Resolving Representation Ambiguity in Feedforward Novel View Synthesis Transformer via Semantic-Spatial Decoupling

Yihang Wu[1,2]* **Yihang Sun**[3]* **Shaofeng Zhang**[4]

Zuxuan Wu[1,2] **Junchi Yan**[3] **Xiaosong Jia**[1,2]✉ **Yu-Gang Jiang**[1,2]

[1]Institute of Trustworthy Embodied Artificial Intelligence (TEAI), Fudan University
[2]Shanghai Key Laboratory of Multimodal Embodied AI
[3]Sch. of Artificial Intelligence & Sch. of Computer Science, Shanghai Jiao Tong University
[4]University of Science and Technology of China
* Equal Contributions ✉ Corresponding Authors

Abstract

Transformer-based models have advanced feedforward novel view synthesis (NVS). Current architectures such as GS-LRM and LVSM mix semantic information (e.g., RGB) and spatial information (e.g., Plücker rays) into a shared feature space. Since Plücker rays naturally carry lattice-like spatial structure, these designs can make the spatial bias interfere with appearance representation and degrade rendering fidelity. To this end, we propose to **decouple the representation of feedforward NVS transformers into separate semantic and spatial tokens**. The decoupled design keeps semantic and spatial information explicit in their branches while preserving cross-branch interaction through shared attention routing. Built on this design, we introduce optional categorized supervision and bidirectional modulation: the former provides branch-specific training signals, while the latter improves interaction between the two branches. Notably, the base decoupled design introduces virtually zero additional inference latency due to its architectural design. The proposed designs achieve consistent improvements, demonstrating effectiveness across decoder-only and encoder-decoder feedforward NVS models.

Project page: https://hangzay.github.io/ssd_lvsm/

1 Introduction

Novel View Synthesis (NVS) has undergone a fundamental paradigm shift, moving from per-scene optimization [1, 2] to generalizable feedforward inference powered by Large Reconstruction Models (LRMs) [3]. Transformer-based architectures like GS-LRM [4] and LVSM [5] typically adopt a unified token approach, where semantic features (RGB patches) and spatial features (Plücker rays) are concatenated and projected into a shared latent space. These inputs serve different roles: RGB provides appearance evidence, while camera rays describe the viewing geometry associated with each patch. However, since Plücker-ray features carry lattice-like spatial structure, jointly propagating both types of information through shared feature transformations can make spatial bias interfere with

*This work is supported by the National Natural Science Foundation of China (Grant No. 62427819), the Science and Technology Commission of Shanghai Municipality (No. 24511103100), and the New Cornerstone Science Foundation through the XPLORER PRIZE.

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

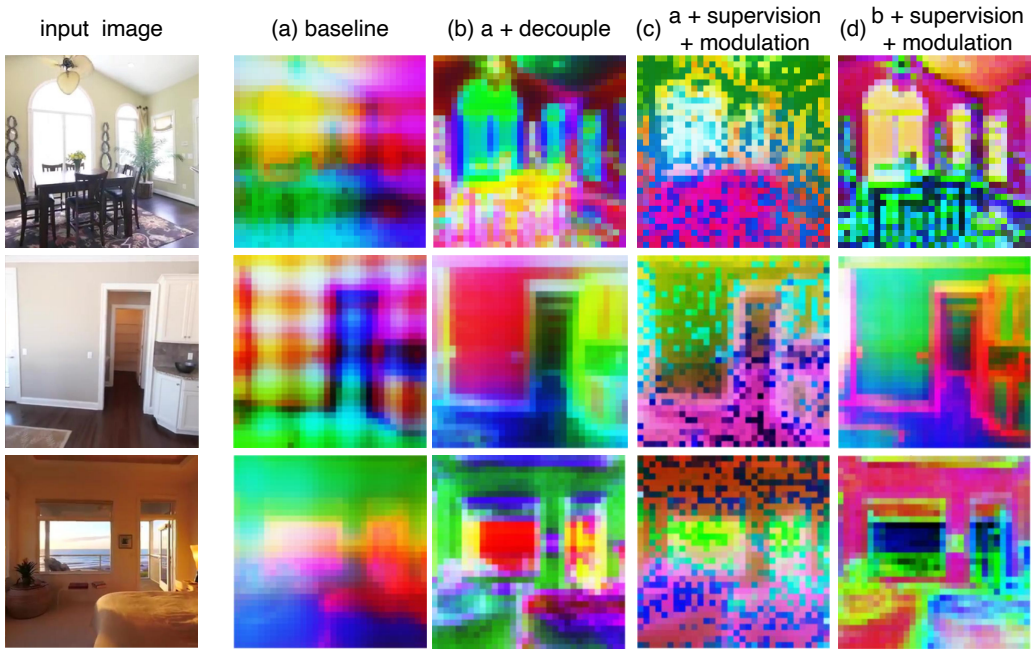


Figure 1: **Feature Map Comparison across Model Variants.** We visualize middle-layer LVSM features: (a) the LVSM baseline; (b) Decouple-only, with separate semantic and spatial branches and no categorized supervision or modulation; (c,d) supervision and modulation added to (a,b), respectively. Decoupling produces more structured representations and enables the auxiliary designs.

appearance representation. Additionally, we observe grid-like distortions in intermediate features (Fig. 1), resembling artifacts previously reported in vision Transformers [6]. This motivates us to examine how appearance and spatial conditioning interact within the token representation.

Inspired by register tokens [7] in ViTs, we first introduce registers to provide extra storage slots, but this control does not resolve the issue. The persistence of grid-like patterns when RGB inputs are removed further points to the role of spatial conditioning. In this work, we propose to **decouple semantic and spatial tokens**, redesigning the Transformer representation space to preserve semantic and spatial feature independence while facilitating cross-stream context exchange. As shown in Fig. 2, the two branches exhibit different feature patterns. The core architectural change is full-pipeline decoupled Transformer by **Independent-V Attention mechanism**: shared query-key routing coordinates the branches, while separate value projections, layer normalization, and feed-forward networks keep their feature updates branch-specific. Based on this decoupled representation, we further introduce optional categorized supervision and bidirectional modulation as auxiliary designs (Fig. 3). The supervision terms provide branch-specific training signals and are removed at inference, while modulation improves controlled cross-branch conditioning.

Notably, the base decoupled design incurs virtually zero additional inference latency while improving rendering quality over the entangled baseline. In our controlled RealEstate10K setting, optional supervision and modulation further improve performance, reaching a 1.1 dB PSNR gain with an 8% inference-time increase. Extensive experiments across decoder-only and encoder–decoder architectures, larger backbones, and scene- and object-level datasets validate the effectiveness of the proposed approach. The benefit also persists with raw raymaps, indicating that semantic–spatial decoupling is not specific to the Plücker parameterization.

2 Related Work

Feedforward Novel View Synthesis. NVS has progressed from image-based rendering [8] and per-scene neural optimization [1, 2] to feed-forward approaches. Early methods [9, 10, 11] conditioned NeRF networks on pixel-aligned features but retained explicit 3D inductive biases, whereas transformer-based LRMs [3, 4, 12] learn scalable reconstruction priors from data. Models without explicit 3D representations [5, 13, 14, 15] further simplify NVS into token-level image prediction.

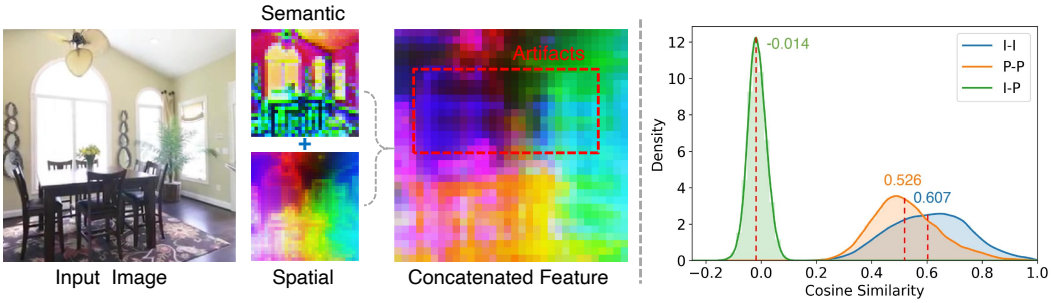


Figure 2: **Feature Coupling and Cosine Similarity.** Left: Semantic and spatial features are structured separately, while their concatenation exposes grid-like Plücker-related artifacts. Right: We sample vector pairs to estimate cosine similarity distributions for I–I (semantic), P–P (spatial), and I–P (cross-branch). Dashed lines indicate the mean of each group.

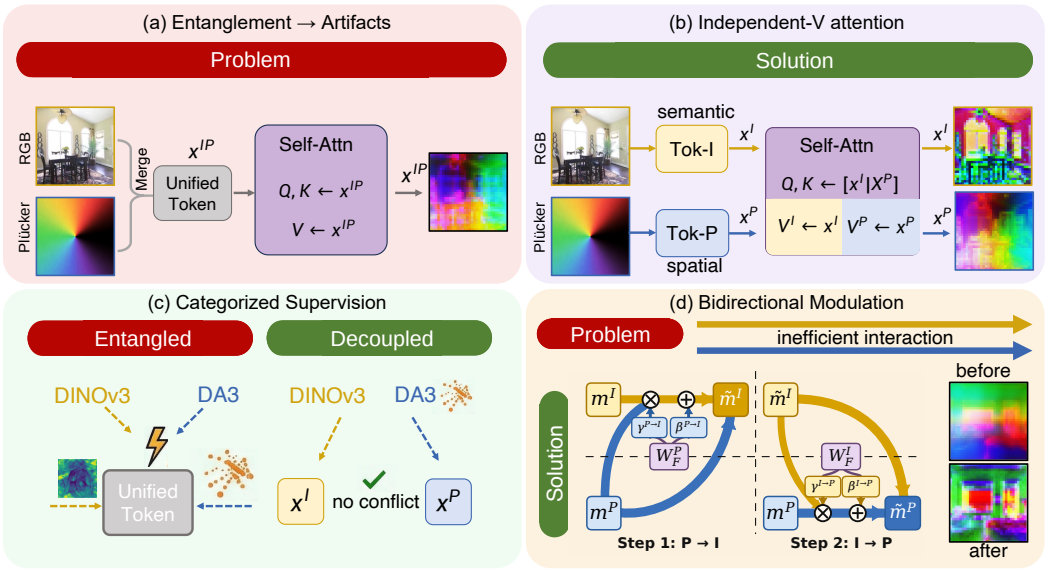


Figure 3: **Overview.** In this work, we identify that mixing RGB and Plücker-ray information in a shared feature space can make spatial bias interfere with appearance representation. We decouple the two information streams while preserving cross-branch interaction to enhance rendering.

Generation-based approaches such as Seva [16], GEN3C [17], and CAT3D [18] synthesize camera-controlled views, while Lyra and Lyra 2.0 [19, 20] use video priors to construct 3D scenes. LGM [21], TRELLIS [22], and Hunyuan3D 2.1 [23] focus on 3D content or asset creation; these settings differ in inputs and outputs from posed-image reconstruction. Despite these advances, recent feedforward NVS transformers commonly encode semantic appearance and spatial geometry within unified tokens. Our work addresses the resulting intra-token representation ambiguity, complementary to concurrent efforts on improved camera conditioning [24].

Token Dynamics and Feature Artifacts in Vision Transformers. ViTs [25] exhibit representational artifacts such as grid-like patterns linked to positional encoding [6] and high-norm outlier tokens associated with implicit memory storage [7]. Register tokens provide dedicated storage slots for such outlier behavior [7, 26], which is also related to attention sink phenomena in language models [27]. Specialized processing of class and patch tokens further improves local representations [28]. In NVS transformers, however, each patch token is also bound to a camera ray, so feature artifacts are coupled with semantic-spatial mixing rather than token dynamics alone. This motivates us to separate semantic and spatial representations instead of relying only on additional register tokens.

Representation Alignment. The DINO series [29, 30, 31] has shown that self-supervised ViTs learn useful visual representations. REPA [32] and its variants [33, 34] align hidden states with such features to improve generative training, while iREPA [35] emphasizes fine-grained spatial structure. Representation autoencoders [36] instead use pretrained visual features as diffusion latents. Applying this supervision to unified NVS tokens can mix semantic and geometric objectives. We therefore use iREPA with DINOv3 as branch-specific supervision for the decoupled semantic stream.

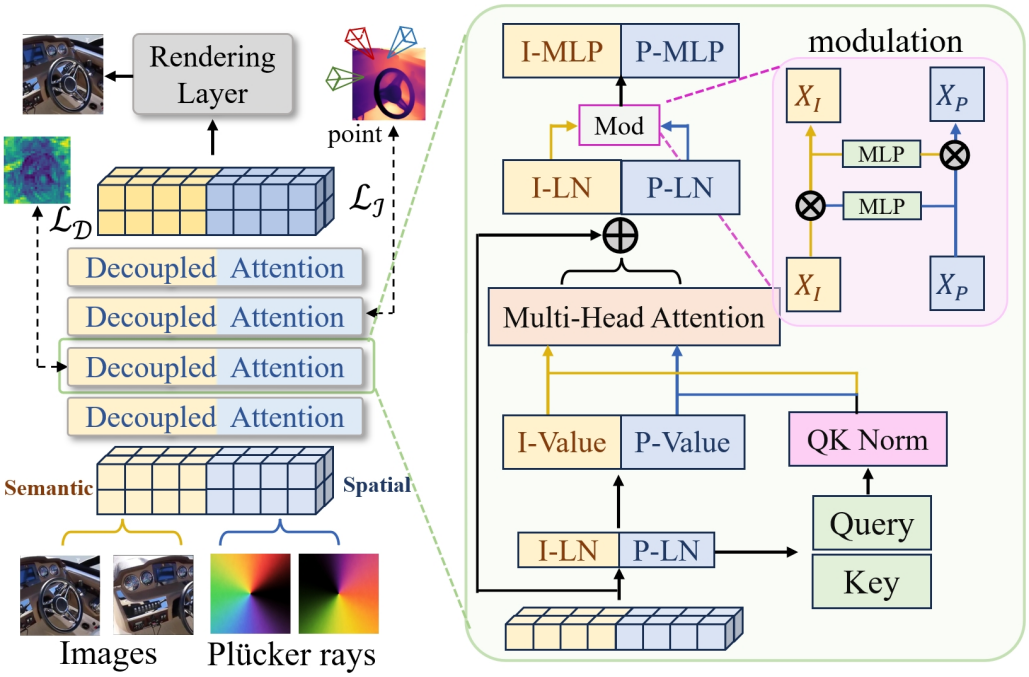


Figure 4: **Semantic–Spatial Decoupled Architecture for Feedforward NVS.** Semantic tokens (I) from RGB and spatial tokens (P) from Plücker rays pass through decoupled-attention blocks. Attention shares query–key interactions while using independent value projections to preserve heterogeneous representations. Bidirectional modulation further enables cross-stream conditioning.

Geometric Priors for 3D Understanding. Monocular and multi-view depth estimation has matured into a family of large-scale foundation models [37, 38, 39, 40, 41, 42]. DA3 [43] extends this line to multi-view consistent geometry prediction through depth and rays. When geometric supervision is enabled, we use its outputs to supervise the decoupled spatial stream during training, without adding an inference-time geometry module. LagerNVS [44] instead initializes its inference-time encoder from a geometry-pretrained reconstruction network.

3 Method

3.1 Preliminary

Given N posed input images $\{(\mathbf{I}_i, \mathbf{E}_i, \mathbf{K}_i)\}_{i=1}^{N}$ and M target poses $\{(\mathbf{E}_j, \mathbf{K}_j)\}_{j=1}^{M}$, feedforward NVS renders $\{\hat{\mathbf{I}}_j\}_{j=1}^{M}$ in a single forward pass, where $\mathbf{I}_i \in \mathbb{R}^{H\times W\times 3}$ is the input RGB image, $\mathbf{E}_i, \mathbf{K}_i \in \mathbb{R}^{4\times 4}$ are the extrinsic and intrinsic matrices, and $\hat{\mathbf{I}}_j \in \mathbb{R}^{H\times W\times 3}$ is the rendered output. In this work, we adopt LVSM [5] as the baseline. The camera viewpoints are encoded as Plücker ray maps [45] $\boldsymbol{\pi}_i \in \mathbb{R}^{6\times H\times W}$. Each input view is divided into $N_p = HW/p^2$ non-overlapping patches of size $p\times p$; the RGB and Plücker patch sequences, denoted $\mathbf{I}_i^{\mathbf{P}}$ and $\boldsymbol{\pi}_i^{\mathbf{P}}$, are *concatenated along the channel dimension* and projected into the D-dimensional token space, while target tokens are Plücker patches alone:

$$\mathbf{t}_i = \mathrm{MLP}_{\mathrm{in}}(\mathrm{concat}(\mathbf{I}_i^{\mathbf{P}}, \boldsymbol{\pi}_i^{\mathbf{P}})) \in \mathbb{R}^{N_p\times D}, \qquad \mathbf{q}_j = \mathrm{MLP}_{\mathrm{tgt}}(\boldsymbol{\pi}_j^{\mathbf{P}}) \in \mathbb{R}^{N_p\times D}. \tag{1}$$

All tokens are then concatenated and processed by L layers of full self-attention, after which target features are fed into the rendering head.

3.2 Decoupled Semantic and Spatial Tokens

Decoupled Tokenization. As shown in Fig. 4, we replace the modality-mixing input projection in LVSM with two separate tokenizers for semantic appearance and spatial geometry. For each input view, RGB patches $\mathbf{I}_i^{\mathbf{P}}$ and Plücker-ray patches $\boldsymbol{\pi}_i^{\mathbf{P}}$ are tokenized into a semantic branch (I-branch)

and a spatial branch (P-branch):

$$\mathbf{x}_i^I = \mathrm{Tok}_I(\mathbf{I}_i^{\mathbf{P}}) \in \mathbb{R}^{N_p \times \frac{D}{2}}, \qquad \mathbf{x}_i^P = \mathrm{Tok}_P(\boldsymbol{\pi}_i^{\mathbf{P}}) \in \mathbb{R}^{N_p \times \frac{D}{2}}. \tag{2}$$

For each target view, no RGB observation is available, so we initialize the target semantic branch with zeros and tokenize the target Plücker-ray patches into the spatial branch:

$$\mathbf{y}_j^I = \mathbf{0} \in \mathbb{R}^{N_p \times \frac{D}{2}}, \qquad \mathbf{y}_j^P = \mathrm{Tok}_P^{\mathrm{tgt}}(\boldsymbol{\pi}_j^{\mathbf{P}}) \in \mathbb{R}^{N_p \times \frac{D}{2}}. \tag{3}$$

The full input and target tokens are then formed by concatenating their I-branch and P-branch features. The total token dimension therefore remains the same as the original LVSM token, while the branch partition is preserved throughout the Transformer blocks.

Independent-V Attention. Each decoupled Transformer block keeps query-key (Q/K) routing on the full token, but uses branch-specific value and output projections. With branch-wise pre-normalized inputs $\bar{\mathbf{x}}^I$ and $\bar{\mathbf{x}}^P$, let $\bar{\mathbf{x}} = \mathrm{concat}(\bar{\mathbf{x}}^I, \bar{\mathbf{x}}^P)$. We compute

$$\mathbf{q} = \bar{\mathbf{x}}\mathbf{W}_q, \qquad \mathbf{k} = \bar{\mathbf{x}}\mathbf{W}_k, \qquad \mathbf{v}^I = \bar{\mathbf{x}}^I\mathbf{W}_v^I, \quad \mathbf{v}^P = \bar{\mathbf{x}}^P\mathbf{W}_v^P. \tag{4}$$

Here $\mathbf{W}_q$ and $\mathbf{W}_k$ project the full token to Q/K, while $\mathbf{W}_v^I$ and $\mathbf{W}_v^P$ project the two value streams separately. The shared attention map is

$$\mathbf{A} = \mathrm{softmax}\left(\frac{\mathbf{q}\mathbf{k}^\top}{\sqrt{d_h}}\right), \tag{5}$$

where d_h is the per-head Q/K dimension. The same attention map is applied to the two branch-specific value streams and followed by branch-specific output projections:

$$\mathbf{o} = \mathrm{concat}\big((\mathbf{A}\mathbf{v}^I)\mathbf{W}_o^I, (\mathbf{A}\mathbf{v}^P)\mathbf{W}_o^P\big). \tag{6}$$

where $\mathbf{W}_o^I$ and $\mathbf{W}_o^P$ are the output projections for the two branches. After attention, the I-branch and P-branch are processed by branch-specific layer normalization (LN) and half-dimensional feed-forward networks (FFNs), then concatenated back to a D-dimensional token. Thus, the block preserves full-token interaction through shared Q/K while keeping value updates and FFN transformations branch-specific. The qualitative feature progression in Fig. 1 visualizes the resulting change in intermediate representations, with detailed analysis provided in Sec. 4.4.

3.3 Optional Categorized Supervision

The decoupled representation enables optional categorized supervision for the I-branch and P-branch.

Semantic Alignment via iREPA with DINOv3. For the I-branch, we align intermediate semantic features with frozen DINOv3 features using an iREPA-style representation alignment [35]. Let $\mathbf{x}_{k_I}^I$ denote the I-branch feature at layer k_I, $f_{\mathrm{DINO}}(\mathbf{I})$ denote the DINOv3 teacher feature, $\mathrm{SN}(\cdot)$ denote spatial normalization, and h_ϕ denote the semantic projection head. The semantic supervision is

$$\mathcal{L}_{\mathrm{iREPA}} = \mathrm{SmoothL1}\big(h_\phi(\mathrm{SN}(\mathbf{x}_{k_I}^I)), f_{\mathrm{DINO}}(\mathbf{I})\big). \tag{7}$$

For a token feature $\mathbf{z}$, spatial normalization is defined as

$$\mathrm{SN}(\mathbf{z}) = \frac{\mathbf{z} - \gamma\mu(\mathbf{z})}{\sigma(\mathbf{z}) + \epsilon}, \tag{8}$$

where $\mu(\mathbf{z})$ and $\sigma(\mathbf{z})$ are computed over the token dimension, γ is a normalization coefficient, and ϵ is a small constant for numerical stability. Inspired by iREPA, this normalization emphasizes local feature variation before alignment. Both h_ϕ and the DINOv3 teacher are discarded at inference.

Geometric Consistency of the Spatial Branch. For the P-branch, we use DA3-derived geometry [43] to construct visible cross-view correspondences. Specifically, source-view 3D points are projected into target views with the aligned camera parameters; when multiple source points map to the same target location, z-buffering keeps the point with the smallest target-view depth and filters occluded matches. Let Ω_j be the resulting correspondence set for target view j, where $(i, u, v) \in \Omega_j$ matches source location u in view i to target location v in view j. We supervise the spatial branch with feature cosine consistency:

$$\mathcal{L}_{\mathrm{geo}} = \frac{1}{|\mathcal{J}|}\sum_{j\in\mathcal{J}}\frac{1}{|\Omega_j|}\sum_{(i,u,v)\in\Omega_j}\big(1 - \cos\big(\mathbf{x}_{k_P,i,u}^P, \mathbf{x}_{k_P,j,v}^P\big)\big), \tag{9}$$

where $\mathcal{J}$ denotes target views with valid correspondences, $\mathbf{x}_{k_P,i,u}^P$ and $\mathbf{x}_{k_P,j,v}^P$ denote P-branch features at layer k_P for the matched locations, and $\cos(\cdot,\cdot)$ denotes cosine similarity. The cached DA3 geometry is used to build correspondence supervision during training and is not used at inference.

3.4 Optional Bidirectional Feature Modulation

Bidirectional modulation is an optional refinement on top of the decoupled Transformer block. It uses cross-branch conditioning [46] after layer normalization and before the branch-specific FFNs.

Spatial → Semantic. Let $\mathbf{m}^I$ and $\mathbf{m}^P$ denote the layer-normalized attention outputs of the I and P branches respectively, i.e., the inputs to their corresponding FFNs. The spatial representation $\mathbf{m}^P$ generates scale and shift parameters that modulate the semantic branch:

$$(\boldsymbol{\gamma}^{P\to I}, \boldsymbol{\beta}^{P\to I}) = \mathbf{m}^P \mathbf{W}_{\mathrm{F}}^P + \mathbf{b}_{\mathrm{F}}^P, \qquad \tilde{\mathbf{m}}^I = \boldsymbol{\gamma}^{P\to I} \odot \mathbf{m}^I + \boldsymbol{\beta}^{P\to I}. \tag{10}$$

This step lets spatial cues such as view direction and ray geometry condition the semantic update without merging spatial channels into the semantic representation.

Semantic → Spatial. The modulated semantic representation $\tilde{\mathbf{m}}^I$ then generates the reverse affine parameters for the spatial branch:

$$(\boldsymbol{\gamma}^{I\to P}, \boldsymbol{\beta}^{I\to P}) = \tilde{\mathbf{m}}^I \mathbf{W}_{\mathrm{F}}^I + \mathbf{b}_{\mathrm{F}}^I, \qquad \tilde{\mathbf{m}}^P = \boldsymbol{\gamma}^{I\to P} \odot \mathbf{m}^P + \boldsymbol{\beta}^{I\to P}. \tag{11}$$

The branch-specific FFNs are then applied to $\tilde{\mathbf{m}}^I$ and $\tilde{\mathbf{m}}^P$ separately. Thus, modulation provides controlled bidirectional conditioning while preserving the decoupled representation.

Here $\mathbf{W}_{\mathrm{F}}^P$ and $\mathbf{W}_{\mathrm{F}}^I$ are learnable generator weights, $\mathbf{b}_{\mathrm{F}}^P$ and $\mathbf{b}_{\mathrm{F}}^I$ are their biases, and $\boldsymbol{\gamma}, \boldsymbol{\beta}$ are the predicted scale and shift parameters. We initialize the generators as an identity modulation, with zero weights, scale bias initialized to 1, and shift bias initialized to 0. Therefore, enabling modulation starts from the original decoupled block and learns cross-branch corrections when they are useful.

3.5 Training Objective

The base decoupled model is trained with the RGB reconstruction loss. When optional categorized supervision is enabled, the objective becomes

$$\mathcal{L} = \mathcal{L}_{\mathrm{RGB}} + \lambda_I \mathcal{L}_{\mathrm{iREPA}} + \lambda_P \mathcal{L}_{\mathrm{geo}}. \tag{12}$$

Here $\mathcal{L}_{\mathrm{iREPA}}$ and $\mathcal{L}_{\mathrm{geo}}$ constrain the I-branch and P-branch only during training. Implementation details and loss weights are provided in Appendix A.

4 Experiments

4.1 Datasets

Scene-Level. For scene-level, we use RealEstate10K [47], which aggregates over 80K video clips of diverse indoor and outdoor environments collected from YouTube. The dataset provides large-scale real-world scenes with varied viewpoints and camera motions. Train and test splits follow LVSM [5].

Object-Level. Object-level experiments use Objaverse [48] for training, with 730K objects each rendered at 32 viewpoints using the GS-LRM [4] pipeline. Evaluation is performed on the Objaverse test split. At test time, each object is presented with 4 structured input views, and performance is measured on 10 held-out target views sampled at random, following the protocol of LVSM [5].

4.2 Implementation Details

Model Configurations. Our main experiments use a 12-layer LVSM Decoder-Only [5] and an LVSM Encoder-Decoder with a 12-layer encoder and 12-layer decoder; both use a hidden dimension of 768. We also test a 24-layer Decoder-Only backbone.

Controlled Reimplementation. Our LVSM baselines and variants share the codebase, data splits, view sampling, resolution, and training budget. We train for 50K iterations, using 2 input and 6 target views at 256×256 on RealEstate10K, and 4 input and 8 target views on Objaverse. External reconstruction methods are evaluated on the same benchmarks but retain different architectures, camera encodings, and pretrained priors. Optimizer, loss, and supervision details are collected in Appendix A. We report PSNR, SSIM [49], and LPIPS [50] for all quantitative comparisons.

Table 1: **Reconstruction and camera-encoding comparisons.** Full: decoupling with optional supervision and modulation. Params in millions; Enc-Dec: encoder–decoder; 24L: 24-layer decoder-only; P/R: Plücker rays/raw raymaps; PRoPE adds projective camera conditioning.

(a) RealEstate10K reconstruction

Method	Params	PSNR↑	SSIM↑	LPIPS↓
LagerNVS-DA3-S/4 [44]	55.3	24.91	0.801	0.146
MVSplat [51]	11.9	25.48	0.851	0.143
PRoPE [52]	85.8	26.79	0.851	0.136
DepthSplat [53]	120.7	26.43	0.865	0.127
pixelSplat [54]	119.0	25.83	0.859	0.138
LVSM [5]	85.8	26.10	0.839	0.144
LVSM Decouple	50.1	26.70	0.851	0.138
Ours (full)	57.2	*27.21*	**0.869**	*0.125*
LVSM (24L)	—	26.18	0.8420	0.1406
Ours (24L)	—	**27.37**	*0.8671*	**0.1249**
LVSM (Enc-Dec)	—	24.06	0.775	0.206
Decouple (Enc-Dec)	—	24.84	0.789	0.170
Ours (Enc-Dec, full)	—	**25.31**	**0.806**	**0.154**

(b) Objaverse

Method	Params	PSNR↑	SSIM↑	LPIPS↓
LGM [21]	56.980	22.61	0.834	0.193
LVSM [5]	85.841	23.75	0.864	0.150
LVSM Decouple	50.083	*25.96*	*0.880*	*0.112*
Ours (full)	57.188	**26.46**	**0.899**	**0.101**

(c) Camera encoding (RealEstate10K)

Method	Params	PSNR↑	SSIM↑	LPIPS↓
PRoPE (P) [52]	85.8	26.79	0.851	*0.136*
LVSM (P)	85.8	26.10	0.839	0.144
LVSM (R)	85.8	25.78	0.829	0.149
Decouple (R)	50.1	*26.82*	*0.853*	0.138
Decouple (P)	50.1	26.70	0.851	0.138
Ours (P)	57.2	**27.21**	**0.869**	**0.125**

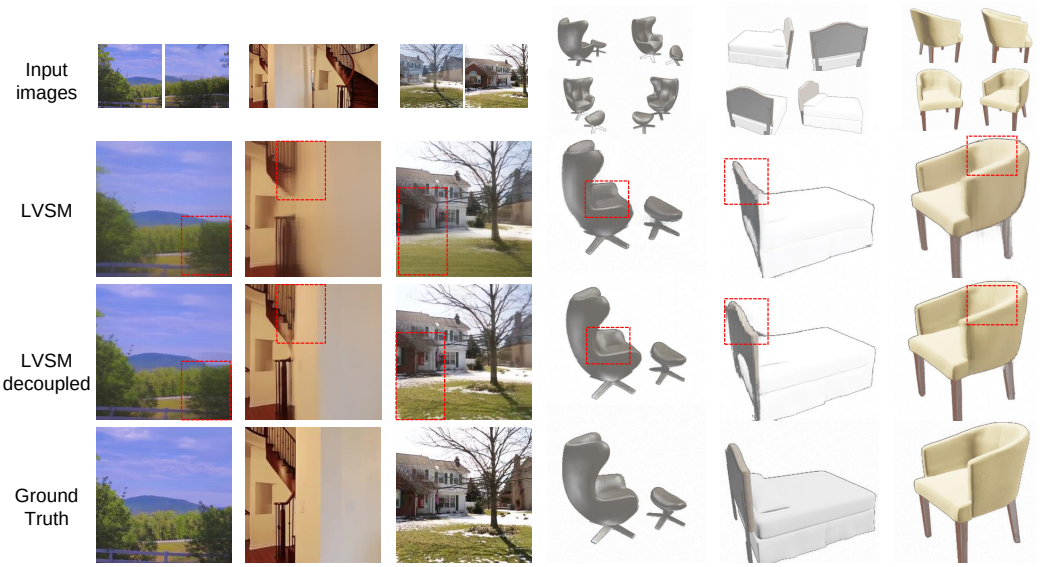


Figure 5: **Novel View Synthesis Visual Comparison.** Our decoupled model produces more coherent structures and sharper details.

4.3 Comparison with Baselines

Table 1 compares our variants with recent reconstruction methods under common training and evaluation protocols. We match data splits, view sampling, resolution, batch size, training budget, and evaluation views. Decoupling improves both LVSM Decoder-Only and Encoder–Decoder backbones over their corresponding baselines; optional supervision and modulation bring further gains in the full variants. Improvements also hold with the 24-layer Decoder-Only backbone and raw raymaps. Fig. 5 shows the visual improvement; Detailed comparison settings are provided in Appendix A. Additional qualitative comparisons on both datasets and RealEstate10K rendering videos are provided online.

4.4 Ablation Study

Table. 2 summarizes the component and control ablations. Unless otherwise specified, feature visualizations are taken from middle Transformer layers.

Decoupling and Categorized Supervision. In the entangled baseline, RGB and Plücker information share the same feature channels, so semantic alignment and geometric consistency are imposed on a mixed representation and do not improve rendering quality. After decoupling, the two objectives become well-defined: iREPA constrains the I-branch, while geometric correspondence constrains the

Table 2: **Component Ablation.** Reg: register tokens. †Training diverged: result unavailable.

Configuration	PSNR↑	SSIM↑	LPIPS↓
Decoupling and Categorized Supervision			
Baseline	**26.10**	**0.839**	**0.144**
Baseline + Sup	26.02	0.831	0.148
Decouple	**26.70**	**0.851**	**0.138**
Decouple + Sup	**26.91**	**0.857**	**0.134**
Register Control			
Baseline + Reg	26.04	0.835	0.146
Baseline	**26.10**	**0.839**	**0.144**
Shared vs. Independent Q/K			
Independent-Q/K	25.90	0.832	0.148
Shared-Q/K	**26.70**	**0.851**	**0.138**
Modulation Placement			
Pre-Attn-LN†	—	—	—
Post-FFN	26.95	0.858	0.132
Post-Attn-LN/Pre-FFN	**27.06**	**0.860**	**0.131**

Configuration	PSNR↑	SSIM↑	LPIPS↓
Effect of Bidirectional Modulation			
Baseline	**26.10**	**0.839**	**0.144**
Baseline + Mod	25.93	0.829	0.150
Decouple	**26.70**	**0.851**	**0.138**
Decouple + Mod	**27.06**	**0.860**	**0.131**
Cross-Branch Modulation Direction			
I→P only	26.80	0.853	0.136
P→I only	26.91	0.856	0.133
Decouple + Mod	**27.06**	**0.860**	**0.131**
LayerNorm Decoupling			
Joint-LN	26.01	0.830	0.149
Decouple	**26.70**	**0.851**	**0.138**
Full Configuration			
Decouple + Sup + Mod	**27.21**	**0.869**	**0.125**

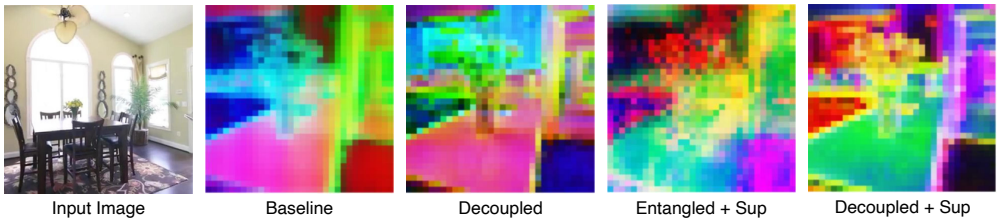


Figure 6: **Supervision Difference.** Categorized supervision benefits the decoupled model but harms the entangled baseline.

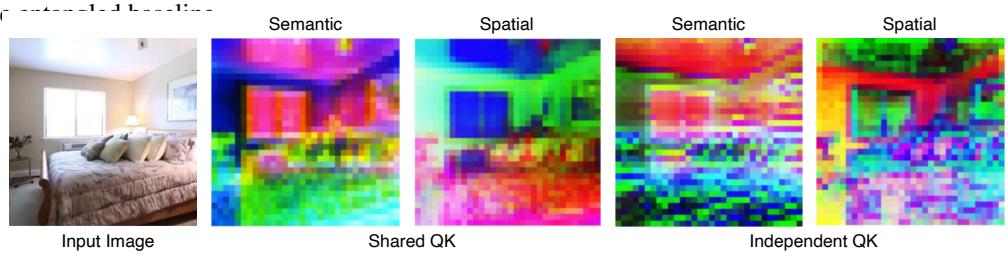


Figure 7: **Shared vs. Independent QK.** Shared Q/K with independent V yields better performance.

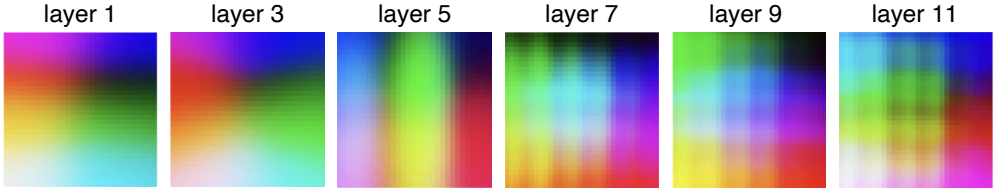
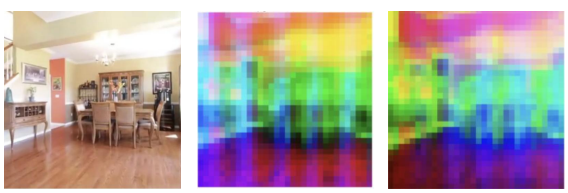


Figure 8: **Plücker-Only Inference Control.** Artifacts originate from the spatial ray representation. P-branch. Fig. 6 shows clearer semantic and geometric structures, confirming that supervision is secondary to, and enabled by, the decoupled representation.

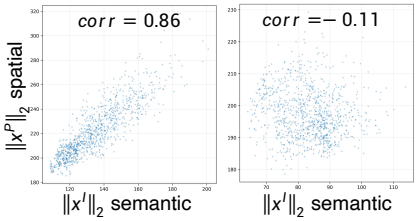
Register Control. We add register tokens to the entangled baseline under the same setting to test whether the grid-like artifacts can be handled by providing extra storage slots. This control does not improve rendering quality, and the grid-like artifacts remain visible in the feature maps (Fig. 9a). This suggests that the artifacts are not fully explained by the lack of extra storage slots. GAS measurements and controlled feature interventions examine grid structure and its effect on rendering (Appendix C).

Plücker-Only Inference Control. To further localize the source of the grid-like artifacts, we run the decoupled model at inference time with all input RGB values set to zero while keeping the Plücker rays unchanged. As shown in Fig. 8, the intermediate features still exhibit clear network-like grid patterns. Since image appearance has been removed in this control, the remaining artifacts provide evidence that the grid-like patterns are induced by the spatial Plücker-ray representation.

Alternative Camera Encodings. We replace Plücker rays $(o \times d, d)$ with raw raymaps (o, d), where o and d denote the ray origin and direction, while keeping the LVSM architecture and training protocol



(a) **Register Tokens Ablation.** Register tokens alone do not remove the grid-like artifacts.



(b) **LayerNorm Ablation.** Entangled representations exhibit strong norm correlation.



Figure 10: **Entangled vs. Decoupled Modulation and Bidirectional Modulation.** Left: Modulation improves performance in the decoupled structure but degrades performance in the entangled architecture. Right: Bidirectional modulation yields more structured representations.

otherwise unchanged. Panel (c) of Table 1 shows that decoupling improves all three rendering metrics with raw raymaps, indicating that its benefit is not specific to the Plücker parameterization. PRoPE [52] incorporates projective camera geometry into attention, whereas our design addresses the entanglement caused by jointly propagating semantic and spatial features. This distinction positions decoupling as a representation-level design complementary to camera conditioning.

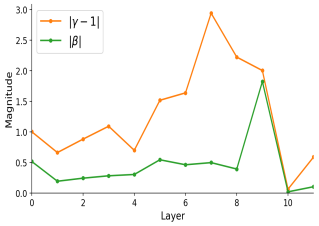
Shared vs. Independent Q/K Projection. We test whether decoupling should also separate query-key (Q/K) routing. In our default block, Q/K are projected from the full token, while values are projected independently from the I-branch and P-branch. The Independent-Q/K variant instead computes Q/K/V separately for the two branches, producing branch-local attention maps, but lowers performance. This suggests that shared Q/K remains useful for spatial coordination, while independent V preserves branch-specific updates (Fig. 7).

Effect of Bidirectional Modulation. We compare modulation in entangled and decoupled representations (Fig. 10). In the entangled model, affine modulation is generated from a mixed feature space, so the scale and shift parameters cannot distinguish whether a response should affect appearance or geometry. In the decoupled model, modulation operates between specialized branches, allowing spatial tokens to condition appearance tokens and semantic tokens to feed back content. Fig. 11a shows modulation in middle-to-late layers, with scale and shift magnitudes peaking at blocks.

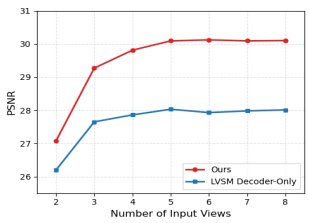
Cross-Branch Modulation Direction. We further ablate the direction of cross-branch conditioning. P→I modulation uses spatial tokens to guide appearance rendering, while I→P modulation uses updated semantic tokens to refine spatial representations. P→I alone is stronger than I→P alone, which matches the NVS pipeline where target-view appearance is rendered from viewing geometry. Bidirectional modulation performs best because the second direction provides complementary feedback after the semantic branch has been spatially conditioned. The feature visualizations in Fig. 10 show that this bidirectional design yields more structured representations.

LayerNorm Decoupling. We compare branch-wise LayerNorm with Joint-LN. Joint-LN normalizes the concatenated token using shared statistics, which couples the I-branch and P-branch even after token decoupling. Branch-wise LayerNorm instead normalizes each branch with its own statistics, preserving semantic–spatial separation. As shown in Fig. 9b, Joint-LN produces stronger cross-branch norm correlation, while Separate-LN reduces this coupling and improves reconstruction quality.

Modulation Placement. We study where to insert bidirectional modulation in each block. Applying modulation before attention LayerNorm perturbs the residual stream and leads to unstable training. Applying it after the FFN is stable but weaker, because the nonlinear transformation has already been computed before cross-branch conditioning is introduced. Our final implementation starts from branch-wise normalized features and applies modulation before the branch-specific FFNs; the reverse generator receives the modulated semantic feature, as defined in Sec. 3.4.



(a) **Intensity of Modulation.**



(b) **Input-View Generalization.**

Model	Latency(ms) ↓
Baseline	12.2
Decouple	12.3
Decouple + Mod	13.2

(c) **Inference Latency.**

Table 3: **Wide-angle robustness.** RE10K: RealEstate10K. ΔPSNR: Ours−LVSM, in dB. Bold indicates the best metric within each setting.

(a) Multi-input trajectory span

Dataset	Inputs	Span	Model	PSNR↑	SSIM↑	LPIPS↓	ΔPSNR
RE10K	3	90°	LVSM	20.572	0.6613	0.2876	—
			Ours	**21.547**	**0.6854**	**0.2602**	+0.975
RE10K	5	180°	LVSM	17.133	0.5000	0.4503	—
			Ours	**18.226**	**0.5783**	**0.3962**	+1.093
DL3DV	3	90°	LVSM	21.059	0.6849	0.2706	—
			Ours	**22.107**	**0.7534**	**0.2314**	+1.048
DL3DV	5	180°	LVSM	16.669	0.5345	0.4665	—
			Ours	**18.660**	**0.6037**	**0.4055**	+1.991
DL3DV	7	270°	LVSM	16.019	0.4188	**0.5096**	—
			Ours	**16.227**	**0.4568**	0.5249	+0.208

(b) Two-input angular separation

Dataset	Angle	Model	PSNR↑	SSIM↑	LPIPS↓	ΔPSNR
RE10K	30°	LVSM	24.511	0.8071	0.1643	—
		Ours	**25.214**	**0.8309**	**0.1465**	+0.703
RE10K	45°	LVSM	22.122	0.7302	0.2291	—
		Ours	**22.616**	**0.7503**	**0.1986**	+0.494
RE10K	90°	LVSM	16.221	0.5522	0.5153	—
		Ours	**16.792**	**0.5596**	**0.4774**	+0.571
RE10K	180°	LVSM	13.304	0.4917	**0.6727**	—
		Ours	**13.616**	**0.4961**	0.7067	+0.312
DL3DV	30°	LVSM	24.800	0.8128	0.1893	—
		Ours	**25.352**	**0.8469**	**0.1660**	+0.552
DL3DV	45°	LVSM	22.848	0.7719	0.2363	—
		Ours	**23.315**	**0.7881**	**0.2256**	+0.467
DL3DV	90°	LVSM	17.013	**0.5982**	0.4908	—
		Ours	**17.036**	0.5895	**0.4818**	+0.023
DL3DV	180°	LVSM	13.437	0.5308	**0.6415**	—
		Ours	**13.848**	**0.5350**	0.6864	+0.411

Generalization to the Number of Input Views. The input-view generalization result in Fig. 11b shows that the design preserves its advantage when the number of test-time input views changes. This suggests that semantic-spatial decoupling improves the underlying representation rather than overfitting to a fixed context size.

Wide-Angle Robustness. We evaluate multi-input trajectory spans and two-input angular separations on RealEstate10K and DL3DV [55]. Table 3 shows PSNR gains in every setting, with improvements for trajectories through 180° and a smaller gain at 270°. SSIM and LPIPS gains are less consistent under extreme viewpoints. Both models degrade with larger viewpoint changes and remain limited by target content absent from the inputs. Examples are provided in the wide-angle gallery.

4.5 Computational Overhead

We evaluate inference latency under the same setting for all variants, measuring only the model forward pass after warm-up and reporting average per-sample latency. Training-only components, including teacher models, point maps, and auxiliary projection heads, are removed before timing.

As shown in Fig. 11c, base decoupling has nearly the same inference latency as the baseline, because it preserves full-token Q/K attention and does not keep supervision modules at inference. The full model adds only the bidirectional modulation modules, resulting in a modest latency increase. Detailed parameter derivation and timing protocol are provided in Appendix D.

5 Conclusion

We revisited feedforward Transformer-based novel view synthesis and identified that entangling semantic appearance and spatial conditioning within a shared token space induces representational ambiguity and grid-like distortions, which in turn degrade rendering fidelity. To address this, we propose to decouple semantic-spatial features with Independent-V attention that preserves branch purity while retaining controlled cross-stream interaction via shared routing. Extensive experiments and analysis show that the decoupling is effective for feedforward NVS Transformers.

References

[1] 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 *European Conference on Computer Vision (ECCV)*, pages 405–421. Springer, 2020.

[2] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics (TOG)*, 42(4): 1–14, 2023.

[3] Yicong Hong, Kai Zhang, Jiuxiang Gu, Sai Bi, Yang Zhou, Difan Liu, Feng Liu, Kalyan Sunkavalli, Trung Bui, and Hao Tan. Lrm: Large reconstruction model for single image to 3d. In *International Conference on Learning Representations (ICLR)*, 2024.

[4] Kai Zhang, Sai Bi, Hao Tan, Yuanbo Xiangli, Nanxuan Zhao, Kalyan Sunkavalli, and Zexiang Xu. Gs-lrm: Large reconstruction model for 3d gaussian splatting. In *European Conference on Computer Vision (ECCV)*, 2024.

[5] 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 *International Conference on Learning Representations (ICLR)*, 2025.

[6] Jiawei Yang, Katie Z Luo, Jiefeng Li, Congyue Deng, Leonidas Guibas, Dilip Krishnan, Kilian Q Weinberger, Yonglong Tian, and Yue Wang. Denoising vision transformers. In *Computer Vision – ECCV 2024*, volume 15143 of *Lecture Notes in Computer Science*, pages 453–469. Springer, 2024. doi: 10.1007/978-3-031-73013-9_26.

[7] Timothée Darcet, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. Vision transformers need registers. In *International Conference on Learning Representations (ICLR)*, 2024.

[8] Paul E Debevec, Camillo J Taylor, and Jitendra Malik. Modeling and rendering architecture from photographs: A hybrid geometry- and image-based approach. In *Proceedings of SIGGRAPH*, pages 11–20, 1996.

[9] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

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

[11] 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 *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

[12] Chen Ziwen, Hao Tan, Kai Zhang, Sai Bi, Fujun Luan, Yicong Hong, Fuxin Li, and Zexiang Xu. Long-lrm: Long-sequence large reconstruction model for wide-coverage gaussian splats. *arXiv preprint arXiv:2410.12781*, 2024. doi: 10.48550/arXiv.2410.12781.

[13] 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. *arXiv preprint arXiv:2505.00702*, 2025. doi: 10.48550/arXiv.2505.00702.

[14] Xiaosong Jia, Yihang Sun, Junqi You, Songbur Wong, Zichen Zou, Junchi Yan, Zuxuan Wu, and Yu-Gang Jiang. Efficient-lvsm: Faster, cheaper, and better large view synthesis model via decoupled co-refinement attention. *arXiv preprint arXiv:2602.06478*, 2026. doi: 10.48550/arXiv.2602.06478.

[15] Shikun Liu, Kam Woh Ng, Wonbong Jang, Jiadong Guo, Junlin Han, Haozhe Liu, Yiannis Douratsos, Juan C Pérez, Zijian Zhou, Chi Phung, Tao Xiang, and Juan-Manuel Pérez-Rúa. Scaling sequence-to-sequence generative neural rendering. *arXiv preprint arXiv:2510.04236*, 2025. doi: 10.48550/arXiv.2510.04236.

[16] Jensen Zhou, Hang Gao, Vikram Voleti, Aaryaman Vasishta, Chun-Han Yao, Mark Boss, Philip Torr, Christian Rupprecht, and Varun Jampani. Stable virtual camera: Generative view synthesis with diffusion models. *arXiv preprint arXiv:2503.14489*, 2025. doi: 10.48550/arXiv.2503.14489.

[17] Xuanchi Ren, Tianchang Shen, Jiahui Huang, Huan Ling, Yifan Lu, Merlin Nimier-David, Thomas Müller, Alexander Keller, Sanja Fidler, and Jun Gao. GEN3C: 3D-Informed World-Consistent Video Generation with Precise Camera Control, 2025. URL https://arxiv.org/abs/2503.03751.

[18] Ruiqi Gao, Aleksander Holynski, Philipp Henzler, Arthur Brussee, Ricardo Martin-Brualla, Pratul Srinivasan, Jonathan T. Barron, and Ben Poole. CAT3D: Create Anything in 3D with Multi-View Diffusion Models, 2024. URL https://arxiv.org/abs/2405.10314.

[19] Sherwin Bahmani, Tianchang Shen, Jiawei Ren, Jiahui Huang, Yifeng Jiang, Haithem Turki, Andrea Tagliasacchi, David B. Lindell, Zan Gojcic, Sanja Fidler, Huan Ling, Jun Gao, and Xuanchi Ren. Lyra: Generative 3D Scene Reconstruction via Video Diffusion Model Self-Distillation, 2025. URL https://arxiv.org/abs/2509.19296.

[20] Tianchang Shen, Sherwin Bahmani, Kai He, Sangeetha Grama Srinivasan, Tianshi Cao, Jiawei Ren, Ruilong Li, Zian Wang, Nicholas Sharp, Zan Gojcic, Sanja Fidler, Jiahui Huang, Huan Ling, Jun Gao, and Xuanchi Ren. Lyra 2.0: Explorable Generative 3D Worlds, 2026. URL https://arxiv.org/abs/2604.13036.

[21] Jiaxiang Tang, Zhaoxi Chen, Xiaokang Chen, Tengfei Wang, Gang Zeng, and Ziwei Liu. LGM: Large Multi-View Gaussian Model for High-Resolution 3D Content Creation, 2024. URL https://arxiv.org/abs/2402.05054.

[22] Jianfeng Xiang, Zelong Lv, Sicheng Xu, Yu Deng, Ruicheng Wang, Bowen Zhang, Dong Chen, Xin Tong, and Jiaolong Yang. Structured 3D Latents for Scalable and Versatile 3D Generation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. URL https://arxiv.org/abs/2412.01506.

[23] Team Hunyuan3D, Shuhui Yang, Mingxin Yang, Yifei Feng, Xin Huang, Sheng Zhang, Zebin He, Di Luo, Haolin Liu, Yunfei Zhao, Qingxiang Lin, Zeqiang Lai, Xianghui Yang, Huiwen Shi, Zibo Zhao, Bowen Zhang, Hongyu Yan, Lifu Wang, Sicong Liu, Jihong Zhang, Meng Chen, Liang Dong, Yiwen Jia, Yulin Cai, Jiaao Yu, Yixuan Tang, Dongyuan Guo, Junlin Yu, Hao Zhang, Zheng Ye, Peng He, Runzhou Wu, Shida Wei, Chao Zhang, Yonghao Tan, Yifu Sun, Lin Niu, Shirui Huang, Bojian Zheng, Shu Liu, Shilin Chen, Xiang Yuan, Xiaofeng Yang, Kai Liu, Jianchen Zhu, Peng Chen, Tian Liu, Di Wang, Yuhong Liu, Linus, Jie Jiang, Jingwei Huang, and Chunchao Guo. Hunyuan3D 2.1: From Images to High-Fidelity 3D Assets with Production-Ready PBR Material, 2025. URL https://arxiv.org/abs/2506.15442.

[24] Zirui Wu, Zeren Jiang, Martin R Oswald, and Jie Song. From rays to projections: Better inputs for feed-forward view synthesis. *arXiv preprint arXiv:2601.05116*, 2026. doi: 10.48550/arXiv.2601.05116.

[25] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations (ICLR)*, 2021.

[26] Alexander Lappe and Martin A Giese. Register and [cls] tokens yield a decoupling of local and global features in large vits. *arXiv preprint arXiv:2505.05892*, 2025. doi: 10.48550/arXiv.2505.05892.

[27] Federico Barbero, Álvaro Arroyo, Xiangming Gu, Christos Perivolaropoulos, Michael Bronstein, Petar Veličković, and Razvan Pascanu. Why do llms attend to the first token? *arXiv preprint arXiv:2504.02732*, 2025. doi: 10.48550/arXiv.2504.02732.

[28] Alexis Marouani, Oriane Siméoni, Hervé Jégou, Piotr Bojanowski, and Huy V Vo. Revisiting [cls] and patch token interaction in vision transformers. *arXiv preprint arXiv:2602.08626*, 2026. doi: 10.48550/arXiv.2602.08626.

[29] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

[30] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023. doi: 10.48550/arXiv.2304.07193.

[31] Oriane Siméoni, Huy V Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, Francisco Massa, Daniel Haziza, Luca Wehrstedt, Jianyuan Wang, Timothée Darcet, Théo Moutakanni, Leonel Sentana, Claire Roberts, Andrea Vedaldi, Jamie Tolan, John Brandt, Camille Couprie, Julien Mairal, Hervé Jégou, Patrick Labatut, and Piotr Bojanowski. DINOv3. *arXiv preprint arXiv:2508.10104*, 2025. doi: 10.48550/arXiv.2508.10104.

[32] Sihyun Yu, Sangkyung Kwak, Huiwon Jang, Jongheon Jeong, Jonathan Huang, Jinwoo Shin, and Saining Xie. Representation alignment for generation: Training diffusion transformers is easier than you think. *arXiv preprint arXiv:2410.06940*, 2024. doi: 10.48550/arXiv.2410.06940.

[33] Yuchuan Tian, Hanting Chen, Mengyu Zheng, Yuchen Liang, Chao Xu, and Yunhe Wang. U-repa: Aligning diffusion u-nets to vits. *arXiv preprint arXiv:2503.18414*, 2025. doi: 10.48550/arXiv.2503.18414.

[34] Xingjian Leng, Jaskirat Singh, Yunzhong Hou, Zhenchang Xing, Saining Xie, and Liang Zheng. Repa-e: Unlocking vae for end-to-end tuning with latent diffusion transformers. *arXiv preprint arXiv:2504.10483*, 2025. doi: 10.48550/arXiv.2504.10483.

[35] Jaskirat Singh, Xingjian Leng, Zongze Wu, Liang Zheng, Richard Zhang, Eli Shechtman, and Saining Xie. What matters for representation alignment: Global information or spatial structure? *arXiv preprint arXiv:2512.10794*, 2025. doi: 10.48550/arXiv.2512.10794.

[36] Boyang Zheng, Nanye Ma, Shengbang Tong, and Saining Xie. Diffusion transformers with representation autoencoders. *arXiv preprint arXiv:2510.11690*, 2025. doi: 10.48550/arXiv.2510.11690.

[37] Shariq Farooq Bhat, Reiner Birkl, Diana Wofk, Peter Wonka, and Matthias Müller. Zoedepth: Zero-shot transfer by combining relative and metric depth. *arXiv preprint arXiv:2302.12288*, 2023. doi: 10.48550/arXiv.2302.12288.

[38] Lihe Yang, Bingyi Kang, Zilong Huang, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything: Unleashing the power of large-scale unlabeled data. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[39] Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything v2. *arXiv preprint arXiv:2406.09414*, 2024. doi: 10.48550/arXiv.2406.09414.

[40] Aleksei Bochkovskii, Amaël Delaunoy, Hugo Germain, Marcel Santos, Yichao Zhou, Stephan R Richter, and Vladlen Koltun. Depth pro: Sharp monocular metric depth in less than a second. *arXiv preprint arXiv:2410.02073*, 2024. doi: 10.48550/arXiv.2410.02073.

[41] Ruicheng Wang, Sicheng Xu, Cassie Dai, Jianfeng Xiang, Yu Deng, Xin Tong, and Jiaolong Yang. Moge: Unlocking accurate monocular geometry estimation for open-domain images with optimal training supervision. *arXiv preprint arXiv:2410.19115*, 2024. doi: 10.48550/arXiv.2410.19115.

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

[43] Haotong Lin, Sili Chen, Junhao Liew, Donny Y Chen, Zhenyu Li, Guang Shi, Jiashi Feng, and Bingyi Kang. Depth anything 3: Recovering the visual space from any views. *arXiv preprint arXiv:2511.10647*, 2025. doi: 10.48550/arXiv.2511.10647.

[44] Stanislaw Szymanowicz, Minghao Chen, Jianyuan Wang, Christian Rupprecht, and Andrea Vedaldi. LagerNVS: Latent Geometry for Fully Neural Real-time Novel View Synthesis, 2026. URL https://arxiv.org/abs/2603.20176.

[45] Julius Plücker. On a new geometry of space. *Philosophical Transactions of the Royal Society of London*, 155:725–791, 1865.

[46] Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron Courville. FiLM: Visual reasoning with a general conditioning layer. In *AAAI Conference on Artificial Intelligence*, pages 3942–3951, 2018.

[47] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: Learning view synthesis using multiplane images. In *ACM SIGGRAPH*, 2018.

[48] Matt Deitke, Dustin Schwenk, Jordi Salvador, Luca Weihs, Oscar Michel, Eli VanderBilt, Ludwig Schmidt, Kiana Ehsani, Aniruddha Kembhavi, and Ali Farhadi. Objaverse: A universe of annotated 3d objects. In *CVPR*, 2023.

[49] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4): 600–612, 2004.

[50] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 586–595, 2018.

[51] 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 *Computer Vision – ECCV 2024*, pages 370–386, 2024. doi: 10.1007/978-3-031-72664-4_21. URL https://arxiv.org/abs/2403.14627.

[52] Ruilong Li, Brent Yi, Junchen Liu, Hang Gao, Yi Ma, and Angjoo Kanazawa. Cameras as Relative Positional Encoding, 2025. URL https://arxiv.org/abs/2507.10496.

[53] Haofei Xu, Songyou Peng, Fangjinhua Wang, Hermann Blum, Daniel Barath, Andreas Geiger, and Marc Pollefeys. DepthSplat: Connecting Gaussian Splatting and Depth. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16453–16463, 2025. URL https://arxiv.org/abs/2410.13862.

[54] David Charatan, Sizhe Lester Li, Andrea Tagliasacchi, and Vincent Sitzmann. pixelSplat: 3D Gaussian Splats from Image Pairs for Scalable Generalizable 3D Reconstruction. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19457–19467, 2024. URL https://arxiv.org/abs/2312.12337.

[55] Lu Ling, Yichen Sheng, Zhi Tu, Wentian Zhao, Cheng Xin, Kun Wan, Lantao Yu, Qianyu Guo, Zixun Yu, Yawen Lu, Xuanmao Li, Xingpeng Sun, Rohan Ashok, Aniruddha Mukherjee, Hao Kang, Xiangrui Kong, Gang Hua, Tianyi Zhang, Bedrich Benes, and Aniket Bera. DL3DV-10K: A Large-Scale Scene Dataset for Deep Learning-based 3D Vision, 2023. URL https://arxiv.org/abs/2312.16256.

[56] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*, 2019.

[57] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *ECCV*, pages 694–711, 2016.

[58] Shinji Umeyama. Least-squares estimation of transformation parameters between two point patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(4):376–380, 1991. doi: 10.1109/34.88573.

[59] Gyeongjin Kang, Seungtae Nam, Seungkwon Yang, Xiangyu Sun, Sameh Khamis, Abdelrahman Mohamed, and Eunbyung Park. ilrm: An iterative large 3d reconstruction model, 2025. URL https://arxiv.org/abs/2507.23277.

A Implementation and Training Details

Controlled Training Budget. The LVSM baselines are controlled reimplementations rather than numbers copied from the original backbone papers. This setting evaluates the relative effect of semantic–spatial decoupling under a fixed budget; absolute baseline numbers may differ from original reports that use different training scales, model variants, or tuning.

Optimizer and Learning Rate Schedule. Our LVSM models are trained with AdamW [56] (β_1=0.9, β_2=0.95, weight decay 5×10^{-2}). For scene-level training, we use a constant learning rate of 4×10^{-4} with a warm-up of 3 000 iterations. For object-level training, we use the same learning rate with a warm-up of 2 500 iterations. These LVSM experiments use mixed precision (BF16).

Dataset-Specific Schedules. For scene-level training on RealEstate10K [47], we use 2 input views and 6 target views, training on 4 A100 80G GPUs with a per-GPU batch size of 4 for 50 K iterations at a resolution of 256×256. For object-level training on Objaverse [48], we use 4 input views and 8 target views, training on 8 A100 80G GPUs with a per-GPU batch size of 4 for 50 K iterations. LVSM baselines and decoupled variants use the same schedule within each benchmark.

External Baseline Protocols. For deterministic sparse-view NVS on RealEstate10K, all trainable comparison methods use the same data split, view sampling, 256×256 resolution, four A100 GPUs, batch size, 50 K training iterations, and fixed evaluation views. We retain each method's native losses, optimizers, and pretrained components. Thus, the comparison matches data, hardware, training budget, and evaluation conditions while preserving method-specific training designs. The methods' pretrained priors are not equated by this protocol.

Model-Specific Configurations. LagerNVS [44] provides a foundation-model-assisted reference. We follow its original design, replacing the VGGT-1B encoder with the DA3-S DINOv2 backbone and reducing the renderer from 12 to 4 layers to match the parameter scale. MVSplat [51] uses camera intrinsics and poses under the same RealEstate10K training and evaluation protocol. PRoPE [52] and raw Raymap are implemented as controlled LVSM variants. DepthSplat [53] and pixelSplat [54] retain their official architectures and pretrained encoders.

Object-Level Comparison. On Objaverse, LGM [21] and the LVSM-based models use the same training and evaluation protocol described above. This comparison tests transfer to object-centric scenes, where decoupling and the optional supervision and modulation also improve reconstruction.

Compute Resources.

Table 4: **Training Compute.** GPU-hours are computed as wall-clock hours multiplied by the number of A100 GPUs.

Experiment	GPUs	Wall-clock time	GPU-hours
RealEstate10K Decoder-Only	4 A100	8 h	32
Objaverse Decoder-Only	8 A100	15 h	120
RealEstate10K Encoder-Decoder	4 A100	12 h	48
RealEstate10K iLRM	4 A100	8 h	32

Loss Weights and Reconstruction Loss. The RGB reconstruction loss $\mathcal{L}_{\text{RGB}}$ is a weighted combination of an L2 loss and a VGG perceptual loss [57]:

$$\mathcal{L}_{\text{RGB}} = \mathcal{L}_{\text{L2}} + \lambda_{\text{VGG}}\mathcal{L}_{\text{VGG}}. \tag{13}$$

We set λ_{VGG}=0.5 for scene-level training and λ_{VGG}=1.0 for object-level training. LPIPS is disabled in the training objective. For supervised variants, the categorized supervision weights are set to λ_I=0.5 for iREPA supervision and λ_P=0.5 for geometric consistency.

iREPA Supervision Details. Following iREPA [35], we use DINOv3 ViT-B/16 [31] as the frozen teacher and take its layer-8 output as the alignment target. The student feature is captured from the semantic branch at layer k_I. Before matching, we apply spatial normalization with γ=0.60 and ϵ=10^{-6}, followed by a lightweight 3×3 convolutional projector that maps the $D/2$-dimensional semantic feature to the DINOv3 feature dimension. The matching loss is Smooth L1. The projector and DINOv3 teacher are discarded at inference.

Geometric Grounding Details. For each scene or object, training views are processed offline by DA3 [43] to obtain depth maps and camera parameters. We align the DA3 camera coordinates to the training camera coordinates with Umeyama registration [58], back-project the depth maps into 3D

point maps, and cache the resulting point maps and validity masks. During training, valid source-view 3D points are projected into target views with the aligned camera parameters, and z-buffering keeps the nearest valid correspondence for each target location. We capture spatial-branch features at layer k_P and apply cosine consistency between corresponding source and target features. View pairs with fewer than 100 valid correspondences are excluded, with the threshold relaxed to 50 for Objaverse. The point maps are used only for training supervision and are not required at inference.

B Additional Ablations and Scaling

B.1 Component Ablations across Datasets and Architectures

We extend the component study to RealEstate10K and Objaverse using LVSM Decoder-Only and Encoder–Decoder architectures. The variants share the training protocol and differ only in the optional categorized supervision and bidirectional modulation. Table 5 shows that either addition improves all three metrics over decoupling alone, and their combination performs best. The same trend across the four settings supports the use of these components on top of the decoupled representation.

Table 5: **Component ablations across datasets and architectures.** All configurations use decoupling; +Sup/+Mod adds categorized supervision/ bidirectional modulation, and Full adds both. Bold/italics indicate the best/second-best metric within each architecture.

(a) RealEstate10K

Config.	PSNR↑	SSIM↑	LPIPS↓
Decoder-Only			
Decouple	26.70	0.851	0.138
+ Sup	26.91	0.857	0.134
+ Mod	*27.06*	*0.860*	*0.131*
Full	**27.21**	**0.869**	**0.125**
Encoder–Decoder			
Decouple	24.84	0.789	0.170
+ Sup	25.03	0.795	0.165
+ Mod	*25.17*	*0.798*	*0.161*
Full	**25.31**	**0.806**	**0.154**

(b) Objaverse

Config.	PSNR↑	SSIM↑	LPIPS↓
Decoder-Only			
Decouple	25.96	0.880	0.112
+ Sup	26.17	0.887	0.108
+ Mod	*26.31*	*0.890*	*0.106*
Full	**26.46**	**0.899**	**0.101**
Encoder–Decoder			
Decouple	24.16	0.816	0.137
+ Sup	24.34	0.822	0.133
+ Mod	*24.48*	*0.826*	*0.130*
Full	**24.61**	**0.834**	**0.124**

B.2 Cross-View Geometry Evaluation

We evaluate whether geometric supervision improves spatial correspondence using a foreground-depth probe on Objaverse. We freeze the decouple-only and Decouple + Geo models and extract their layer-10 P-branch features. For each foreground target patch, cosine similarity selects the most similar source patch. We back-project the selected source patch using its mesh ground-truth depth, transform the resulting point into the target camera frame, and use its target-frame depth as the prediction. This probe measures the quality of feature correspondences using source ground-truth geometry; it does not introduce a learned depth-prediction head.

Table 6: **Cross-view foreground-depth evaluation on Objaverse.** Geo denotes spatial-branch geometric supervision. Bold indicates the better result for each metric.

Model	AbsRel↓	Norm. RMSE↓	δ_1 ↑
Decouple-only	0.1210	0.0968	0.8535
Decouple + Geo	**0.0988**	**0.0829**	**0.8970**

Geometric supervision improves all three foreground-depth metrics (Table 6). The improvement supports more accurate cross-view correspondences in the supervised spatial representation, providing geometric evidence beyond the rendering metrics.

B.3 Supervision Layer Placement

This ablation is performed on the LVSM Decoder-Only model on RealEstate10K. We study the attachment layer of the two optional branch-specific supervision terms introduced in Sec. 3.3. When varying the semantic iREPA layer $k_I \in \{2, 4, 8\}$, we keep the geometric supervision layer fixed at

Table 7: **Ablation on Supervision Layer Placement.** We vary the layer of semantic iREPA alignment and geometric grounding.

(a) iREPA layer k_I

k_I	PSNR↑	SSIM↑	LPIPS↓
2	26.51	0.844	0.138
4	26.91	0.857	0.134
8	26.12	0.840	0.144

Geometric supervision fixed at k_P=10.

(b) Geometric layer k_P

k_P	PSNR↑	SSIM↑	LPIPS↓
6	26.61	0.843	0.138
10	26.91	0.857	0.134
12	26.47	0.841	0.140

iREPA fixed at k_I=4.

k_P=10. When varying the geometric layer $k_P \in \{6, 10, 12\}$, we keep the semantic iREPA layer fixed at k_I=4.

The best setting is k_I=4 and k_P=10. Very early supervision can over-constrain low-level features before the two branches have formed stable representations, while very late supervision is closer to the rendering head and provides weaker guidance to intermediate representations.

B.4 Resolution and Backbone Scaling

We test resolution scaling with LVSM Decoder-Only on RealEstate10K. Ours denotes semantic–spatial decoupling with bidirectional modulation, without categorized supervision. The 512×512 models are fine-tuned from their corresponding 256×256 checkpoints for 10K iterations with the same training settings. In Table 8, Steps denotes training iterations at resolution 256 and fine-tuning iterations at resolution 512. Our decoupled variant improves all three rendering metrics at both resolutions and across the reported checkpoints. The retained advantage at resolution 512 suggests that the representation benefit persists as more detailed image evidence becomes available.

Table 8: **Resolution scaling on RealEstate10K.** Res. denotes image side length in pixels. Bold highlights the best Ours result in each metric across the reported 512-resolution checkpoints; reported ΔPSNR is in dB.

Res.	Steps	LVSM			Ours			ΔPSNR
		PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	
256	10K	23.54	0.7555	0.2205	24.66	0.7949	0.1891	+1.12
256	20K	24.86	0.8026	0.1747	25.95	0.8360	0.1514	+1.09
256	30K	25.14	0.8142	0.1611	26.41	0.8475	0.1409	+1.27
256	40K	25.22	0.8066	0.1553	26.93	0.8582	0.1341	+1.71
256	50K	26.10	0.8387	0.1444	27.06	0.8603	0.1313	+0.96
512	2K	27.20	0.8656	0.1808	27.97	0.8795	0.1722	+0.77
512	4K	27.30	0.8665	0.1786	28.02	0.8801	0.1714	+0.72
512	6K	27.26	0.8676	0.1767	28.03	0.8815	0.1698	+0.77
512	8K	27.47	0.8707	0.1770	**28.28**	0.8833	0.1682	+0.81
512	10K	27.45	0.8712	0.1759	28.19	**0.8834**	**0.1681**	+0.74

We also train a 24-layer LVSM Decoder-Only backbone under the same RealEstate10K protocol. Ours uses decoupling with bidirectional modulation and no categorized supervision. Table 9 shows improvements throughout training, indicating that additional model capacity does not remove the benefit of decoupling in this setting. The 50K results are also included in Table 1. These experiments test resolution and backbone depth under the stated budgets; broader training-scale validation remains a direction for future work.

Table 9: **Backbone scaling on RealEstate10K.** Both models use a 24-layer decoder-only backbone; Ours adds decoupling and modulation without supervision. Bold/italics indicate the best/second-best result in each metric column across training checkpoints.

Steps	LVSM			Ours			ΔPSNR
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	
10K	23.72	0.7637	0.2143	25.27	0.8152	0.1729	+1.55
20K	25.04	0.8044	0.1715	26.36	0.8452	0.1426	+1.32
30K	25.47	0.8213	0.1557	26.66	0.8537	0.1326	+1.19
40K	*25.76*	*0.8264*	*0.1497*	*26.98*	*0.8598*	*0.1304*	+1.22
50K	**26.18**	**0.8420**	**0.1406**	**27.37**	**0.8671**	**0.1249**	+1.19

B.5 Generalization to the iLRM Architecture

We further apply decoupling with bidirectional modulation to iLRM [59], a feed-forward 3D reconstruction model that predicts 3D Gaussian Splatting parameters. Unlike LVSM, iLRM refines image-free Gaussian query tokens through alternating Read blocks and Self-Attention blocks. Read blocks let query tokens attend to context-view support tokens, while Self-Attention blocks propagate interactions among the query tokens.

In our controlled iLRM setting, the processor uses 12 alternating blocks with patch size 8, token dimension 768, and head dimension 64. We decouple both the query stream and the support stream. Query tokens use zero-initialized semantic branches and Plücker-initialized spatial branches; support tokens use RGB patches for the semantic branch and Plücker-ray patches for the spatial branch. The Read and Self-Attention blocks then use the same Independent-V, branch-wise normalization, branch-specific FFN, and bidirectional modulation design described in Sec. 3.2 and Sec. 3.4. The final token is decoded by the original iLRM Gaussian head, leaving the output representation unchanged.

We train the iLRM baseline and decoupled variant on RealEstate10K under the same controlled protocol. We use 2 context views and 4 target views at 256×256 resolution. Both models use AdamW, learning rate 4×10^{-4}, 2 500 warm-up iterations, gradient clipping at 1.0, BF16 mixed precision, 4 A100 80G GPUs, per-GPU batch size 32, and 5 K training iterations. The reconstruction objective follows the iLRM configuration and uses RGB MSE loss. No teacher model, point map, or auxiliary projection head is used at inference.

Under this matched setting, Table 10 shows improvements in all three metrics, supporting transfer of the combined decoupling and modulation design to iLRM.

Table 10: **Quantitative Comparison on iLRM.** Decoupling with bidirectional modulation improves reconstruction under the matched training protocol.

Model	PSNR ↑	SSIM ↑	LPIPS ↓
iLRM baseline	22.42	0.764	0.276
iLRM-Decouple + Mod (Ours)	23.15	0.789	0.245

B.6 Complementary Generative Comparisons

We compare with adapted GEN3C [17] and Lyra [19]. Their input protocols, pretrained priors, and model scales differ from those of feedforward reconstruction. We therefore report these experiments as complementary references and separate adapted models from published full-scale results.

GEN3C. We replace its large video-diffusion backbone with MobileI2V, retaining the independent 3D-cache construction and in-layer cache fusion. We adapt the latent projections and temporal/spatial interfaces to the new backbone and train this variant on the same RealEstate10K split for 50 K iterations. Table 11(a) shows that our model improves all three metrics over this adapted variant with fewer parameters. The published full-scale GEN3C result has lower LPIPS under its original protocol and is included as a contextual reference.

Lyra. We follow its static single-image-to-3D setting: one source image is provided, while the model internally generates and fuses multiple camera-controlled trajectories. Our GEN3C-MobileI2V model serves as a frozen teacher, and we train Lyra's 3DGS decoder for 50 K iterations using trajectories generated from RealEstate10K source images. Plücker camera conditioning, multi-trajectory fusion, and static 3DGS reconstruction are retained. Table 11(b) shows that our model improves over the adapted Lyra model, while the published full-scale Lyra result is stronger under its original setting. The latter uses a roughly 7B-parameter pretrained video model, longer and higher-resolution generated trajectories, and broader pretraining.

Table 11: **Complementary generative comparisons on RealEstate10K.** Adapted denotes MobileI2V-based variants; Params are in millions. * denotes published results under the original protocols. Bold compares Ours with the adapted model only.

(a) GEN3C

Model	Params	PSNR↑	SSIM↑	LPIPS↓
Adapted	267.4	20.57	0.652	0.318
Published*	1521	24.08	0.860	0.110
Ours	57.2	**27.21**	**0.869**	**0.125**

(b) Lyra (single-image input)

Model	Params	PSNR↑	SSIM↑	LPIPS↓
Adapted	304.6	17.71	0.570	0.364
Published*	7033	21.79	0.752	0.219
Ours	57.2	**18.65**	**0.589**	**0.328**

These comparisons clarify the scope of our contribution: decoupling improves observation-faithful reconstruction without iterative diffusion sampling, whereas generative priors can support perceptual quality and plausible completion of unobserved content. The two directions can also be combined, as in pipelines that use generated views for reconstruction [21, 18, 19]. Related generative approaches, including Lyra 2.0 [20], TRELLIS [22], and Hunyuan3D 2.1 [23], also use different inputs, outputs, and evaluation objectives. These differences call for task-specific evaluation when relating their progress to feedforward NVS.

C Representation Diagnostics and Visualization

C.1 Quantifying Grid-Like Artifacts

We define the axis-aligned Grid Artifact Score (GAS) as

$$\mathrm{GAS} = 10 \log_{10} \left(\frac{E_{\mathrm{axis}}}{E_{\mathrm{diag}}} \right), \tag{14}$$

where E_{axis} is the mid/high-frequency spectral energy near the horizontal and vertical axes, and E_{diag} is the energy in matched diagonal sectors. Higher GAS indicates stronger axis-aligned grid structure. On 128 synthetic $1/f$ feature backgrounds, GAS increases monotonically with injected grid strength. Adding white noise or low-frequency structure changes GAS by less than 0.12 dB and 0.01 dB, respectively.

Table 12 combines the layer-wise measurements and the rendering association study on RealEstate10K. Register tokens do not reduce GAS in the entangled representation. Decoupling reduces it in the semantic I-branch, while the spatial P-branch retains strong axis-aligned structure. This distinction supports separating spatial structure from appearance features, rather than suppressing it in both branches. Across the five rendering variants, final-layer semantic GAS has Spearman correlation $\rho = -0.700$ with PSNR and $\rho = +0.700$ with LPIPS. This association concerns these five variants and alone does not establish a causal relationship.

Table 12: **Grid artifacts on RealEstate10K.** GAS (dB) at layers 8/12; I/P: semantic/spatial.

Variant	L8	L12	PSNR↑	LPIPS↓
Baseline	13.11	5.56	26.10	0.144
Baseline + Registers	13.85	5.67	26.04	0.146
Decouple + Joint LN (I)	—	4.55	26.01	0.149
Decouple (I)	5.06	*4.21*	*26.70*	*0.138*
Decouple + Modulation (I)	—	**3.88**	**27.06**	**0.131**
Decouple (P)	19.18	17.05	—	—

Table 13: **Layer-8 feature intervention.** A/D: axial/diagonal; ΔGAS and ΔPSNR in dB.

α	A ΔGAS	A ΔPSNR	D ΔPSNR	A−D ΔPSNR	RMS (A/D)
0.02	+0.24	-0.002	-0.003	+0.001	0.0165/0.0199
0.05	+0.59	-0.031	-0.022	-0.009	0.0407/0.0495
0.10	+1.15	-0.140	-0.095	-0.045	0.0796/0.0984
0.20	+2.19	-0.507	-0.394	-0.113	0.1515/0.1933

C.2 Controlled Feature Intervention

We intervene at layer 8 of the entangled baseline. At each strength α, we inject either an axis-aligned component or a diagonal component from the same frequency band. The diagonal injection controls for the general effect of perturbing features, while the axial injection increases grid-like structure. Table 13 reports changes relative to the unmodified baseline and the RMS feature changes.

For $\alpha \geq 0.05$, axial injection causes a larger PSNR reduction than the diagonal control despite a smaller RMS feature change. The excess degradation grows with intervention strength. At $\alpha = 0.02$, this difference is absent. Together, the results provide controlled evidence that increasing axis-aligned artifacts can harm rendering beyond the general effect of feature perturbations, without implying that all reconstruction errors arise from these artifacts. Layer-wise GAS curves and intervention plots are available in the online artifact analysis.

C.3 Layer-Wise Feature Visualization

Layer-wise Feature Evolution. Fig. 12 provides PCA visualizations of input-view and target-view features from the decouple-only model across multiple Transformer layers. The visualizations show that the decoupled representation produces organized feature structures throughout the network.

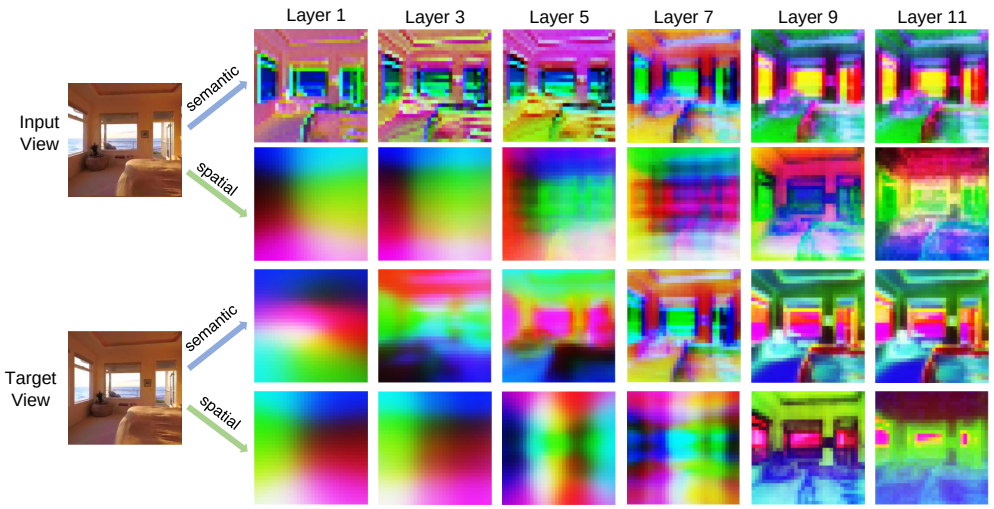


Figure 12: **Layer-wise Feature Visualization of Decouple-Only.** PCA visualizations of intermediate input-view and target-view features from the decouple-only model.

C.4 Qualitative Comparisons and Failure Cases

The online qualitative comparisons contain 12 RealEstate10K scenes and 12 Objaverse objects, with three novel views for each example. They complement Fig. 5 with sharper contours and reduced blur across varied scenes and objects. Rendering videos provide additional trajectory visualizations.

Challenging viewpoints and visibility. The wide-angle examples accompany Table 3. Although PSNR improves in every tested setting, the advantage shrinks under the extreme $270°$ trajectory, and LPIPS worsens in this setting and in both $180°$ two-input settings. When target content is observed but input-view overlap is low, both models can blur details, with decoupling generally reducing the blur. When target regions are absent from every input, both may produce gray or meaningless pixels because the necessary visual evidence is missing.

Complex structures. Fine structures, cluttered backgrounds, severe occlusions, and repetitive textures remain challenging. Our model generally produces sharper contours, but both models can still generate broken structures or inaccurate textures, as shown in the failure-case gallery. Decoupling improves the use of available evidence; it does not resolve the ambiguity of unobserved content in deterministic NVS.

D Inference Complexity Analysis

We provide the parameter derivation for Sec. 4.5. Let D denote the token dimension and r the FFN expansion ratio. Bias terms are omitted because they are lower-order $O(D)$ terms.

Baseline Block. For the entangled baseline, attention uses QKV and output projections, and the FFN uses hidden dimension rD:

$$\mathcal{P}_{\text{attn}}^{\text{base}} = 4D^2, \qquad \mathcal{P}_{\text{ffn}}^{\text{base}} = 2rD^2. \tag{15}$$

Thus,

$$\mathcal{P}_{\text{block}}^{\text{base}} = (2r+4)D^2 + O(D). \tag{16}$$

Base Decoupled Block. In the base decoupled block, Q/K are shared over the full token, while value, output, and FFN transformations are branch-specific:

$$\mathcal{P}_{\text{attn}}^{\text{dec}} = 3D^2, \qquad \mathcal{P}_{\text{ffn}}^{\text{dec}} = rD^2. \tag{17}$$

Therefore,

$$\mathcal{P}_{\text{block}}^{\text{dec}} = (r+3)D^2 + O(D), \tag{18}$$

which is no larger than the baseline block.

LayerNorm and Modulation. Branch-wise LayerNorm preserves parameter count because two full-dimensional LayerNorms are replaced by four half-dimensional LayerNorms:

$$2D = 4 \cdot \frac{D}{2}. \tag{19}$$

Bidirectional modulation adds two FiLM generators:

$$\mathcal{P}_{\text{FiLM}} = 2\left(\frac{D}{2} \cdot D + D\right) = D^2 + 2D. \tag{20}$$

With D=768, this adds about 0.59M parameters per block, or about 7.1M parameters across 12 layers.

Latency Protocol. Latency is measured under the same evaluation setting for all variants. We exclude warm-up batches and report average per-sample forward time. Image saving, metric computation, teacher models, point maps, and auxiliary heads are not included. Categorized supervision therefore does not affect inference latency, while bidirectional modulation adds the only extra inference-time projections.

E Limitations and Future Directions

Our evaluation uses controlled feedforward NVS budgets. Unobserved target content, extreme camera changes, and complex structures remain challenging (Appendix C.4). Future work includes larger training scales, more diverse outdoor and dynamic scenes, and longer trajectories; extending the design to other camera encodings and reconstruction backbones; and developing lighter supervision to reduce offline preprocessing.

F Broader Impacts

Improved feedforward reconstruction may benefit 3D content creation, AR/VR, and robotics simulation. It could also be misused to create misleading imagery, reconstruct private spaces, or generate unauthorized 3D representations. We recommend public, licensed, or consented training data, permission for privacy-sensitive applications, and clear disclosure of synthetic outputs. For planned code and model releases, we plan to document intended research use, dataset assumptions, and known limitations, and discourage privacy-invasive or misleading applications.

G Existing Assets, Licenses, and Terms

We use existing assets only for research and cite their original sources. RealEstate10K camera trajectories are released by Google LLC under CC BY 4.0; video frames remain subject to their source terms. Objaverse uses ODC-By v1.0, while objects retain their metadata licenses: CC BY 4.0, CC BY-NC 4.0, CC BY-NC-SA 4.0, CC BY-SA 4.0, or CC0 1.0. We follow asset-specific terms and do not redistribute the original objects.

For training-time supervision, DINOv3 ViT-B/16 uses the DINOv3 License; DA3NESTED-GIANT-LARGE uses CC BY-NC 4.0, and DA3 code uses Apache-2.0. Their checkpoints, intermediate caches, and teacher outputs are not redistributed with this submission.

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: The abstract and introduction distinguish semantic–spatial decoupling from optional supervision and modulation, with supporting methods and controlled experiments in Sections 3 and 4. Appendix B.6 separately identifies adapted generative comparisons and published results under different protocols.

   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: Section 4 and Appendix C.4 discuss extreme viewpoints, low overlap, unobserved content, and complex structures, including settings where SSIM or LPIPS does not improve. The appendix also includes a Limitations and Future Directions section addressing the controlled training scale and remaining extensions.

   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 present theoretical results, theorems, or formal proofs; its contribution is architectural and empirical.

   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: Sections 3 and 4, together with Appendix A, describe the architecture, objectives, supervision construction, and main benchmark protocols. Appendix B provides the additional ablation, scaling, iLRM, and adapted generative settings.

   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: [Yes]

   Justification: The public repository, linked from the project page, releases the core semantic–spatial decoupling implementation, configurations, and installation, training, and evaluation instructions for the main LVSM experiments. We use publicly available RealEstate10K and Objaverse data, with benchmark protocols described in Section 4 and Appendix A.

   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: Section 4 and Appendix A specify the main datasets, view sampling, resolution, optimizer, loss weights, batch sizes, training schedules, and metrics, while distinguishing external methods' native training designs. Appendix B gives additional protocols and the supervision-layer selection study.

   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 reported results do not include error bars, confidence intervals, or statistical significance tests across independent training runs. Trends across metrics, architectures, and datasets are empirical support, rather than a claim of statistical significance.

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: Appendix A specifies GPU type, GPU memory, GPU counts, batch sizes, and training schedules; Table 4 reports wall-clock training time and GPU-hours for the principal training settings. Additional training settings are described in Appendix B.

   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 study uses credited research datasets and pretrained models, and does not involve crowdsourcing or direct human-subject experiments. Appendices G and F describe asset usage and discuss privacy-sensitive and misleading applications.

   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: [Yes]

    Justification: Appendix F discusses benefits for 3D content creation, AR/VR, and robotics simulation, alongside misleading imagery, reconstruction of private spaces, and unauthorized 3D representations. It also recommends appropriate data use, permission for privacy-sensitive applications, and disclosure of synthetic outputs.

    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: [No]

    Justification: Appendix F discusses potential misuse and recommends responsible data use and release documentation. It does not report implemented access controls, safety filters, or other release safeguards.

    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: [No]

Justification: Appendix G names the licenses and terms for RealEstate10K, Objaverse, DINOv3, and DA3, and the paper credits the other datasets and methods. The manuscript does not explicitly enumerate licenses and versions for every additional dataset, baseline implementation, pretrained component, and evaluation asset.

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 released code includes a README describing dependencies, data preparation, supported configurations, and training and evaluation commands. Its license file states the project-code license and distinguishes third-party code and data terms.

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 work 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: The work does not involve crowdsourcing or human-subject research, so IRB approval or equivalent review is not applicable.

    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: LLM assistance is limited to writing, editing, and formatting the manuscript. LLMs are not part of the core method or experimental pipeline.

    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.