightarrow$ GELU $\rightarrow$ Linear) to predict the 9D pose (6D rotation and 3D translation) for each respective view. Per-view validity indicators mask out padded entries during the loss computation.

Optimization. The sequence head is trained on the RealEstate 10K and DL3DV train splits and evaluated on their held-out validation splits. Training utilizes a peak learning rate of 2×10^{-5}, governed by a cosine annealing schedule with 2,500 warmup steps. The model is optimized using a 9D pose regression loss based on the Smooth-L1 error of the rotation and translation predictions.

C Robot Manipulation

Here we discuss details about the sim experiments, including detailed information about the inputs to the policies, hyperparameter choices, etc.

Tasks. – We evaluate on three MimicGen manipulation tasks: STACK THREE-D1, SQUARE-D1, and COFFEE-D1. Each task is solved by a single 7-DoF Franka Panda arm with a parallel-jaw gripper. For every task we train on the full MimicGen demonstration set of 1,000 demonstrations generated from the source human demos. Observations are RGB images from two cameras: a fixed third-person camera and a wrist-mounted camera on the robot. The policy is only conditioned on these visual features.

Vision Encoders. – We benchmark six vision backbones with otherwise identical training and policy hyperparameters. The encoders we benchmark are the following: (i) **ImageNet-pretrained ResNet-18**; (ii) **Adapt3R**, a 3D-aware encoder built on a CLIP RN50 backbone with a point-cloud attention extractor and pyramid feature network; (iii) **DINOv3** ViT-B/16; (iv) **Muskie**, a multi-view backbone with alternating attention; (v) **VGGT**, a geometry-aware transformer aggregator; (vi) **SNAP** (ours), a

multi-view backbone built on a frozen DINOv3 featurizer. For VGGT and SNAP we are using their patch, register, and camera (only applicable to VGGT) tokens. Note that all multi-view encoders see both cameras as input views (V=2), whereas single-view encoders (ResNet, Adapt3R, DINOv3) process each camera independently and their tokens are concatenated downstream. As for the policy, most encoders (Muskie, DINOv3, ResNet, SNAP) we use their last layer features to condition the policy. For VGGT, we use the element-wise mean of all four intermediate aggregator-layer features, preserving the per-token shape, which has been shown to an extent to provide better viewpoint robustness Abouzeid et al. [2025].

Policy. – For all encoders we use the same diffusion policy, a Diffusion-Transformer (DiT) that predicts a chunk of H_a=8 future actions in delta pose representation, of which the first H_e=4 are executed (action-horizon) before re-planning. The policy uses 1 encoder block and 3 transformer decoder blocks with hidden dimension d_{embed}=512, 4 attention heads, GELU activations, and dropout 0.1. The diffusion process uses 100 training steps and 8 DDIM Song et al. [2021] sampling steps at evaluation. We train all models using AdamW (β_1=0.95, β_2=0.999) with a learning rate of 1×10^{-4} and weight decay of 1×10^{-6}, a cosine schedule with 1,000 warm-up steps decaying to 1% of the peak, batch size 128, and gradient clipping at 100 for 350 epochs. Models use embedding dimension d_e=512, chunk size 8, and action horizon 4. Results are reported over 400 rollouts per task per policy.

Image Resolution and Augmentation. – We use a common training resolution of 256×256 for DINOv3, SNAP. For VGGT and Muskie we used a resolution of 224×224. ResNet and Adapt3R are trained at 128×128 resolution (following a similar spirit as Adapt3r's training resolution). The same lightweight augmentation is applied to every encoder: per-batch color jitter (brightness, contrast, saturation, hue) followed by a random translation crop of ±8 pixels.

D Comparison with Teachers

D.1 Comparison with Frozen Teachers

SNAP uses a frozen visual teacher to provide feature targets for its feature-space reconstruction objective. While our main experiments use DINOv3 Siméoni et al. [2025], the proposed multi-view pretraining objective is not specific to this teacher. We therefore compare SNAP using frozen DINOv3 Siméoni et al. [2025], SAM Kirillov et al. [2023], and MAE He et al. [2022] features.

Point Correspondence. – As shown in Table 4, SNAP substantially improves point correspondence for every teacher, dataset and threshold. [1]

On held-out Hypersim, PCK@0.1 improves from 29.29 to 55.35 with DINOv3 Siméoni et al. [2025], from 40.82 to 52.27 with SAM Kirillov et al. [2023], and from 11.01 to 50.67 with MAE He et al. [2022]. The gains are largest at tight thresholds and shrink at PCK@0.3 for the stronger teachers (DINOv3: 84.22 to 88.35; SAM: 78.74 to 83.64), which approach saturation, while MAE still improves by 23.35 points (60.81 to 84.16). The same trends hold on RealEstate10K and DL3DV. Thus, the geometric gains are not specific to the DINOv3 feature space. Instead, the multi-view, pose-conditioned reconstruction objective consistently improves the spatial consistency and geometric utility of the underlying teacher representation.

Among the base teachers, DINOv3 is strongest at PCK@0.3, whereas SAM is stronger (or tied) at PCK@0.1 and PCK@0.2 on all three datasets (e.g., Hypersim PCK@0.1: 40.82 vs. 29.29). After SNAP, the spread between teachers narrows considerably (Hypersim PCK@0.1 ranges from 50.67 to 55.35, versus 11.01 to 40.82 for the base features), and DINOv3 yields the strongest SNAP representation in all but one setting (DL3DV PCK@0.1). DINOv3 is therefore a good choice but not a requirement for SNAP. The substantial improvements obtained with SAM Kirillov et al. [2023] and MAE He et al. [2022] demonstrate that the gains arise from combining high-order abstract representations with the view synthesis objective rather than simply inheriting the capabilities of the DINOv3 teacher. The remaining differences between teachers show that the quality of the underlying feature space is still significant.

[1] All point-correspondence numbers in this appendix are computed via training free methods and use global soft-argmax matching with a low temperature ($\tau = 0.02$): each source keypoint is compared against the full target feature map, and the prediction is the soft-argmax of the resulting similarity map. Results are aggregated over multiple random seeds, for roughly 2 million evaluated pairs in total. SNAP, DINOv3, SAM and MAE use ViT-B backbones; V-JEPA 2.1 uses a ViT-L variant, a larger model that favors the baseline.

Table 4: Point correspondence with frozen teacher features. PCK (%) at thresholds 0.1, 0.2 and 0.3 for frozen teacher features (Base) and the corresponding SNAP representations, using global soft-argmax matching ($\tau = 0.02$). Hypersim is held out from SNAP's training. SNAP improves over its teacher for every teacher, dataset and threshold.

		RealEstate10K			DL3DV			Hypersim		
Teacher	Features	@0.1	@0.2	@0.3	@0.1	@0.2	@0.3	@0.1	@0.2	@0.3
DINOv3 Siméoni et al. [2025]	Base	45.83	80.13	93.71	28.00	62.53	84.45	29.29	63.67	84.22
	SNAP	**78.21**	**91.32**	**95.62**	**58.74**	**80.27**	**89.61**	**55.35**	**77.90**	**88.35**
SAM Kirillov et al. [2023]	Base	64.60	81.61	89.99	49.17	70.75	82.71	40.82	64.80	78.74
	SNAP	**75.58**	**87.40**	**92.81**	**60.09**	**77.53**	**86.58**	**52.27**	**72.62**	**83.64**
MAE He et al. [2022]	Base	17.71	50.33	76.96	13.50	42.67	70.15	11.01	35.31	60.81
	SNAP	**75.34**	**89.02**	**94.17**	**59.64**	**78.19**	**87.21**	**50.67**	**72.81**	**84.16**

Table 5: Relative depth estimation on Hypersim and 7Scenes. AbsRel (↓) and δ (↑) for DINOv3 Siméoni et al. [2025] and SNAP. Best results are highlighted in **bold**.

	Hypersim		7Scenes	
Method	AbsRel ↓	δ_1 ↑	AbsRel ↓	δ_1 ↑
DINOv3 Siméoni et al. [2025]	0.7147	61.8738	1.0991	**52.8681**
SNAP	**0.7093**	**64.6815**	**0.9257**	52.0218

Table 6: ImageNet k-NN classification with frozen representations. Top-1 and Top-5 accuracy (%) are reported for the original teacher representations and their corresponding SNAP representations. Best results are highlighted in **bold**.

Backbone	Base		SNAP	
	Top-1 ↑	Top-5 ↑	Top-1 ↑	Top-5 ↑
DINOv3 Siméoni et al. [2025]	68.45	87.38	**68.81**	**87.41**
SAM Kirillov et al. [2023]	18.06	34.45	**21.46**	**38.28**
MAE He et al. [2022]	**30.78**	**49.59**	28.64	46.98

Table 7: ADE20K semantic segmentation with frozen representations. Linear-probe performance is reported in terms of mean Intersection-over-Union (mIoU) and pixel accuracy (%). Best results are highlighted in **bold**.

Representation	mIoU ↑	Pixel Acc. ↑
DINOv3 Siméoni et al. [2025]	**51.92**	**87.27**
SNAP	49.87	84.31

Relative Depth Estimation. – We further compare SNAP against its DINOv3 Siméoni et al. [2025] teacher in Table 5. SNAP outperforms its teacher on Hypersim, maintaining more accurate relative depth along with a better δ_1. This pattern holds in 7Scenes where SNAP displays a 15.8% improvement in accuracy in depth estimation while displaying a comparable δ_1. The improvements demonstrate that SNAP exceeds its teacher at dense prediction tasks as well as point correspondence.

Single-View Semantics. – We additionally evaluate whether the multi-view transformation preserves single-view semantic information. As shown in Table 6, the DINOv3 Siméoni et al. [2025] variant improves slightly on ImageNet Deng et al. [2009] k-NN, from 68.45 to 68.81 Top-1 accuracy. The SAM Kirillov et al. [2023] variant also improves, from 18.06 to 21.46 Top-1 accuracy, while the MAE He et al. [2022] variant decreases from 30.78 to 28.64. On ADE20K Zhou et al. [2017] linear-probe segmentation, DINOv3 Siméoni et al. [2025] obtains 51.92 mIoU and 87.27 pixel accuracy, compared with 49.87 mIoU and 84.31 pixel accuracy for SNAP (Table 7). These results show that SNAP largely retains the semantic discriminability of the teacher while deliberately shifting the representation toward multi-view geometric transfer.

D.2 Comparison with V-JEPA

SNAP and V-JEPA share the use of feature-space prediction rather than direct pixel reconstruction, but they differ fundamentally in the type of representation they are designed to learn. V-JEPA Bardes et al. [2024] is a video representation learner that predicts latent features across video observations, while V-JEPA 2.1 Mur-Labadia et al. [2026] extends this paradigm toward dense video features. In contrast, SNAP is explicitly a multi-view encoder: it learns a shared scene representation from multiple posed images and trains that representation to predict features at arbitrary target viewpoints.

Table 8: Point correspondence compared with V-JEPA 2.1. PCK (%) at thresholds 0.1, 0.2 and 0.3 for DINOv3 Siméoni et al. [2025] (ViT-B), SNAP (ViT-B), and V-JEPA 2.1-L Mur-Labadia et al. [2026], using global soft-argmax matching ($\tau = 0.02$). Values are mean ± CI. Hypersim is held out from SNAP's training. Best results are highlighted in **bold**.

	RealEstate10K			DL3DV			Hypersim		
Method	@0.1	@0.2	@0.3	@0.1	@0.2	@0.3	@0.1	@0.2	@0.3
DINOv3 Siméoni et al. [2025]	45.83 ±0.80	80.13 ±0.74	93.71 ±0.47	28.00 ±0.65	62.53 ±0.87	84.45 ±0.67	29.29 ±0.73	63.67 ±0.95	84.22 ±0.79
V-JEPA 2.1-L Mur-Labadia et al. [2026]	35.85 ±0.66	67.05 ±0.77	85.05 ±0.62	24.55 ±0.55	54.94 ±0.76	77.27 ±0.67	21.45 ±0.52	50.21 ±0.80	72.44 ±0.79
SNAP	**78.21** ±1.06	**91.32** ±0.78	**95.62** ±0.59	**58.74** ±1.14	**80.27** ±1.02	**89.61** ±0.81	**55.35** ±0.98	**77.90** ±0.95	**88.35** ±0.80

Thus, the comparison separates the benefit of feature-space prediction from the additional geometric constraint introduced by explicit multi-view representation learning.

Hypersim, RealEstate10K and DL3DV. – We evaluate a V-JEPA 2.1 ViT-L model using the same frozen-feature point-correspondence protocol as SNAP, i.e., global soft-argmax matching with a low temperature. As shown in Table 8, SNAP outperforms both V-JEPA 2.1-L and DINOv3 on RealEstate10K, DL3DV and Hypersim at every threshold, with the largest margins at tight thresholds.

On Hypersim, SNAP achieves 55.35, 77.90 and 88.35 PCK at thresholds of 0.1, 0.2 and 0.3, respectively, compared with 21.45, 50.21 and 72.44 for V-JEPA 2.1-L. DINOv3 obtains 29.29, 63.67 and 84.22. On RealEstate10K, SNAP achieves 78.21, 91.32 and 95.62, compared with 35.85, 67.05 and 85.05 for V-JEPA 2.1-L and 45.83, 80.13 and 93.71 for DINOv3. DL3DV shows the same pattern (PCK@0.1: 58.74 for SNAP vs. 24.55 and 28.00). The advantage narrows at PCK@0.3, where all methods approach saturation; on RealEstate10K the SNAP–DINOv3 gap there is under 2 points.

The consistent advantage across all three benchmarks indicates that feature-space prediction alone is insufficient to obtain SNAP's level of geometric transfer. The explicit multi-view constraint requires the representation to remain useful across viewpoint changes, rather than only across temporally related video observations.

Overall, the comparison highlights a fundamental distinction between video/image representation learning and explicit multi-view representation learning. Although both V-JEPA Bardes et al. [2024] and SNAP operate in feature space, their training constraints are different. V-JEPA learns representations through latent prediction between video observations, whereas SNAP learns a shared scene representation from multiple posed images and requires that representation to support feature prediction across viewpoints. The stronger correspondence performance of SNAP across Hypersim, RealEstate10K, and DL3DV indicates that this explicit multi-view geometric constraint provides a distinct advantage beyond feature-space prediction alone.

E Inference Speed

We evaluate the inference efficiency of SNAP against Muskie Li et al. [2025b] and VGGT Wang et al. [2025a] to characterize the computational overhead of our architecture. As shown in Table 9, SNAP demonstrates superior scalability and memory efficiency as the number of input views (V) increases.

Memory and Parameter Efficiency. – SNAP is significantly more memory-efficient than both baselines. At a high-resolution regime of $V = 100$, SNAP requires only **1.03 GB** of peak GPU memory, representing a **40.5% reduction** compared to Muskie (1.73 GB) and a **69.8% reduction** compared to VGGT (3.41 GB). This efficiency is underpinned by SNAP's compact parameterization; with only **233.38M** parameters including the DINOv3 backbone, SNAP is approximately $1.3\times$ smaller than Muskie and $3.9\times$ smaller than the geometry-grounded VGGT.

Temporal Performance. – While SNAP consistently outperforms VGGT, achieving a **2.9**$\times$ speedup at $V = 100$, it is roughly **36% slower** than Muskie in the high-view regime (0.245s vs. 0.180s). We attribute this latency gap to Muskie's architecture, which utilizes fewer multi-view self-attention blocks compared to our design. However, for practical deployment, SNAP offers a more accessible memory profile: at $V = 100$, SNAP requires only **1.03 GB** of VRAM, whereas Muskie consumes 67.9% more memory. For practitioners working with limited GPU memory or ultra-dense view counts, SNAP provides a viable alternative that prioritizes a smaller memory footprint over raw inference throughput. Notably, SNAP provides a different tradeoff of performance, speed, and memory.

Table 9: Encoder forward-pass efficiency vs. number of input views. For each encoder we report forward-pass time (milliseconds), peak GPU memory, and number of parameters. All encoders are benchmarked under identical conditions: bfloat16, single forward call per measurement, and we utilized an NVIDIA RTX PRO 6000 Blackwell GPU.

Method	Metric	V=1	V=2	V=4	V=8	V=10	V=20	V=50	V=100
Muskie Li et al. [2025b]	Time (ms) ↓	**12**	**13**	**13**	**16**	**16**	**25**	**67**	**180**
	Mem (GB) ↓	0.59	0.61	0.64	0.67	0.70	0.81	1.16	1.73
	Params (M) ↓	303.10	303.10	303.10	303.10	303.10	303.10	303.10	303.10
VGGT Wang et al. [2025a]	Time (ms) ↓	60	55	41	71	89	123	294	712
	Mem (GB) ↓	2.18	2.20	2.22	2.27	2.29	2.42	2.79	3.41
	Params (M) ↓	909.11	909.11	909.11	909.11	909.11	909.11	909.11	909.11
SNAP	Time (ms) ↓	20	23	23	22	22	30	85	245
	Mem (GB) ↓	**0.45**	**0.46**	**0.49**	**0.49**	**0.51**	**0.57**	**0.74**	**1.03**
	Params (M) ↓	233.38	233.38	233.38	233.38	233.38	233.38	233.38	233.38

F Qualitative Depth Estimation Results

We provide qualitative relative depth estimation results from our trained probe (see Figure 7). Given a sparse collection of four input images, we use a frozen backbone with trained DPT layers to predict relative depth. Overall, SNAP produces coherent depth predictions that are comparable to those of LagerNVS (*geometry pretrained*). In contrast, LVSM exhibits noisier predictions and less consistent depth structure across the examples.

The comparison with LagerNVS is not resolution-matched: LVSM and SNAP are decoded at 256px, whereas LagerNVS is decoded at 518px. The higher decoding resolution of LagerNVS likely contributes to its sharper performance on finegrained edges.

G Landscape Plots

We expand the sweep methodology from Section 4.2 to evaluate a broader set of metrics: mAP/MRR (reflecting bag-of-words correspondence retrieval), cycle consistency, and variance ratio (the ratio of within-scene to across-scene variance, indicating how well features cluster by scene) (Figure 8). Interestingly, we observe a distinct trade-off. While pairing mid-layer inputs with last-layer prediction targets yields higher raw correspondence, this configuration results in lower cycle consistency compared to initializing with the last layer of DINO. This performance valley for mid-layer inputs is consistently prominent across mAP/MRR, cycle consistency, and the variance ratio. Ultimately, this highlights that optimizing for raw local correspondence does not necessarily yield the most structurally robust, 3D-grounded representations, justifying our use of deeper semantic layers to enforce global multi-view consistency.

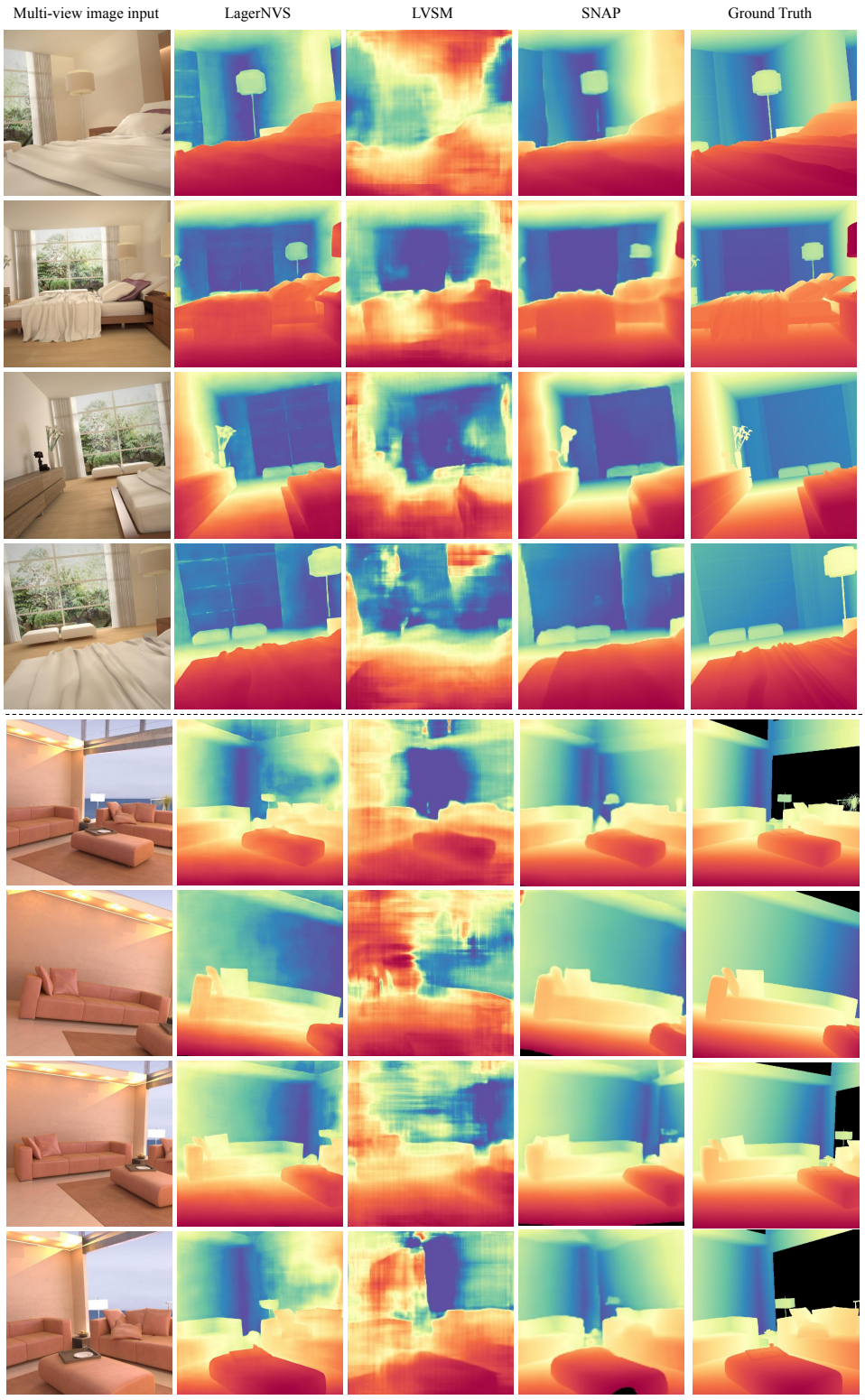

Figure 7: Qualitative Depth Evaluation: Hypersim relative depth predictions from a sparse collection of four input images using a frozen backbone with trained DPT layers. SNAP features yield depth predictions comparable to LagerNVS, despite using a lower decoding resolution (256px vs. 518px for LagerNVS). The higher resolution of LagerNVS likely contributes to its stronger predictions on thin structures.

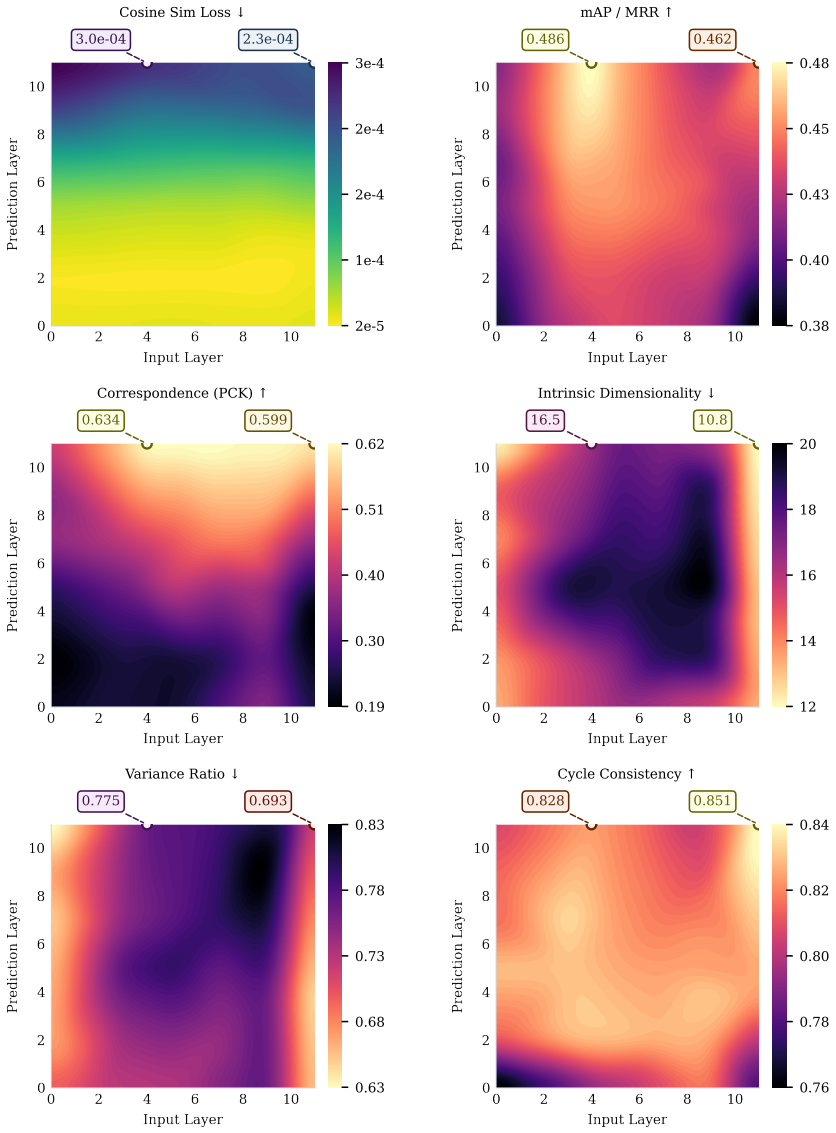

Figure 8: Extended Input vs. Prediction Layer Landscape. We observe a fundamental trade-off depending on the chosen input layer (yellow indicates superior performance across all metrics). While mid-layer inputs (e.g., Layer 4) optimize local discriminability (PCK, mAP/MRR), they create a distinct performance valley in holistic metrics. To maximize cycle consistency and scene clustering (Variance Ratio), initializing and predicting with the deepest layers (Layer 11) proves to demonstrate a robust optimal operating point.

H Qualitative Feature Visualizations

To qualitatively assess what these models learn, we visualize their high-dimensional feature maps projected into RGB space. As illustrated in Figure 9, standard 2D priors like DINO group scenes into flat, semantic regions but lack fine-grained 3D geometric structure. In contrast, SNAP clearly learns a representation distinct from its DINOv3 teacher: it introduces smooth, continuous spatial gradients across physical surfaces (e.g., ground planes), indicating an emergent understanding of 3D geometry while preserving fine-grained, discriminative object boundaries. Alternative multi-view baselines fail to achieve this geometric abstraction. Muskie heavily encodes raw appearance and texture rather than underlying structure, resulting in representations that simply mimic the RGB input. Meanwhile, LagerNVS suffers from severe, high-frequency grid artifacts, disrupting the continuous spatial coherence required for robust multi-view reasoning along with feature anisotropy.

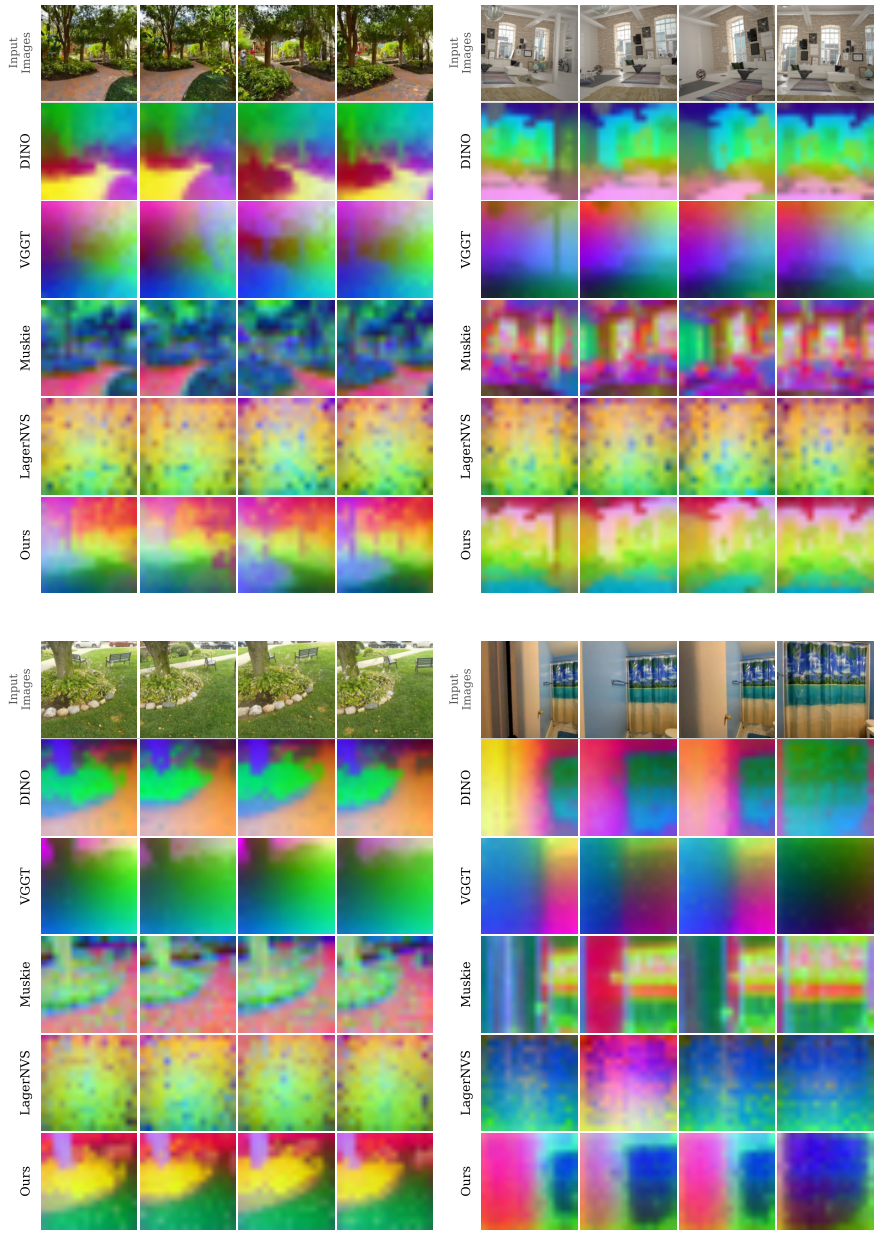

Figure 9: Feature map visualizations across vision encoders. High-dimensional latents are projected to RGB space using PCA to visualize spatial abstraction. While the 2D teacher (DINO) produces flat semantic regions, SNAP introduces clear spatial gradients that track underlying 3D geometry and depth, all while preserving fine-grained, discriminative object boundaries. Baselines struggle with this geometric abstraction: Muskie heavily encodes raw 2D appearance, and LagerNVS exhibits severe spatial grid artifacts.

NeurIPS Paper Checklist

1. **Claims**
   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?
   Answer: [Yes]
   Justification: Yes, the abstract and introduction accurately reflect the paper's scope by explicitly outlining its core contributions: the SNAP architecture, mechanistic analyses, and empirical evaluations across five geometric tasks. The subsequent sections directly fulfill these claims by thoroughly detailing the architecture, providing the promised ablations, and presenting the experimental results.
   Guidelines:
   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.
2. **Limitations**
   Question: Does the paper discuss the limitations of the work performed by the authors?
   Answer: [Yes]
   Justification: The paper discusses its limitations in the Conclusion (Section 6), specifically noting its reliance on posed video sequences and static, textured scenes. It also acknowledges that its findings are bounded by a fixed compute and data budget, and identifies the extension to dynamic video as a critical next step.
   Guidelines:
   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.
3. **Theory assumptions and proofs**
   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?
   Answer: [N/A]
   Justification: The paper does not include formal theoretical results, theorems, or proofs. Its contributions are primarily empirical and architectural.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The paper provides extensive details necessary for reproduction. Appendix B details the pretraining setup, datasets, optimization schedules, and architectural details for the geometric downstream task heads. Appendix C provides full hyperparameter and policy details for the robot manipulation experiments.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: While the paper uses publicly available datasets, including RealEstate 10K, DL3DV-10K, Co3Dv2, Hypersim, and 7Scenes, the proprietary code for the SNAP architecture and experimental pipelines is currently undergoing internal review for future release. However, comprehensive implementation details, architectural configurations, and optimization hyperparameters are provided in Appendices B, C, and D to support independent reproduction.

   Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**
   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?
   Answer: [Yes]
   Justification: The paper provides comprehensive experimental details across several sections. Section 3.4 defines the reconstruction objective, and Section 5 describes the evaluation protocol for downstream tasks. Specifics regarding data splits, the AdamW optimizer, and learning rate schedules are detailed in Appendix B and Appendix C.
   Guidelines:
   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.
7. **Experiment statistical significance**
   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?
   Answer: [No]
   Justification: The paper reports mean performance metrics across multiple datasets and tasks to establish empirical trends. Due to the high computational cost of pretraining the multi-view backbones, results are reported for single training runs rather than across multiple seeds with error bars.
   Guidelines:
   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**
   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?
   Answer: [Yes]
   Justification: The paper specifies that training was conducted over 40,000 steps using Distributed Data Parallel (DDP) across multiple GPU classes ranging from NVIDIA L40S to NVIDIA H200. Furthermore, Appendix E provides a detailed efficiency analysis, including peak GPU memory usage (GB) and forward-pass time (seconds) for various view counts, specifically noting the use of an NVIDIA RTX PRO 6000 Blackwell GPU for benchmarking.
   Guidelines:
   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).
9. **Code of ethics**
   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?
   Answer: [Yes]
   Justification: The research adheres to the NeurIPS Code of Ethics. The study utilizes publicly available datasets for self-supervised learning, does not involve human subjects or sensitive personal data, and aims to improve the transparency and efficiency of geometric representation learning.
   Guidelines:
   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).
10. **Broader impacts**
    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?
    Answer: [No]
    Justification: The paper focuses on foundational research regarding the architectural pathologies of self-supervised Novel View Synthesis and their impact on geometric representation learning. While the work demonstrates applications in robot manipulation, it does not explicitly discuss broader societal impacts, as the primary contributions are technical and algorithmic in nature.
    Guidelines:
    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.

- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The SNAP architecture is a 3D geometric backbone and does not possess generative capabilities for sensitive content (e.g., human faces, speech, or text) that are typically associated with high risks of misuse like deepfakes or disinformation. Furthermore, the datasets used are standard, publicly available benchmarks for 3D computer vision research.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The paper cites the original publications for all datasets and models used, including DINOv3, RoMa v2, and the RealEstate 10K, DL3DV-10K, and Co3Dv2 datasets. These assets are publicly available for academic research; for instance, DL3DV-10K and Co3Dv2 are typically distributed under Creative Commons (CC-BY) licenses, and 7-Scenes is provided under a Microsoft Research License Agreement (MSR-LA).

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The paper introduces a new model framework, SNAP, which is thoroughly documented within the manuscript.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.

- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**
    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?
    Answer: [N/A]
    Justification: The research conducted in this paper does not involve crowdsourcing experiments or research with human subjects.
    Guidelines:
    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.
15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**
    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?
    Answer: [N/A]
    Justification: This research does not involve human subjects, clinical trials, or the collection of new data from people. The study is limited to the use of publicly available, anonymized video and robotics datasets for self-supervised geometric representation learning.
    Guidelines:
    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.
16. **Declaration of LLM usage**
    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.
    Answer: [N/A]
    Justification: The core methodology of this research—specifically the SNAP architecture, the mechanistic analyses of decoders, and the latent reconstruction objectives—does not involve the use of Large Language Models (LLMs) . The models used are strictly vision-based transformers and encoders.
    Guidelines:
    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.