AIMS: ANCHOR-INTEGRATED MULTI-VIEW SYNTHESIS FOR SCALABLE NOVEL VIEW RENDERING

Anonymous authors
Paper under double-blind review

ABSTRACT

Feed-forward novel view synthesis methods achieve strong generalization from posed multi-view inputs, but scaling them to large input view sets remains challenging. Transformer-based approaches that jointly process all input-view tokens incur rapidly increasing computation and memory as the number of views grows, while simple view subsampling discards potentially useful observations. We introduce **Anchor-Integrated Multi-View Synthesis (AIMS)**, a scalable framework that decouples the number of available observations from the number of views processed by the global synthesis model. AIMS selects a fixed set of spatially distributed anchor views using farthest point sampling, groups nearby observations around each anchor, and uses a lightweight learnable integrator to fuse their information into enriched anchor representations. This allows additional observations to contribute to synthesis while keeping the downstream global view budget fixed. Evaluations on RealEstate10K and ScanNet demonstrate a favorable quality–efficiency trade-off against transformer-based and Gaussian-based baselines. AIMS achieves 29.41 dB and 17.73 dB PSNR on the two datasets, respectively, with rendering averaging 7.24 ms per view.

1 INTRODUCTION

All views are useful, but not all views need attention. Additional observations can reveal occluded surfaces, reduce geometric ambiguity, and provide richer evidence for novel view synthesis. Yet in transformer-based models, allowing every observation to participate directly in global attention makes computation grow with the input collection. We ask whether a model can benefit from many observations while reasoning globally over only a compact representation of them.

Recent progress in feed-forward novel view synthesis has made this question increasingly relevant. Many approaches rely on explicit 3D representations or geometry-aware priors, such as neural radiance fields, 3D Gaussians, and epipolar constraints. LVSM (Jin et al., 2025) takes a different direction, formulating novel view synthesis as a largely data-driven transformer problem with minimal 3D inductive bias. Given posed input images, it directly learns to synthesize target views without relying on a predefined 3D representation or rendering equation.

This flexibility, however, makes the size of the visual context an important computational bottleneck. In the decoder-only LVSM, input and target tokens interact through full self-attention, causing computation to grow rapidly as additional views are introduced. Efficient-LVSM (Jia et al., 2026) substantially improves this scaling through decoupled attention, but incorporating more observations still requires processing a larger collection of input tokens. The encoder–decoder variant of LVSM may appear to avoid this issue by compressing the inputs into a fixed number of latent tokens; however, the encoder must still apply full self-attention over the input tokens before producing this fixed-size representation. As a result, its computational cost also grows with the number of input views. Thus, existing formulations still couple the amount of available scene evidence with the computation required to exploit it.

In this work, we introduce **Anchor-Integrated Multi-View Synthesis** (**AIMS**), a framework that incorporates evidence from large view collections under a *fixed global attention budget*. Inspired by the Point Transformer (Zhao et al., 2021), which handles large point sets by selecting representative points and aggregating information from their local neighborhoods, AIMS applies a similar principle to large collections of input views. Specifically, AIMS selects a compact set of representative

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

anchor views and enriches their representations with information from neighboring observations. The resulting integrated anchors summarize a larger input collection and serve as the inputs to the global synthesis model. With a fixed anchor count and per-anchor token budget, the model's per-target global attention computation remains independent of the number of input observations. The anchor and neighbor counts provide complementary controls over scene detail and computational cost: more anchors expand the capacity of the global representation, while more neighbors supply additional evidence for local refinement. Increasing these counts can support finer scene detail at higher computational cost, whereas smaller settings favor a more compact or coarser representation with lower processing cost.

Our evaluation of AIMS on RealEstate10K (Zhou et al., 2018) and ScanNet (Dai et al., 2017) demonstrates improvements in both synthesis quality and computational efficiency. With 12 anchors and six views per group, AIMS achieves 29.41 dB and 17.73 dB PSNR, respectively, outperforming the evaluated baselines. Its reusable scene representation further enables efficient synthesis of additional views, with rendering averaging 7.24 ms per view.

Our contributions are summarized as follows:

- We formulate large-input novel view synthesis under a *fixed global attention budget*, allowing N observations to contribute through $G \ll N$ integrated anchor views while keeping the number of views participating in global synthesis fixed.
- We introduce **AIMS**, a hierarchical view integration framework that combines coverage-aware anchor selection with learnable local aggregation, enabling non-anchor observations to contribute without directly expanding the global attention context.
- We conduct comprehensive evaluations on RealEstate10K and ScanNet, demonstrating that AIMS offers a favorable trade-off between reconstruction quality and computational efficiency compared with the evaluated baselines.

2 Related Work

Foundations of Novel View Synthesis. Early novel view synthesis (NVS) generated new views from photographs using geometric proxies (Debevec et al., 1996) or densely sampled light fields (Levoy & Hanrahan, 1996; Gortler et al., 1996). Later methods improved image alignment and combination through depth-guided warping (Chaurasia et al., 2013), appearance flow (Zhou et al., 2016), learned blending (Hedman et al., 2018), and depth prediction (Choi et al., 2019). Neural scene representations offered another route: NeRF (Mildenhall et al., 2021) optimizes continuous density and radiance fields through differentiable volume rendering. Subsequent work improved anti-aliasing and view-dependent appearance (Barron et al., 2021; Verbin et al., 2024; Barron et al., 2023), while voxel grids (Sun et al., 2022; Fridovich-Keil et al., 2022), tensor factorization (Chen et al., 2022), and hash encoding (Müller et al., 2022) increased efficiency. Point-based alternatives associate neural features with 3D points in Point-NeRF (Xu et al., 2022) or rasterize anisotropic Gaussians in 3D Gaussian Splatting (Kerbl et al., 2023), with extensions addressing aliasing (Yu et al., 2024), distorted cameras, and secondary rays (Wu et al., 2025). While these developments advance scene reconstruction, the cost of per-scene optimization motivates models that infer scene content from learned priors.

Generalizable Reconstruction and Rendering. Generalizable NVS learns priors across scenes to predict novel views or scene representations in a feed-forward manner. Early examples include pixelNeRF (Yu et al., 2021), which conditions radiance fields on pixel-aligned features, and IBRNet (Wang et al., 2021), which aggregates multi-view evidence. Geometric reasoning was further developed through plane-sweep cost volumes in MVSNeRF (Chen et al., 2021) and geometry and visibility cues in GeoNeRF (Johari et al., 2022) and NeuRay (Liu et al., 2022). Following advances in Gaussian rendering, feed-forward models began predicting Gaussian primitives directly: pixelSplat (Charatan et al., 2024) uses image pairs, MVSplat (Chen et al., 2024b) uses multi-view cost volumes, and DepthSplat (Xu et al., 2025) combines matching with pretrained monocular depth features. NoPoSplat (Ye et al., 2025) extends this direction to canonical reconstruction from sparse unposed images. LRM (Hong et al., 2024) and GS-LRM (Zhang et al., 2024) scale image-to-scene prediction using transformers and triplane or Gaussian representations. To improve multi-view pro-

cessing, LaRa (Chen et al., 2024a) uses group attention over Gaussian volumes. Alongside deterministic reconstruction, generative methods address unobserved regions through camera-conditioned diffusion (Watson et al., 2023; Liu et al., 2023), with CAT3D (Gao et al., 2024) and Stable Virtual Camera (Zhou et al., 2025) supporting multi-view generation.

Scalable Transformer-Based View Synthesis. Alongside methods that predict predefined 3D representations, learned renderers synthesize views directly from scene context and target cameras. Light Field Networks (Sitzmann et al., 2021) predict color from ray coordinates, while Geometry-Free View Synthesis (Rombach et al., 2021) and ViewFormer (Kulhánek et al., 2022) use transformers to synthesize views from encoded image context. Scene Representation Transformer (Sajjadi et al., 2022) organizes observations into latent tokens queried by target rays, whereas GPNR (Suhail et al., 2022) retains geometric guidance through epipolar feature sampling. LVSM (Jin et al., 2025) advances direct synthesis at the patch level, using encoder–decoder and decoder-only architectures with minimal 3D inductive bias. However, extending transformer-based synthesis to larger input collections makes attention cost a central concern. Efficient-LVSM (Jia et al., 2026) addresses this through intra-view input attention and target self-then-cross attention, achieving linear input-view scaling and enabling input-feature caching. Further work pursues efficient inference through reusable contexts and unidirectional decoding in SVSM (Kim et al., 2026), shared reconstruction and rendering weights with key–value caching in DVSM (Sun et al., 2026), and geometry-pretrained features with lightweight rendering in LagerNVS (Szymanowicz et al., 2026). LaCT's NVS model (Zhang et al., 2026) instead compresses multi-view evidence into fast weights through test-time updates. AIMS addresses the input-scaling problem through local view integration, compressing neighboring observations into fixed-size anchor representations before global synthesis. This allows additional views to contribute without expanding the global attention context or applying full-input self-attention.

3 METHOD

We introduce **Anchor-Integrated Multi-View Synthesis (AIMS)**, which synthesizes novel views from posed RGB observations through a compact scene representation. Given N observations $\mathcal{O} = \{(I_i, \mathbf{A}_i, \mathbf{T}_i)\}_{i=1}^{N}$, where $\mathbf{A}_i$ and $\mathbf{T}_i$ denote camera intrinsics and camera-to-world poses, our goal is to predict an image $\widehat{I}_t$ at a target camera $(\mathbf{A}_t, \mathbf{T}_t)$. The pipeline consists of three stages: *coverage-aware view grouping*, which selects G representative anchors and groups nearby observations around them; *view group encoding*, where our **View Group Encoder (VGE)** integrates ray-conditioned image tokens from each anchor and its neighbors into L scene tokens; and *target-view decoding*, which synthesizes the target image from the resulting GL scene tokens. With G and L fixed, the global attention cost remains independent of the number of input observations N. The overall pipeline is visualized in Fig. 1.

3.1 COVERAGE-AWARE VIEW GROUPING

Inspired by the sampling and local aggregation in Point Transformer (Zhao et al., 2021), we select representative anchor views and group nearby observations around them. Let $\mathbf{R}_i \in \mathbb{R}^{3\times3}$ and $\mathbf{c}_i \in \mathbb{R}^3$ denote the rotation matrix and center of camera i in a normalized scene frame. Anchor selection uses deterministic farthest-point sampling in camera-pose space. We define the pose distance by combining the separation between camera centers with the angle between their orientations:

$$\begin{aligned} \delta(i,j) &= \sqrt{w_{\mathrm{t}}\|\mathbf{c}_i - \mathbf{c}_j\|_2^2 + w_{\mathrm{r}}\theta_{ij}^2}, \\ \theta_{ij} &= \arccos\left(\frac{\mathrm{tr}(\mathbf{R}_i^\top \mathbf{R}_j) - 1}{2}\right). \end{aligned} \tag{1}$$

The relative rotation $\mathbf{R}_i^\top \mathbf{R}_j$ satisfies $\mathrm{tr}(\mathbf{R}_i^\top \mathbf{R}_j) = 1 + 2\cos\theta_{ij}$, where tr sums the diagonal entries. Solving this relation using $\arccos$ yields the smallest alignment angle $\theta_{ij} \in [0, \pi]$, a scalar measure of orientation difference in radians. The weights w_{t} and w_{r} balance this angular distance with the distance between camera centers. We use $(w_{\mathrm{t}}, w_{\mathrm{r}}) = (1, 0.25)$.

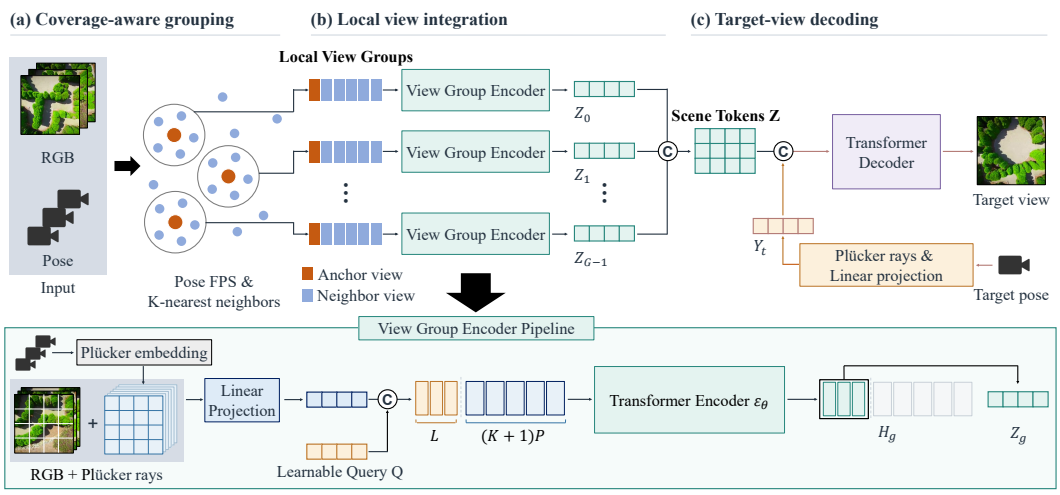


Figure 1: Overview of AIMS. Coverage-aware grouping selects anchors and neighboring views. The View Group Encoder compresses each group's ray-conditioned image tokens into compact scene tokens. A global decoder combines these with target ray tokens to synthesize novel views.

We initialize the selected set with the pose medoid $a_1 = \arg\min_i \sum_j \delta_{\mathrm{a}}(i,j)$ and iteratively select the camera farthest from the current anchors:

$$a_g = \arg\max_{i \notin \mathcal{A}_{g-1}} \min_{a \in \mathcal{A}_{g-1}} \delta_{\mathrm{a}}(i,a), \tag{2}$$

where $\mathcal{A}_{g-1} = \{a_1, \ldots, a_{g-1}\}$. This procedure encourages coverage of the available camera poses without requiring image features or reconstructed geometry.

After selecting the anchors, we form a local view group $\mathcal{G}_g$ around each anchor a_g by including the anchor and its K nearest neighbors under δ_{n}, allowing overlaps between groups.

3.2 View Group Encoder

The VGE integrates the complementary observations within each local view group $\mathcal{G}_g$ into a fixed number of scene tokens. We first tokenize the group's images together with their camera poses. The camera poses are encoded using per-pixel Plücker ray embeddings (Plucker, 1865), computed using the corresponding camera intrinsics. For each source view, we concatenate corresponding RGB and ray embedding patches and linearly project them into tokens $\mathbf{E}_i \in \mathbb{R}^{P \times D}$, where $P = HW/p^2$ is the number of $p \times p$ patches in an $H \times W$ image and D is the token dimension.

We concatenate the view tokens $\mathbf{E}_i$ for $i \in \mathcal{G}_g$ into $\mathbf{X}_g \in \mathbb{R}^{(K+1)P \times D}$ and prepend L learnable tokens $\mathbf{Q}$ shared across groups. The VGE is implemented as a shared transformer encoder $\mathcal{E}_\theta$ that jointly processes these tokens. We retain the outputs corresponding to the learnable tokens:

$$\mathbf{H}_g = \mathcal{E}_\theta([\mathbf{Q}; \mathbf{X}_g]), \qquad \mathbf{Z}_g = \mathbf{H}_g[:L] \in \mathbb{R}^{L \times D}, \tag{3}$$

where $[:L]$ selects the first L output tokens corresponding to the learnable queries $\mathbf{Q}$. The resulting $\mathbf{Z}_g$ summarizes the anchor and its neighbors in $\mathcal{G}_g$. Concatenating all group outputs gives the scene tokens $\mathbf{Z} = [\mathbf{Z}_1; \ldots; \mathbf{Z}_G] \in \mathbb{R}^{GL \times D}$.

3.3 Target View Decoding

Following the target view decoding formulation of LVSM (Jin et al., 2025), we obtain target query tokens $\mathbf{Y}_t \in \mathbb{R}^{P_t \times D}$ by patchifying the target camera's Plücker ray embeddings and applying a separate linear projection, where P_t is the number of target patches. A global transformer decoder $\mathcal{D}_\phi$ processes the concatenation of the target query tokens $\mathbf{Y}_t$ and scene tokens $\mathbf{Z}$, and we retain the target-token outputs:

$$\mathbf{V}_t = \mathcal{D}_\phi([\mathbf{Y}_t; \mathbf{Z}]), \qquad \mathbf{F}_t = \mathbf{V}_t[:P_t] \in \mathbb{R}^{P_t \times D}, \tag{4}$$

where $[:P_t]$ selects the first P_t output tokens corresponding to the target queries. The retained tokens $\mathbf{F}_t$ are projected to RGB patches and assembled into the predicted image $\widehat{I}_t$. Since $\mathbf{Z}$ is independent of the target camera, the scene tokens support synthesis for multiple target viewpoints.

By compressing each group into L scene tokens, AIMS limits the sequence used for target-view decoding to GL scene tokens and P_t target query tokens. This gives a per-target attention cost of

$$\text{attn_cost} = \mathcal{O}\left((GL + P_t)^2\right). \tag{5}$$

For fixed G and L, the global attention cost remains independent of the input count N, maintaining a fixed global attention budget for target view decoding.

3.4 Loss Function

For each target view, we supervise the predicted image $\widehat{I}_t$ against the ground-truth image I_t using a combination of mean squared error and perceptual loss:

$$\mathcal{L} = \text{MSE}(\widehat{I}_t, I_t) + \lambda\ \text{Perceptual}(\widehat{I}_t, I_t), \tag{6}$$

where $\lambda = 0.5$ and Perceptual denotes LPIPS with a VGG backbone (Zhang et al., 2018). The MSE term encourages pixel-wise accuracy, while the perceptual term encourages visual similarity.

4 Experiments

4.1 Datasets

RealEstate10K. We use RealEstate10K (Zhou et al., 2018), which contains approximately 80K video clips curated from 10K YouTube videos covering both indoor and outdoor scenes.

ScanNet. ScanNet (Dai et al., 2017) contains RGB-D sequences of indoor environments with calibrated cameras. We include ScanNet as a complementary benchmark, as qualitative inspection of our sampled sequences suggested greater intra-scene camera-pose variation in ScanNet than in RealEstate10K.

Input and target views. We sample input and target views from the RealEstate10K test set and 843 ScanNet scenes. For each scene, we first uniformly sample 100 frames to form the observation set. We then reserve up to eight uniformly spaced views as targets. From the remaining views, we construct the input set using the coverage-aware grouping strategy described in Sec. 3.1, with 12 anchors and 5 neighbors per anchor. Using these inputs, AIMS integrates each local view group through the View Group Encoder (VGE) before global synthesis, while the baselines process the selected images using their original architectures. Since the released baseline checkpoints were trained with few input views and may not generalize well to large input collections, we additionally evaluate them using only $2, 4, 8$, or 12 representative anchor images selected via respective coverage-aware view grouping.

4.2 Comparison with Baselines

We compare against the encoder–decoder and decoder-only variants of LVSM (Jin et al., 2025), Efficient-LVSM (Jia et al., 2026), MVSplat (Chen et al., 2024b), and DepthSplat (Xu et al., 2025) on the RealEstate10K and ScanNet datasets. For each baseline, we use its officially released RealEstate10K checkpoint. We likewise train AIMS on RealEstate10K for 20 epochs and evaluate it on both datasets. See Appendix A for additional details on training. We report PSNR, SSIM, and VGG-based LPIPS (Zhang et al., 2018), averaged over target images, alongside inference time and qualitative comparisons (Fig. 2). Tables 1 and 2 jointly compare reconstruction quality and inference time using all selected observations and anchor-only baseline inputs, respectively.

4.2.1 Reconstruction quality

Using all selected observations (Table 1), AIMS achieves 29.41 dB PSNR, 0.8971 SSIM, and 0.0943 LPIPS on RealEstate10K, outperforming the baselines across all three quality metrics and exceeding

Table 1: Reconstruction quality and inference time for baselines evaluated on all observations selected by coverage-aware view grouping with $G = 12$ and $K = 5$. Encoding is performed once per scene; rendering uses cached representations where available, while LVSM (D) reports full per-view inference. MVSplat fails due to out-of-memory errors. Best results are bold.

	RealEstate10K			ScanNet			Encode	Render
Method	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	(ms/scene)↓	(ms/view)↓
DepthSplat	18.28	0.6780	0.3169	15.70	0.5528	**0.5118**	1043.04	9.84
MVSplat	–	–	–	–	–	–	–	–
LVSM (E–D)	18.72	0.5751	0.3896	12.69	0.4954	0.6916	758.93	9.62
LVSM (D)	24.32	0.8009	0.1770	17.34	0.5698	0.5496	–	1584.35
Efficient-LVSM	22.06	0.7301	0.2093	14.68	0.5453	0.6271	112.08	52.25
AIMS (ours)	**29.41**	**0.8971**	**0.0943**	**17.73**	**0.5774**	0.5201	161.54	7.24

Table 2: Reconstruction quality and inference time for baselines using only 2, 4, 8, or 12 anchor views. AIMS retains $G = 12$ anchors and $K = 5$ neighbors per anchor. Encoding is performed once per scene; rendering uses cached representations where available, while LVSM (D) reports full per-view inference. Best results are bold.

		RealEstate10K			ScanNet			Encode	Render
Method	Anchors	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	(ms/scene)↓	(ms/view)↓
DepthSplat	2	22.97	0.8017	0.1980	11.36	0.4345	0.6177	41.66	7.21
	4	24.03	0.8598	0.1526	13.55	0.5180	0.5754	65.15	7.30
	8	22.21	0.8305	0.1834	15.21	0.5612	0.5223	114.10	8.50
	12	21.19	0.8002	0.2108	15.76	0.5757	**0.4971**	172.59	9.07
MVSplat	2	21.50	0.7590	0.2196	8.52	0.1810	0.6437	34.08	4.11
	4	21.71	0.7989	0.2084	9.94	0.2915	0.6154	58.21	4.65
	8	20.21	0.7718	0.2429	11.90	0.4087	0.5894	151.95	5.20
	12	19.57	0.7551	0.2599	12.94	0.4529	0.5818	309.39	6.16
LVSM (E–D)	2	24.93	0.8174	0.1818	12.81	0.5058	0.6625	10.60	9.50
	4	25.60	0.8280	0.1520	13.86	0.5239	0.6369	15.07	9.49
	8	22.68	0.7387	0.2289	14.61	0.5349	0.6247	26.25	9.50
	12	21.36	0.6888	0.2760	14.71	0.5358	0.6293	42.58	9.62
LVSM (D)	2	25.67	0.8326	0.1629	12.78	0.5103	0.6615	–	12.72
	4	28.02	0.8795	0.1051	14.14	0.5291	0.6201	–	20.33
	8	28.24	0.8896	0.0987	15.92	0.5614	0.5691	–	40.07
	12	27.78	0.8832	0.1039	16.85	**0.5779**	0.5473	–	66.43
Efficient-LVSM	2	23.27	0.7631	0.2012	12.72	0.5069	0.6635	7.50	8.79
	4	21.43	0.7022	0.2117	13.49	0.5200	0.6437	7.53	9.19
	8	21.97	0.7239	0.2041	14.13	0.5307	0.6303	11.61	9.01
	12	22.08	0.7290	0.2054	14.38	0.5356	0.6277	14.87	10.62
AIMS (ours)	12	**29.41**	**0.8971**	**0.0943**	**17.73**	0.5774	0.5201	161.54	7.24

decoder-only LVSM by 5.08 dB PSNR. On ScanNet, AIMS achieves the highest PSNR and SSIM in the table, at 17.73 dB and 0.5774, respectively, compared with 17.34 dB PSNR for decoder-only LVSM. AIMS obtains an LPIPS of 0.5201, improving on the transformer baselines, while DepthSplat achieves a lower LPIPS of 0.5118.

With anchor-only inputs (Table 2), baselines including encoder–decoder LVSM and DepthSplat achieve their best RealEstate10K results with four anchors. However, on ScanNet, where camera configurations are more complex, all evaluated baselines achieve higher PSNR as the anchor count increases from two to 12, suggesting that such scenes benefit from broader view coverage. Yet moving from anchor-only inputs to all selected observations in Table 1 yields limited or inconsistent quality gains for the released baselines, highlighting their difficulty in exploiting larger input collections.



Figure 2: Qualitative comparison with transformer-based baselines, including LVSM encoder–decoder (E–D), LVSM decoder-only (D), and Efficient-LVSM, on RealEstate10K and ScanNet.

4.2.2 COMPUTATIONAL COST

We assess the quality–cost trade-off using the inference times measured on an NVIDIA RTX PRO 6000 Blackwell Server Edition GPU. AIMS, encoder–decoder LVSM, Efficient-LVSM, MVSplat, and DepthSplat encode the scene once and reuse the resulting representation across target views, whereas decoder-only LVSM performs full inference for each target view.

When all selected observations are used (Table 1), Efficient-LVSM achieves the lowest one-time encoding cost at 112.08 ms per scene, compared with 161.54 ms for AIMS. Once encoded, however, AIMS achieves the fastest per-view rendering time at 7.24 ms, corresponding to a $1.33\times$ speedup over the next-fastest method, encoder–decoder LVSM at 9.62 ms per view.

The anchor-only comparison in Table 2 further isolates rendering efficiency under different input budgets. Despite integrating substantially more observations, AIMS achieves the fastest per-view rendering time among the transformer-based methods at 7.24 ms.

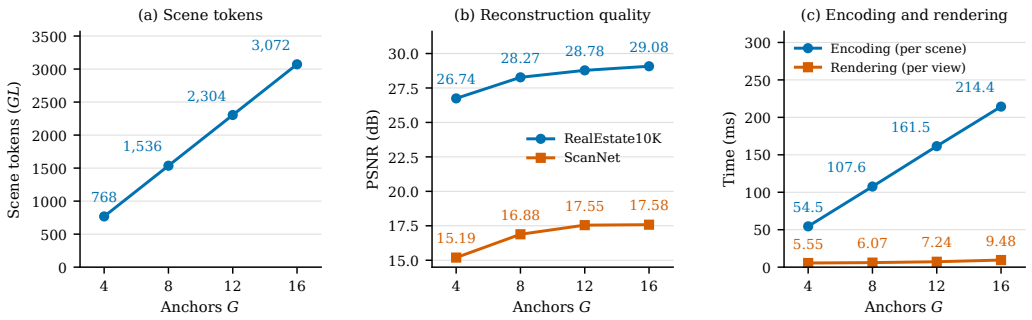


Figure 3: Quality–cost trade-off across anchor counts for controlled 10-epoch training runs with $G \in \{4, 8, 12, 16\}$ and fixed $K = 5$. (a) Number of scene tokens, (b) reconstruction quality on RealEstate10K and ScanNet, and (c) scene encoding and per-view rendering time.

Table 3: Effect of inference-time neighborhood size K on reconstruction quality. All settings use the same AIMS model trained with $G = 12$ and $K = 5$, without retraining. Best results are bold.

	RealEstate10K			ScanNet		
K	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
1	29.07	0.8916	0.0973	16.82	0.5649	0.5352
3	**29.50**	**0.8988**	**0.0932**	17.49	0.5749	0.5237
5	29.41	0.8971	0.0943	17.73	0.5774	0.5201
7	29.23	0.8941	0.0961	**17.82**	**0.5779**	**0.5192**

These results demonstrate that AIMS effectively integrates larger input sets to deliver strong reconstruction quality while maintaining low per-view rendering cost. Its fixed global attention budget offers a favorable trade-off between reconstruction quality and computational cost.

4.3 Ablation Studies and Analysis

4.3.1 Effects of anchor count and neighborhood size

To examine the trade-off between reconstruction quality and computational cost, we conduct controlled 10-epoch training runs with $G \in \{4, 8, 12, 16\}$ while fixing the neighborhood size to $K = 5$. As shown in Fig. 3, reconstruction quality improves as the number of anchors increases, but the gains diminish at larger G. Increasing G from 12 to 16 increases the number of scene tokens from 2,304 to 3,072, resulting in higher computational cost: scene encoding time increases from 161.54 to 214.36 ms and per-view rendering time from 7.24 to 9.48 ms. This corresponds to approximately 33% more scene tokens, 33% higher encoding cost, and 31% higher rendering cost, while PSNR improves by only 0.29 dB on RealEstate10K and 0.03 dB on ScanNet. After weighing the performance gains against the additional computational cost, we select $G = 12$ as our main configuration.

While G determines the size of the global scene representation, K controls the local context aggregated into each anchor. We next examine this complementary design choice by varying $K \in \{1, 3, 5, 7\}$ at inference time while keeping $G = 12$. This experiment uses a single 20-epoch checkpoint trained with $G = 12$ and $K = 5$—the same checkpoint used in our main baseline comparison—without retraining. As shown in Table 3, the effect of neighborhood size differs across datasets. On RealEstate10K, which predominantly contains smooth camera trajectories and relatively constrained viewpoint variation, reconstruction quality peaks at $K = 3$, with larger neighborhoods providing no further improvement. In contrast, ScanNet exhibits consistent gains as K increases from 1 to 7: PSNR increases from 16.82 to 17.82 dB, while SSIM improves from 0.5649 to 0.5779 and LPIPS decreases from 0.5352 to 0.5192. This suggests that larger local neighborhoods provide useful complementary observations in more complex scenes, where individual anchors may capture insufficient local scene information.

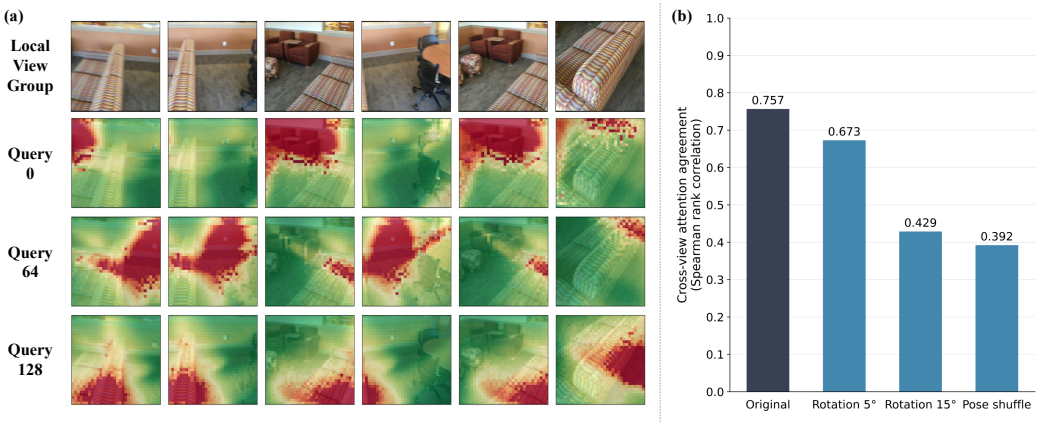


Figure 4: Analysis of VGE attention and its sensitivity to camera pose. (a) Head-averaged attention maps at VGE layer 6 for learnable queries 0, 64, and 128 across a local view group. (b) Cross-view attention agreement under camera-pose perturbations, measured by the mean Spearman rank correlation over matched physical locations across 50 ScanNet scenes.

4.3.2 Analyzing camera-pose utilization in VGE attention

VGE aggregates information from multiple observations within each view group. To understand what guides this aggregation, we begin by visualizing how its learnable queries attend to the input views. Fig. 4(a) shows the head-averaged attention maps of queries 0, 64, and 128 at VGE layer 6. The queries exhibit distinct spatial preferences, emphasizing the armchairs, the wall near the electrical outlet, and the sofa corner, respectively. These preferences recur across views in which the corresponding regions are visible. We therefore ask whether these patterns reflect image appearance alone or whether camera pose also guides cross-view aggregation.

To investigate this, we randomly select 50 of the 843 ScanNet scenes and construct 12 anchor–neighbor groups per scene, each containing one anchor and five neighbors, yielding 600 groups. Within each group, we use depth and camera poses to identify patches observing the same scene locations. For each learnable query at layer 6, we compute Spearman's rank correlation (Spearman, 1987) between attention values at these matched patches, measuring how consistently the query prioritizes the same locations across views. We then repeat the analysis while modifying only the camera poses supplied to the model. Each camera orientation is rotated by 5° or 15° about an independently sampled random axis while keeping its position fixed. We also shuffle complete poses among views. The images, view groups, and patch correspondences established from the original poses remain fixed across all conditions.

As shown in Fig. 4(b), the mean correlation decreases from 0.757 with the original poses to 0.673 and 0.429 under 5° and 15° rotations, respectively, and to 0.392 under pose shuffling. This decline despite unchanged image content suggests that camera pose plays a nontrivial role in shaping VGE's cross-view aggregation. We hypothesize that this sensitivity enables VGE to encode aggregated scene information in a more spatially aware manner, yielding more informative anchor representations and contributing to AIMS's reconstruction performance. Further details on the analysis procedure are provided in Appendix A.2.

5 Conclusion

In this work, we present AIMS, a transformer-based novel view synthesis framework that addresses the growing attention cost of large input collections. By combining coverage-aware view grouping with the View Group Encoder, AIMS integrates anchors and neighboring observations into compact scene tokens for synthesis under a fixed global attention budget. Experiments on RealEstate10K and ScanNet demonstrate high-quality reconstructions with low per-view rendering cost. These results establish hierarchical view integration as an effective approach to balancing reconstruction quality and computational cost in scalable novel view synthesis.

6 STATEMENTS

6.1 AI USE STATEMENT

In this work, we used generative AI tools to assist with code implementation, refine the mathematical description of our method, and support the drafting and revision of manuscript text for clarity, grammar, and consistency. The authors reviewed all AI-assisted content, verified quantitative claims against experimental outputs, and checked citations against the original sources. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

6.2 REPRODUCIBILITY STATEMENT

We describe the AIMS architecture and training objective in Sec. 3, the datasets and evaluation protocol in Sec. 4, and implementation and training details in Appendix A.1. Appendix A.2 provides additional details of the attention analysis. Upon publication, we will publicly release our source code, pretrained checkpoints, and training and evaluation configurations to support reproduction of our results.

REFERENCES

Jonathan T Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P Srinivasan. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 5835–5844. IEEE, 2021.

Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Zip-nerf: Anti-aliased grid-based neural radiance fields. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 19640–19648. IEEE, 2023.

David Charatan, Sizhe Lester Li, Andrea Tagliasacchi, and Vincent Sitzmann. pixelsplat: 3d gaussian splats from image pairs for scalable generalizable 3d reconstruction. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 19457–19467. IEEE, 2024.

Gaurav Chaurasia, Sylvain Duchene, Olga Sorkine-Hornung, and George Drettakis. Depth synthesis and local warps for plausible image-based navigation. *ACM transactions on graphics (TOG)*, 32 (3):1–12, 2013.

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 *2021 IEEE/CVF international conference on computer vision (ICCV)*, pp. 14104–14113. IEEE, 2021.

Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. Tensorf: Tensorial radiance fields. In *European conference on computer vision*, pp. 333–350. Springer, 2022.

Anpei Chen, Haofei Xu, Stefano Esposito, Siyu Tang, and Andreas Geiger. Lara: Efficient large-baseline radiance fields. In *European conference on computer vision*, pp. 338–355. Springer, 2024a.

Yuedong Chen, Haofei Xu, Chuanxia Zheng, Bohan Zhuang, Marc Pollefeys, Andreas Geiger, Tat-Jen Cham, and Jianfei Cai. Mvsplat: Efficient 3d gaussian splatting from sparse multi-view images. In *European conference on computer vision*, pp. 370–386. Springer, 2024b.

Inchang Choi, Orazio Gallo, Alejandro Troccoli, Min H Kim, and Jan Kautz. Extreme view synthesis. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 7780–7789. IEEE, 2019.

Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *2017 IEEE conference on computer vision and pattern recognition (CVPR)*, pp. 2432–2443. IEEE, 2017.

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 the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, pp. 11–20, 1996. doi: 10.1145/237170.237191.

Sara Fridovich-Keil, Alex Yu, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance fields without neural networks. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5491–5500. IEEE, 2022.

Ruiqi Gao, Aleksander Holynski, Philipp Henzler, Arthur Brussee, Ricardo Martin-Brualla, Pratul P. Srinivasan, Jonathan T. Barron, and Ben Poole. Cat3d: Create anything in 3d with multi-view diffusion models. *Advances in Neural Information Processing Systems*, 2024.

Steven J. Gortler, Radek Grzeszczuk, Richard Szeliski, and Michael F. Cohen. The lumigraph. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, pp. 43–54, 1996. doi: 10.1145/237170.237200.

Peter Hedman, Julien Philip, True Price, Jan-Michael Frahm, George Drettakis, and Gabriel Brostow. Deep blending for free-viewpoint image-based rendering. *ACM Transactions on Graphics (ToG)*, 37(6):1–15, 2018.

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*, volume 2024, pp. 50678–50702, 2024.

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. *International Conference on Learning Representations (ICLR)*, 2026.

Haian Jin, Hanwen Jiang, Hao Tan, Kai Zhang, Sai Bi, Tianyuan Zhang, Fujun Luan, Noah Snavely, and Zexiang Xu. Lvsm: A large view synthesis model with minimal 3d inductive bias. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=QQBPWtvtcn.

Mohammad Mahdi Johari, Yann Lepoittevin, and François Fleuret. Geonerf: Generalizing nerf with geometry priors. In *2022 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 18344–18347. IEEE, 2022.

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

Evan Kim, Hyunwoo Ryu, Thomas W Mitchel, and Vincent Sitzmann. Scaling view synthesis transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 28893–28902, 2026.

Jonáš Kulhánek, Erik Derner, Torsten Sattler, and Robert Babuška. Viewformer: Nerf-free neural rendering from few images using transformers. In *European Conference on Computer Vision*, pp. 198–216. Springer, 2022.

Marc Levoy and Pat Hanrahan. Light field rendering. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, pp. 31–42, 1996. doi: 10.1145/237170.237199.

Ruoshi Liu, Rundi Wu, Basile Van Hoorick, Pavel Tokmakov, Sergey Zakharov, and Carl Vondrick. Zero-1-to-3: Zero-shot one image to 3d object. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9264–9275. IEEE, 2023.

Yuan Liu, Sida Peng, Lingjie Liu, Qianqian Wang, Peng Wang, Christian Theobalt, Xiaowei Zhou, and Wenping Wang. Neural rays for occlusion-aware image-based rendering. In *2022 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 7814–7823. IEEE, 2022.

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

Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM transactions on graphics (TOG)*, 41(4):1–15, 2022.

Julius Plucker. I. on a new geometry of space. *Proceedings of the Royal Society of London*, (14): 53–58, 1865.

Robin Rombach, Patrick Esser, and Björn Ommer. Geometry-free view synthesis: Transformers and no 3d priors. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 14336–14346. IEEE, 2021.

Mehdi SM Sajjadi, Henning Meyer, Etienne Pot, Urs Bergmann, Klaus Greff, Noha Radwan, Suhani Vora, Mario Lučić, Daniel Duckworth, Alexey Dosovitskiy, et al. Scene representation transformer: Geometry-free novel view synthesis through set-latent scene representations. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6219–6228. IEEE, 2022.

Vincent Sitzmann, Semon Rezchikov, Bill Freeman, Josh Tenenbaum, and Fredo Durand. Light field networks: Neural scene representations with single-evaluation rendering. *Advances in neural information processing systems*, 34:19313–19325, 2021.

Charles Spearman. The proof and measurement of association between two things. *The American journal of psychology*, 100(3/4):441–471, 1987.

Mohammed Suhail, Carlos Esteves, Leonid Sigal, and Ameesh Makadia. Generalizable patch-based neural rendering. In *European Conference on Computer Vision*, pp. 156–174. Springer, 2022.

Cheng Sun, Min Sun, and Hwann-Tzong Chen. Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5449–5459. IEEE, 2022.

Cheng Sun, Jaesung Choe, Min-Hung Chen, Ryo Hachiuma, and Yu-Chiang Frank Wang. Dvsm: Decoder-only view synthesis model done right. *arXiv preprint arXiv:2605.29891*, 2026.

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

Dor Verbin, Peter Hedman, Ben Mildenhall, Todd Zickler, Jonathan T Barron, and Pratul P Srinivasan. Ref-nerf: Structured view-dependent appearance for neural radiance fields. *IEEE transactions on pattern analysis and machine intelligence*, 47(11):9426–9437, 2024.

Qianqian Wang, Zhicheng Wang, Kyle Genova, Pratul P Srinivasan, Howard Zhou, Jonathan T Barron, Ricardo Martin-Brualla, Noah Snavely, and Thomas Funkhouser. Ibrnet: Learning multi-view image-based rendering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4690–4699, 2021.

Daniel Watson, William Chan, Ricardo Martin-Brualla, Jonathan Ho, Andrea Tagliasacchi, and Mohammad Norouzi. Novel view synthesis with diffusion models. In *International Conference on Learning Representations*, 2023. URL https://arxiv.org/abs/2210.04628.

Qi Wu, Janick Martinez Esturo, Ashkan Mirzaei, Nicolas Moenne-Loccoz, and Zan Gojcic. 3dgut: Enabling distorted cameras and secondary rays in gaussian splatting. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 26036–26046. IEEE, 2025.

Haofei Xu, Songyou Peng, Fangjinhua Wang, Hermann Blum, Daniel Barath, Andreas Geiger, and Marc Pollefeys. Depthsplat: Connecting gaussian splatting and depth. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16453–16463. IEEE, 2025.

Qiangeng Xu, Zexiang Xu, Julien Philip, Sai Bi, Zhixin Shu, Kalyan Sunkavalli, and Ulrich Neumann. Point-nerf: Point-based neural radiance fields. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5428–5438. IEEE, 2022.

Botao Ye, Sifei Liu, Haofei Xu, Xueting Li, Marc Pollefeys, Ming-Hsuan Yang, and Songyou Peng. No pose, no problem: Surprisingly simple 3d gaussian splats from sparse unposed images. In *International Conference on Learning Representations*, volume 2025, pp. 54009–54033, 2025.

Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4576–4585. IEEE, 2021.

Zehao Yu, Anpei Chen, Binbin Huang, Torsten Sattler, and Andreas Geiger. Mip-splatting: Alias-free 3d gaussian splatting. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 19447–19456. IEEE, 2024.

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*, pp. 1–19. Springer, 2024.

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *2018 IEEE/CVF conference on computer vision and pattern recognition*, pp. 586–595. IEEE, 2018.

Tianyuan Zhang, Sai Bi, Yicong Hong, Kai Zhang, Fujun Luan, Songlin Yang, Kalyan Sunkavalli, William Freeman, and Hao Tan. Test-time training done right. In *International Conference on Learning Representations*, volume 2026, pp. 157604–157638, 2026.

Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip HS Torr, and Vladlen Koltun. Point transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 16259–16268, 2021.

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. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 12405–12414. IEEE, 2025.

Tinghui Zhou, Shubham Tulsiani, Weilun Sun, Jitendra Malik, and Alexei A Efros. View synthesis by appearance flow. In *European conference on computer vision*, pp. 286–301. Springer, 2016.

Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: Learning view synthesis using multiplane images. In *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2018)*, volume 37, New York, NY, USA, 2018. ACM. doi: 10.1145/3197517.3201323.

A Appendix

A.1 AIMS Implementation and Training Details

Images are resized and center-cropped to 256×256, with camera intrinsics adjusted accordingly and poses normalized to a common scene frame. We use 8×8 image patches, as in LVSM (Jin et al., 2025) and Efficient-LVSM (Jia et al., 2026). The View Group Encoder (VGE) and global decoder each contain 12 transformer blocks with hidden dimension $D = 768$ and 12 attention heads. Each group produces $L = 192$ scene tokens. Our main configuration uses $G = 12$ anchors and $K = 5$ neighbors per anchor, yielding six-view groups and $GL = 2,304$ scene tokens. Both anchor selection and neighborhood construction use pose-distance weights $w_{\mathrm{t}} = 1$ and $w_{\mathrm{r}} = 0.25$.

We initialize the source image tokenizer and decoding modules, including the target ray tokenizer and RGB prediction head, from LVSM's pretrained encoder–decoder checkpoint. The source image tokenizer remains frozen, while the randomly initialized VGE and pretrained decoding modules are optimized jointly. Training uses AdamW with learning rates of 2×10^{-4} for the VGE and

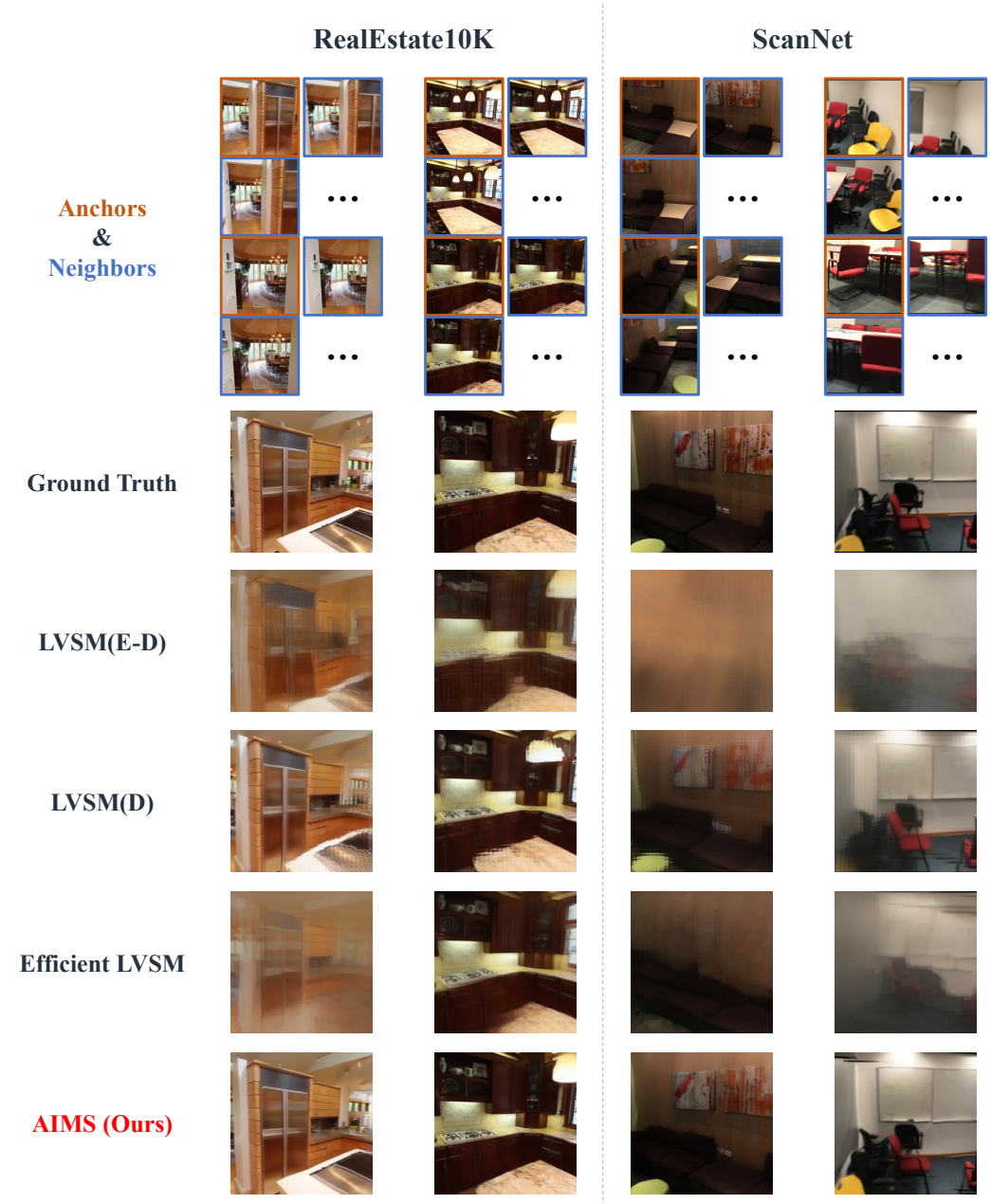


Figure 5: Additional qualitative comparisons with transformer-based baselines, including LVSM encoder–decoder (E–D), LVSM decoder-only (D), and Efficient-LVSM, on RealEstate10K and ScanNet.

5×10^{-5} for the decoding modules, $(\beta_1, \beta_2) = (0.9, 0.95)$, and weight decay 0.05. We train for 20 epochs with 500 linear warm-up steps followed by cosine decay to 5% of the initial learning rates. Training uses five NVIDIA RTX PRO 6000 Blackwell Server Edition GPUs, with 20 scenes per GPU and a global batch size of 100, without gradient accumulation. We use BF16 mixed precision, gradient checkpointing, gradient clipping at a norm of 1, and random seed 42. The training objective combines MSE and VGG-based LPIPS with perceptual-loss weight $\lambda = 0.5$.

We train on 66,033 RealEstate10K clips. For each scene, we uniformly sample 100 views to form the candidate view set. During training, we sample up to eight target views at random from this set, while validation uses up to eight uniformly spaced target views. In contrast to the evaluation setting

in the main paper, we permit source–target overlap during training. We select the checkpoint with the highest validation PSNR, which occurs at epoch 20.

For the anchor-count study, each configuration is trained for 10 epochs with $K = 5$. Gradient accumulation uses four steps for $G = 4, 8$ and two steps for $G = 12, 16$.

A.2 Implementation Details of the Attention Analysis

Setup and attention extraction. We use the same trained checkpoint as in the main comparison. The fixed evaluation cohort contains 50 distinct ScanNet scenes. For each scan, we load 100 uniformly sampled frames and exclude eight held-out target frames before applying the standard AIMS grouping procedure. We evaluate all 12 groups per scene, each containing one anchor and five neighbors. Images are resized and cropped to 256×256, producing 32×32 grids of 8×8 patches. We extract attention from all 192 learnable queries at the sixth VGE layer and average over attention heads. We use the original softmax probabilities, without separately normalizing the attention map of each image.

Depth-based patch matching. Depth is used only to establish correspondences, not as an input to VGE. We retain depths between 0.1 and $10\,\mathrm{m}$ and require at least 25% valid depth pixels within a patch. To avoid patches spanning large depth discontinuities, we reject those whose 90th–10th percentile depth difference exceeds $0.10 + 0.05d\,\mathrm{m}$, where d is the patch's median depth in meters. For each remaining patch, we select a valid, median-consistent depth sample near its center, back-project it into 3D, and project it into another view using the original camera poses. A match must fall inside that image and satisfy

$$|z_{\mathrm{projected}} - z_{\mathrm{observed}}| \leq 0.05\,\mathrm{m} + 0.02 z_{\mathrm{projected}}.$$

We retain at most one source-patch match per destination patch. Unmatched or occluded regions are excluded rather than assigned zero attention.

Pose perturbations. For the rotation conditions, we independently sample a random axis in each camera's local coordinate system and rotate its orientation by 5° or 15°, keeping its position fixed. The two rotation magnitudes use the same sampled axes. For pose shuffling, we reassign complete poses among the group's unique views so that no image retains its original pose. Perturbations are applied after the standard pose normalization, without subsequent renormalization. Images, intrinsics, group membership, and reference patch matches remain unchanged across conditions.

Correlation and averaging. For each query and ordered pair of views, we compute Spearman's correlation between the attention values at matched patches, using average ranks for ties. We require at least 32 valid matches and exclude nearly constant attention vectors, for which the correlation is unreliable. Queries assigning less than 10^{-4} total attention to source-image patches are also excluded. Valid correlations are averaged first over view pairs for each query, then over queries within each group, and finally over groups within each scene. The reported score is the arithmetic mean of the resulting 50 scene scores, giving each scene equal weight.

A.3 Additional View Synthesis Results

Fig. 5 presents additional novel view synthesis results comparing AIMS with transformer-based baselines.

A.4 Additional attention map visualizations

Fig. 6 presents more visualizations on VGE attention maps for the given local view group.

A.5 Limitations and Future Work

AIMS maintains a fixed global decoding budget, but view grouping and local integration incur additional computation as the number of views increases. Moreover, compressing each group into a fixed number of tokens may discard fine detail in complex scenes. Future work could improve encoding efficiency and allocate scene tokens adaptively according to scene complexity.

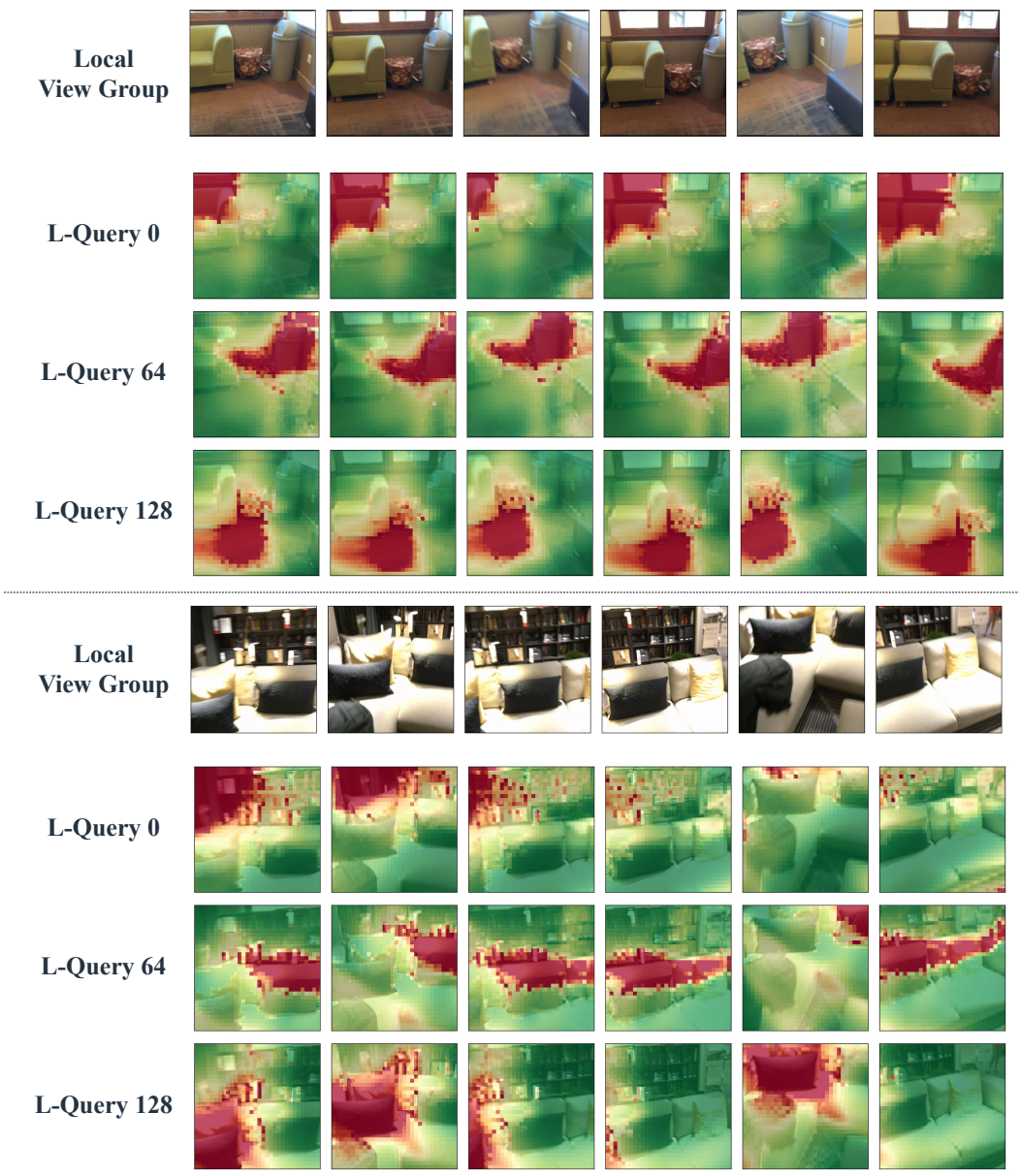


Figure 6: More visualizations on head-averaged attention maps at VGE layer 6 for learnable queries 0, 64, and 128 across a local view group.

Our grouping strategy relies on calibrated camera poses and does not explicitly account for visibility or image content. Nearby cameras may provide redundant observations, while fixed-size neighborhoods do not guarantee that every available view contributes to the scene representation. Combining pose coverage with appearance or visibility cues could improve view selection and integration.