ZEBRA3D: EFFICIENT HYBRID FOUNDATION MODELS FOR SCALABLE 3D RECONSTRUCTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Feed-forward 3D foundation models reconstruct scenes from unposed images, but the quadratic cost of global attention limits their scalability to large view sets. We present ZEBRA3D, a layerwise sparse–linear hybrid attention framework that improves the efficiency of pretrained 3D foundation models without training from scratch. Across VGGT, HunyuanWorld-Mirror, and π^3, we observe a broadly consistent depth-wise pattern: early stages are relatively tolerant to linear replacement, while sensitive middle-to-late stages tend to exhibit more concentrated attention. ZEBRA3D exploits this structure by assigning linear-centered attention to tolerant stages and tile-sparse attention to sensitive stages, using landmark summaries to preserve unselected context. Closed-form affine fitting enables training-free conversion, while optional post-conversion training combines task supervision with output and relational feature distillation. Across all three backbones, ZEBRA3D achieves strong reconstruction and camera-pose accuracy at high attention sparsity, with training-free variants remaining close to their dense counterparts. End-to-end speedups increase with input view count, reaching up to $7.1\times$ at 1,024 views. After post-conversion training, ZEBRA3D-VGGT reaches 80.5% RealEstate10K AUC@30 at 94% mean attention sparsity, within 1.1 percentage points of dense VGGT fine-tuned on the same data and schedule.

1 INTRODUCTION

Recovering scene geometry and camera poses from images is fundamental to 3D vision, supporting autonomous systems, immersive AR/VR (Cadena et al., 2016; Marchand et al., 2015), and providing geometric representations for 3D world modeling (HY-Team et al., 2026). Classical structure-from-motion (SfM) and multi-view stereo pipelines combine correspondence estimation, camera optimization, and dense reconstruction (Schonberger & Frahm, 2016; Schönberger et al., 2016). Neural scene representations such as NeRF require iterative per-scene optimization, typically from images with known camera poses (Mildenhall et al., 2021). The computational cost of scene-specific optimization motivates a shift toward feed-forward prediction. DUSt3R directly regresses dense point maps from image pairs, but reconstructing an image collection still requires global alignment (Wang et al., 2024). Recent feed-forward 3D (FF3D) foundation models, including VGGT (Wang et al., 2025), HunyuanWorld-Mirror (Liu et al., 2025), and π^3 (Wang et al., 2026c), jointly predict camera poses and dense scene geometry from multiple unposed views in a single forward pass, enabling reconstruction without per-scene optimization. These models alternate frame-wise attention with global attention for cross-view information exchange, allowing each image to benefit from observations elsewhere in the scene.

Eliminating per-scene optimization, however, does not eliminate the computational bottleneck: dense global attention scales quadratically with the number of input views. In our measurements, global attention accounts for 88% of VGGT's end-to-end runtime at 512 views (Figure 2a). Existing acceleration methods reduce this cost by merging image tokens, subsampling key/value context, or replacing dense attention with sparse or linear operators (Shen et al., 2026; Sun et al., 2026; Zhang et al., 2026a; Wang et al., 2026a). Among attention replacements, sparse and linear attention offer complementary ways to reduce this cost. Sparse attention preserves token-level softmax interactions over selected regions, whereas linear attention aggregates global context at linear cost. Applying either operator uniformly, however, does not account for differences in how pretrained stages respond

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.

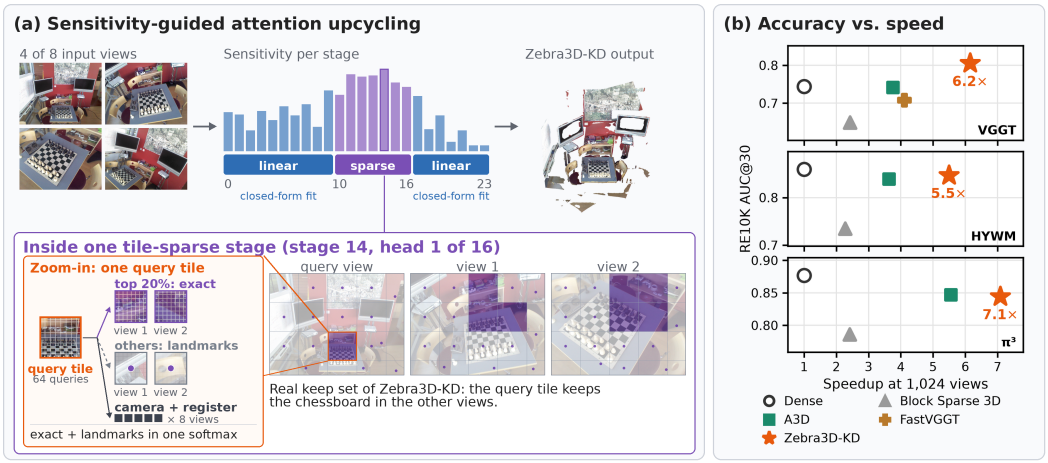


Figure 1: **Zebra3D overview.** **(a)** Sensitivity-guided assignment of linear and tile-sparse attention. Selected tiles retain full softmax interactions; omitted tiles become mean-key/value landmarks. **(b)** RE10K pose AUC@30 versus end-to-end speedup at 1,024 views.

to approximation. This motivates a central question: can the pretrained model itself reveal which stages tolerate linear replacement and which require more faithful softmax interactions?

We investigate this question by profiling every global-attention stage in pretrained VGGT, HunyuanWorld-Mirror (HYWM), and π^3 using two independent diagnostics: sensitivity to linear conversion and attention concentration. Across all three backbones, we observe a recurring depth-wise structural pattern: the middle stages are especially sensitive to linear replacement, yet their attention is highly concentrated. In the preceding and following stages, attention mass is distributed across many tokens, and these stages are surprisingly tolerant to linear approximation. This alignment turns an apparent trade-off into a common allocation principle: use sparse attention where concentrated interactions matter and linear attention everywhere else.

Building on this insight, we introduce ZEBRA3D, a sensitivity-guided framework for upcycling pretrained dense FF3D models (Figure 1). For conversion-tolerant stages, we introduce *linear-centered attention*, which reuses pretrained projections and applies a centered affine correction fitted to the dense operator by a closed-form least-squares solve per stage. For sensitive stages, *tile-sparse attention* evaluates selected spatial tiles with softmax and summarizes omitted tiles with landmarks to preserve global context. Together, the two operators replace the entire dense global-attention stack. The converted model supports immediate use without gradient updates. To further improve the quality–efficiency trade-off, we introduce a lightweight refinement which combines task supervision with distillation of outputs, centered features, and sampled cross-view relations from the frozen dense teacher.

Across three backbones and multiple benchmarks, ZEBRA3D provides strong accuracy–efficiency trade-offs. At 79% mean attention sparsity, training-free variants remain within 0.4–0.7 percentage points of dense RealEstate10K pose AUC@30 and achieve the highest CO3Dv2 AUC@30 among the evaluated non-dense methods for each backbone. At 1,024 views, ZEBRA3D delivers end-to-end speedups of 6.2×, 5.5×, and 7.1× on VGGT, HYWM, and π^3, respectively. After 3,000 refinement steps, ZEBRA3D -VGGT achieves 80.5% RealEstate10K pose AUC@30 at 94% mean attention sparsity, within 1.1 percentage points of dense VGGT fine-tuned on the same data and training schedule.

Our contributions are:

- An empirical analysis of depth-wise sensitivity to linear replacement, showing that sensitive stages tend to exhibit more concentrated attention, and a sensitivity-guided strategy that assigns linear attention to tolerant stages and sparse attention to sensitive ones.
- A pretrained-model conversion framework combining linear-centered attention and tile-sparse attention with landmark summaries. Closed-form affine fitting enables training-

free conversion, while optional post-conversion training incorporates output and relational feature distillation.

- Validation across three 3D backbones, demonstrating strong accuracy at high attention sparsity and 5.5–$7.1\times$ end-to-end speedups at $1{,}024$ views with packed Triton kernels.

2 KEY RELATED WORKS

Token and context compression. FastVGGT, LiteVGGT, HTTM, and Co-Me reduce token redundancy through saliency-, geometry-, temporal-, or confidence-guided merging (Shen et al., 2026; Shu et al., 2026; Wang et al., 2026b; Chen et al., 2026). AVGGT combines early frame attention with key/value subsampling (Sun et al., 2026), FlashVGGT compresses views into descriptor tokens (Wang & Xu, 2026), and RegimeVGGT combines merging and key/value downsampling across depth (You et al., 2026).

Efficient attention and representations. Block-Sparse VGGT evaluates selected attention blocks (Wang et al., 2026a), Lite3R distills sparse linear attention (Zhang et al., 2026a), and Speed3R combines compressed context with selected full-resolution interactions (Ren et al., 2026). VGG-T^3 instead adapts a fixed-size scene MLP at test time (Elflein et al., 2026). ZEBRA3D uses measured conversion sensitivity to assign linear and tile-sparse attention to different stages of a pretrained model. Detailed related works can be found in Appendix A.2.

3 METHOD

ZEBRA3D converts pretrained FF3D models into sparse–linear hybrids, assigning tile-sparse attention to stages sensitive to linearization and linear-centered attention to tolerant stages (Figure 2b,c). Here, stages denote global-attention layers and are indexed independently of interleaved frame-attention layers. We describe the allocation, both operators, and relational distillation below.

3.1 SENSITIVITY-GUIDED STAGE ALLOCATION

Measuring sensitivity. We replace one global-attention stage ℓ at a time with the fitted linear-centered operator described in Section 3.2, keeping all other stages unchanged. For each output modality u, we measure the deviation from the dense teacher over a validation set $\mathcal{D}$:

$$e_\ell^{(u)} = \frac{1}{|\mathcal{D}|} \sum_{x \in \mathcal{D}} d_u\Big(\widehat{y}_\ell^{(u)}(x), y_T^{(u)}(x)\Big), \qquad r_\ell = \frac{1}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} \mathrm{rank}_\ell\Big(e_\ell^{(u)}\Big), \tag{1}$$

where d_u is the modality-specific discrepancy and lower ranks indicate smaller deviations. We average ranks rather than raw errors to combine modalities with different units and scales. On VGGT, the output deviations from linear replacement peak in stages 11–15 and fall off toward both ends (Figure 2c).

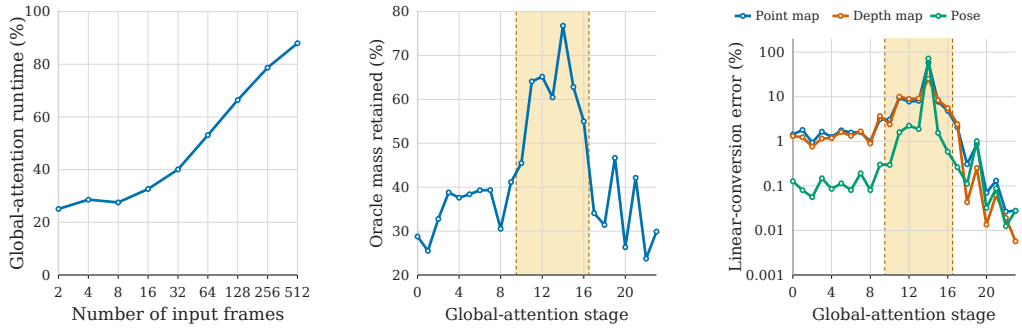
We measure concentration by the dense-attention mass retained by oracle selection of the top 20% of 64-token key/value blocks. Our key observation is that *stages sensitive to linear conversion retain more attention mass, whereas tolerant stages approach uniform attention*, supporting sparse and linear attention, respectively (Figures 2b and 3). This pattern extends to HYWM and π^3 (Figure 8).

Choosing the allocation. For a target linear fraction p, we assign the $K = \mathrm{round}(pL)$ stages with the lowest sensitivity scores r_ℓ to linear attention and the remaining stages to tile-sparse attention. Although this ranking does not account for interactions between simultaneous replacements, Table 3(d) shows that this allocation works well in practice and achieves good performance.

3.2 LINEAR-CENTERED ATTENTION WITH CLOSED-FORM INITIALIZATION

In conversion-tolerant stages, we replace softmax attention with a linear operator, reusing the pretrained QKV projections, positional encoding, normalization, and output projection from pretrained model. For one head of width d_h, let ϕ apply `softmax` along the channel dimension. We compute

$$O_{\mathrm{lin}} = \frac{\phi(Q)\left(\phi(K)^\top V\right)}{\phi(Q)\left(\phi(K)^\top \mathbf{1}\right) + \epsilon}, \tag{2}$$



(a) Global-attention runtime. **(b)** Oracle top-20% block mass. **(c)** Per-stage teacher deviation.

Figure 2: **Measurements motivating the Zebra3D design.** (a) Global attention reaches 88.0% of VGGT runtime at 512 views. (b) Attention Mass retained by an oracle Top-20% key/value blocks. (c) Relative output deviation after converting one global-attention stage to linear attention.

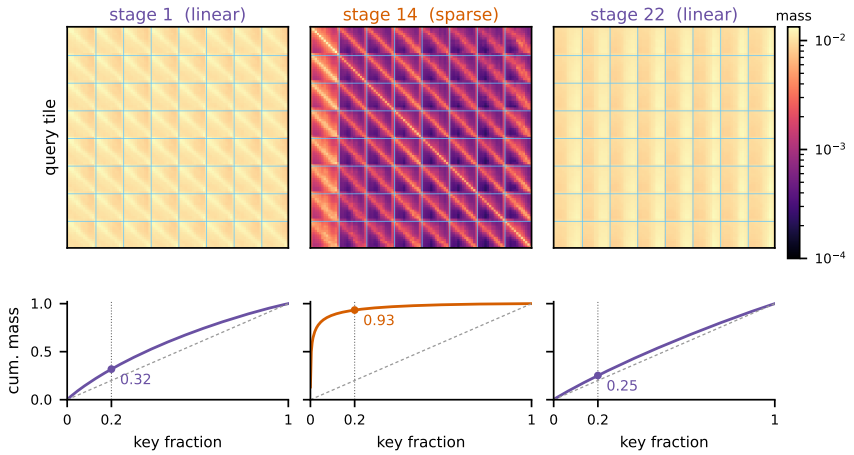


Figure 3: **Attention feature map across depth (VGGT).** Top: attention mass on one calibration scene, with cyan lines at view boundaries. Bottom: cumulative attention mass, with the value at a 20% key budget marked. A linear tolerant stage (1, 22) exhibits more diffuse attention while stage in the sensitive band (14) concentrates on few tokens. More visualizations in Figure 9

with row-wise division. Linear attention avoids quadratic compute of softmax attention and requires $O(Nd_h^2)$ computation. Further, to compensate for the mismatch with pretrained softmax attention, we fit an affine map (P_ℓ) and apply it to the residual around the value mean:

$$\overline{v} = \frac{1}{N}\mathbf{1}^\top V, \qquad \overline{V} = \mathbf{1}\overline{v}, \qquad \widetilde{O}_\ell = \overline{V} + (O_{\text{lin}} - \overline{V})P_\ell + \mathbf{1}b_\ell^\top. \tag{3}$$

Centering around mean of values decomposes the output into the global value mean and token-specific residuals. Uniform attention reduces exactly to value-mean pooling, so applying the fitted projection only to the residuals anchors the approximation at this natural baseline while correcting fine-grained token differences. This performs better than fitting an affine map directly to the uncentered output. When $P_\ell = I$ and $b_\ell = 0$, we recover raw linear attention; when $P_\ell = 0$ and $b_\ell = 0$, we recover value-mean pooling.

Closed-form fitting. Using frozen-teacher activations from a calibration dataset, we pool tokens across scenes and heads and form $X_\ell = [O_{\text{lin}} - \overline{V}, \mathbf{1}]$ and $Y_\ell = O_{\text{softmax}} - \overline{V}$. With $\Theta_\ell = [P_\ell; b_\ell^\top]$, ridge regression gives

$$\Theta_\ell^* = \arg\min_\Theta \|X_\ell\Theta - Y_\ell\|_F^2 + \lambda_{\text{fit}}\|\Theta\|_F^2 = \left(X_\ell^\top X_\ell + \lambda_{\text{fit}} I\right)^{-1} X_\ell^\top Y_\ell. \tag{4}$$

One solve per converted stage initializes the correction without backpropagation or updates to inherited weights. Fitting uses 13 calibration scenes and a float64 solve (Appendix B.3).

3.3 TILE-SPARSE ATTENTION WITH A LANDMARK TAIL

Stages sensitive to linear conversion retain token-level softmax interactions through tile-sparse attention (Figure 1a). We reshape each view's patch tokens onto their 2D image grid and partition the grid into non-overlapping 8×8 tiles ($B = 64$ tokens). Each tile therefore represents a contiguous image region, unlike blocks of consecutive positions in the flattened multi-view sequence, which may not form a coherent spatial neighborhood.

Selecting exact interactions. For each head, let a index a query tile and b a key tile, with mean query $\bar{\mathbf{q}}_a$ and mean key $\bar{\mathbf{k}}_b$. We score tile pairs and select the highest-scoring key tiles:

$$r_{ab} = \frac{\overline{q}_a^\top \overline{k}_b}{\sqrt{d_h}}, \qquad \mathcal{S}_a = \mathrm{TopK}_b(r_{ab}; \rho), \tag{5}$$

where the retention ratio ρ sets the tile budget k. Queries in tile a share $\mathcal{S}_a$, but evaluate individual query–key scores within the selected tiles.

Preserving unselected context. Each unselected tile contributes a landmark consisting of its mean key $\bar{\mathbf{k}}_b$, mean value $\bar{\mathbf{v}}_b$, and valid patch-token count n_b. For patch query i in tile $a(i)$, let $\mathcal{E}_i$ contain the selected patch keys, all special-token keys, and its self key. The mean-landmark operator is:

$$\begin{aligned} Z_i &= \sum_{j \in \mathcal{E}_i} e^{s_{ij}} + \sum_{b \notin \mathcal{S}_{a(i)}} n_b e^{\overline{s}_{ib}}, \\ \widetilde{o}_i &= \frac{\sum_{j \in \mathcal{E}_i} e^{s_{ij}} v_j + \sum_{b \notin \mathcal{S}_{a(i)}} n_b e^{\overline{s}_{ib}} \overline{v}_b}{Z_i}, \end{aligned} \tag{6}$$

where $s_{ij} = q_i^\top k_j / \sqrt{d_h}$ and $\overline{s}_{ib} = q_i^\top \overline{k}_b / \sqrt{d_h}$. The occupancy factor n_b, implemented as a $\log n_b$ logit offset, accounts for the number of summarized tokens. Selected tokens and landmarks share one softmax normalization, preserving detailed interactions in selected regions and compressed context elsewhere. Special-token queries attend densely. If a self key lies in an unselected tile, its explicit contribution supplements that tile's unchanged landmark.

Computational cost. Ignoring boundary and special-token terms, selected tiles require approximately ρN^2 token-pair evaluations, landmarks require $(1-\rho)N^2/B$ query–landmark evaluations, and tile scoring uses $(N/B)^2$ descriptor pairs. Sparse stages substantially reduce the cost of global attention at fixed ρ and B. We implement a packed tile-sparse attention kernel in Triton for efficient implementation without materializing the dense attention matrix (Appendix B.4).

3.4 LIGHTWEIGHT REFINEMENT VIA KNOWLEDGE DISTILLATION

The converted model can be used without gradient updates. Optional refinement combines task supervision with knowledge distillation from the frozen dense model. To preserve cross-view feature relations, we apply additional supervision at sampled linear-centered stages. Patch features are mean-centered within each view and ℓ_2-normalized to obtain z. A cosine loss $\mathcal{L}_{\cos}$ aligns corresponding student and teacher tokens, while a relational loss matches cross-view similarities:

$$\mathcal{L}_{\mathrm{rel}} = \mathbb{E}_{(a,c)\in\mathcal{P}} \left[\mathrm{SmoothL1}\left(z_{S,a}^\top z_{S,c}, z_{T,a}^\top z_{T,c} \right) \right]. \tag{7}$$

Here, S and T denote student and teacher, and $\mathcal{P}$ contains token pairs (a, c) from different views. For each anchor a, candidates c include both teacher-selected high-similarity tokens and randomly sampled tokens from other views. Stage and pair sampling avoid constructing dense relation matrices.

Complementing these linear-stage losses, intermediate-feature distillation $\mathcal{L}_1$ supervises backbone-specific feature taps, including tile-sparse stages, while output distillation $\mathcal{L}_{\mathrm{out}}$ matches final predictions shared by student and teacher. Both use teacher-magnitude-normalized ℓ_1 matching. The refinement objective is

$$\begin{aligned} \mathcal{L} &= \mathcal{L}_{\mathrm{task}} + \lambda_{\mathrm{KD}}(t) \left(\mathcal{L}_{\mathrm{out}} + \mathcal{L}_{\mathrm{feat}} \right), \\ \mathcal{L}_{\mathrm{feat}} &= 0.25\mathcal{L}_1 + 0.50\mathcal{L}_{\cos} + 0.25\mathcal{L}_{\mathrm{rel}}, \end{aligned} \tag{8}$$

where $\mathcal{L}_{\mathrm{task}}$ provides backbone-specific geometry supervision. Refinement lasts 3,000 optimizer updates, with λ_{KD} decreasing from 1.0 to approximately 0.6. Distillation requires an additional frozen-teacher forward pass during training but adds no inference-time computation. Exact loss definitions, feature taps, sampling, and optimization settings are given in Appendices B.5 and B.6.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Models and training. We use released VGGT, HYWM, and π^3 checkpoints (Wang et al., 2025; Liu et al., 2025; Wang et al., 2026c). Training-free ZEBRA3D uses 13 held-out calibration scenes for closed-form projection fitting without gradient updates. ZEBRA3D-KD starts from the converted model and refines all student parameters for 3,000 optimizer updates using a frozen dense teacher (Section 3.4). All training runs use 20% subset of training data used in HunyuanWorld-Mirror training, with BF16 mixed precision on eight MI355X GPUs and at most 16 views per GPU microbatch.

Evaluation. We evaluate reconstruction on 7Scenes, NRGBD, and DTU (Shotton et al., 2013; Azinović et al., 2022; Aanæs et al., 2016), and camera pose on RealEstate10K and CO3Dv2 using ten-view inputs (Zhou et al., 2018; Reizenstein et al., 2021). Long-sequence reconstruction is evaluated on ScanNet-50 (Dai et al., 2017) with 100–1,000 views. All evaluation datasets are strictly excluded from the refinement training mixture except CO3Dv2, for which we use disjoint training and test splits. To evaluate reconstruction beyond densely sampled video sequences, we additionally evaluate wide-baseline reconstruction on Tanks and Temples and ETH3D (Knapitsch et al., 2017; Schops et al., 2017) at $N \in \{8, 16, 32, 64\}$ views. Timing uses compiled BF16 inference on one MI355X (Appendix B.10).

Baselines. We compare against the original dense backbones, FastVGGT (Shen et al., 2026), SAF3R (Deng et al., 2026), RegimeVGGT (You et al., 2026), and Lite3R (Zhang et al., 2026a). We additionally evaluate architecture-matched implementations of AVGGT-f4 (Sun et al., 2026) and Block-Sparse VGGT (Wang et al., 2026a), denoted A3D and Block Sparse 3D, respectively. In addition, we fine-tune the dense VGGT backbone on the same training data, for the same $3{,}000$ steps, and with the same ground-truth supervision used for refinement (Appendix B.6). We denote this control VGGT + GT-FT and report it alongside the dense backbones. Unsupported backbone–baseline combinations are omitted. Further details are in Appendix B.

Configuration notation. xL/yS uses linear attention in fraction x of stages and omits fraction y of key tiles in the remaining sparse stages. Thus, 0.5L/0.8S is half linear and retains the top 20% of tiles elsewhere, with landmarks summarizing omitted tiles. Reported mean sparsity also accounts for landmarks, protected tokens, and linear stages (Appendix B.11).

4.2 MAIN RESULTS

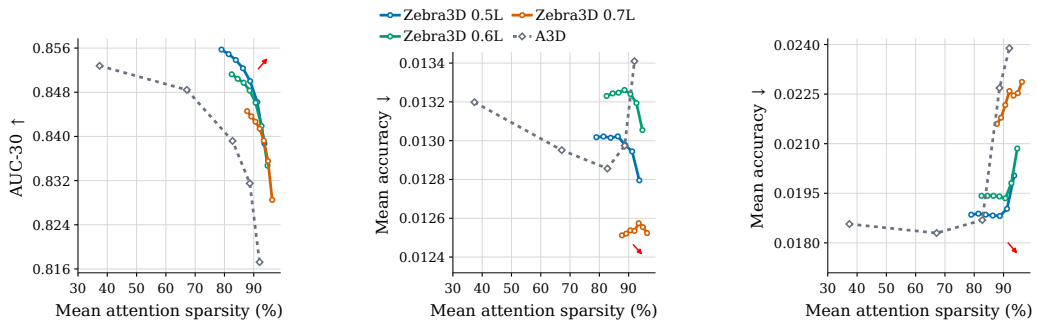
ZEBRA3D provides configurable quality–speed trade-offs through the linear-stage fraction and sparse-stage retention budget. The conservative 0.5L/0.6S configuration prioritizes accuracy retention, whereas the more aggressive 0.7L/0.8S configuration delivers the largest measured speedups among our evaluated settings.

Training-free conversion. We evaluate ZEBRA3D across three backbones on reconstruction and camera-pose benchmarks, shown in Table 1. At 0.5L/0.6S, training-free conversion retains Re10K AUC-30 within 0.007, 0.004, and 0.006 of dense VGGT, HYWM, and π^3, respectively, at approximately 79% mean attention sparsity. It also achieves the highest CO3Dv2 AUC-30 among the evaluated non-dense methods for each backbone: 0.897, 0.862, and 0.900. On 7Scenes, ZEBRA3D reaches mean accuracy errors at 0.017, 0.013, and 0.012 across the 3 models, matching the corresponding dense results at the reported precision. On NRGBD, the errors are 0.018, 0.019, and 0.014, compared with 0.016, 0.018, and 0.014 for the dense models. Figure 4 extends this comparison to a quality–sparsity sweep against A3D on HYWM.

Refinement and operating points. Refinement offers additional gains, although these depend on the backbone and benchmark. For VGGT at 0.5L/0.6S, ZEBRA3D-KD raises Re10K AUC-30 from 0.737 to 0.805 and reduces 7Scenes accuracy error from 0.017 to 0.015. The more aggressive 0.7L/0.8S configuration also reaches 0.805 at approximately 94% mean attention sparsity; Gains are benchmark-dependent: CO3Dv2 and NRGBD performance declines after refinement, with similar regressions in the dense VGGT+GT-FT control. At 0.5L/0.6S, the refined hybrid remains within 0.002 CO3Dv2 AUC-30 of this control and matches its NRGBD accuracy error at the reported precision.

Table 1: **Reconstruction and pose.** Point-map results on 7Scenes/NRGBD (stride 10) and pose on RE10K/CO3Dv2 (10 views). $\bar{s}_N$ denotes mean attention sparsity at N views. Bold/underline: best/second-best non-dense. VGGT + GT-FT is dense VGGT finetuned on the same data as Zebra3D-KD. Full metrics: Appendix E.1.

Method	Mean sparsity (%) ↑		7Scenes ↓		NRGBD ↓		RE10K ↑		CO3Dv2 ↑	
	$\bar{s}_{300}$	$\bar{s}_{10}$	Acc. mean	Comp. mean	Acc. mean	Comp. mean	AUC-30	RTA-5	AUC-3	AUC-30
VGGT	0	0	0.017	0.027	0.016	0.011	0.744	0.543	0.566	0.900
VGGT + GT-FT	0	0	0.024	0.031	0.024	0.018	0.816	0.674	0.474	0.885
FastVGGT	88	83	**0.015**	0.025	0.020	**0.013**	0.708	0.476	0.412	0.867
SAF3R	84	78	0.018	**0.022**	0.054	0.022	0.714	0.493	0.504	0.883
RegimeVGGT	90	87	0.016	0.026	0.024	0.016	0.722	0.498	0.445	0.876
Block Sparse 3D	70	70	0.017	0.025	0.042	0.016	0.649	0.379	0.207	0.835
Lite3R	70	70	0.039	0.023	0.068	0.018	0.578	0.253	0.078	0.779
A3D	83	75	0.016	0.026	0.025	0.017	0.742	0.536	0.468	0.881
Zebra3D (0.5L/0.8S)	89	89	0.018	0.027	0.021	0.014	0.730	0.507	0.535	0.892
Zebra3D (0.5L/0.6S)	79	79	0.017	0.027	**0.018**	**0.013**	0.737	0.526	**0.554**	**0.897**
Zebra3D-KD (0.5L/0.6S)	79	79	**0.015**	0.026	0.024	0.018	**0.805**	0.649	0.458	0.883
Zebra3D-KD (0.7L/0.8S)	**94**	**94**	0.016	0.025	0.025	0.019	**0.805**	**0.654**	0.403	0.872
HYWM	0	0	0.013	0.024	0.018	0.014	0.860	0.774	0.403	0.865
Block Sparse 3D	70	70	0.014	0.024	0.041	0.023	0.735	0.573	0.085	0.761
A3D	83	75	**0.013**	0.024	0.019	0.014	0.839	0.740	0.323	0.844
RegimeVGGT	90	87	**0.013**	0.024	0.019	0.014	0.840	0.741	0.315	0.843
Zebra3D (0.5L/0.8S)	89	89	**0.013**	0.024	0.019	**0.013**	0.850	0.758	0.376	0.854
Zebra3D (0.5L/0.6S)	79	79	**0.013**	0.025	0.019	0.014	**0.856**	**0.769**	**0.400**	**0.862**
Zebra3D-KD (0.5L/0.6S)	79	79	**0.013**	0.023	**0.018**	0.014	0.854	0.766	0.332	0.849
Zebra3D-KD (0.7L/0.8S)	**94**	**94**	0.014	**0.022**	0.019	0.014	0.847	0.752	0.295	0.838
π^3	0	0	0.012	0.017	0.014	0.009	0.877	0.809	0.582	0.903
Block Sparse 3D	70	70	0.020	**0.016**	0.049	0.017	0.786	0.655	0.213	0.835
A3D	88	82	**0.012**	**0.016**	0.018	0.010	0.847	0.765	0.457	0.880
Zebra3D (0.5L/0.8S)	89	89	**0.012**	0.018	0.015	0.010	0.866	0.792	0.542	0.895
Zebra3D (0.5L/0.6S)	79	79	**0.012**	0.018	**0.014**	**0.009**	**0.871**	**0.800**	**0.563**	**0.900**
Zebra3D (0.7L/0.8S)	**94**	**94**	0.021	0.020	0.043	0.015	0.836	0.726	0.256	0.841
Zebra3D-KD (0.5L/0.6S)	79	79	**0.012**	0.018	0.015	**0.009**	0.855	0.773	0.527	0.895
Zebra3D-KD (0.7L/0.8S)	**94**	**94**	0.013	0.018	0.017	0.010	0.844	0.751	0.440	0.876



(a) RE10K camera pose. **(b)** 7Scenes point maps. **(c)** NRGBD point maps.

Figure 4: **HYWM quality–sparsity trade-off.** Zebra3D varies sparse retention within fixed placements; A3D varies subsampling.

Long-sequence reconstruction. At 1,000 views, ZEBRA3D-KD (0.7L/0.6S) achieves the lowest reported Chamfer distance (CD) of 0.447, compared with 0.466 for dense VGGT, while achieving 3.54× speedup. The more aggressive 0.7L/0.8S variant achieves a Chamfer distance of 0.460 and a 5.6× speedup over dense VGGT, improving both measures over dense VGGT, FastVGGT, and RegimeVGGT at this sequence length. Finetuned dense VGGT (VGGT + GTFT in Table 2) increases CD to 0.503 at 1,000 views, showing performance regression. A3D provides a competitive intermediate trade-off (0.452 Chamfer distance; 3.83× speedup). Our 0.7L/0.6S configuration provides higher accuracy at a longer runtime, while 0.7L/0.8S provides faster inference at slightly lower accuracy. Figure 6 illustrates reconstruction quality at 1,000 views; additional examples and visualization details, including outdoor KITTI and Neural RGB-D rooms (Figure 12), are presented in appendix F.1.

Table 2: **ScanNet-50 long-sequence reconstruction.** Chamfer distance (CD; ↓) and end-to-end speedup. Bold/underline: best/second-best non-dense. VGGT + GT-FT is dense VGGT finetuned on the same data as Zebra3D-KD. See Appendix B.8 for the detailed evaluation protocol.

Method	1,000 frames		500 frames		300 frames		100 frames	
	CD ↓	Speedup ↑	CD ↓	Speedup ↑	CD ↓	Speedup ↑	CD ↓	Speedup ↑
VGGT	0.466	1.00	0.465	1.00	0.451	1.00	0.440	1.00
VGGT + GT-FT	0.503	1.00	0.496	1.00	0.486	1.00	0.474	1.00
FastVGGT	0.464	<u>3.97</u>	0.462	<u>3.40</u>	0.450	<u>2.89</u>	0.435	1.81
RegimeVGGT	0.478	3.50	0.468	3.03	0.453	2.59	0.439	1.70
A3D	0.452	3.83	0.445	3.38	0.438	**2.95**	<u>0.427</u>	**2.00**
Block Sparse 3D	0.471	2.33	0.464	2.11	0.465	1.90	0.446	1.42
Zebra3D-KD (0.5L/0.8S)	<u>0.449</u>	3.78	**0.443**	3.34	0.438	1.74	<u>0.427</u>	1.50
Zebra3D-KD (0.5L/0.6S)	0.451	2.30	<u>0.444</u>	2.20	<u>0.435</u>	1.38	0.430	1.30
Zebra3D-KD (0.7L/0.8S)	0.460	**5.60**	0.449	**4.59**	0.447	2.51	0.439	<u>1.86</u>
Zebra3D-KD (0.7L/0.6S)	**0.447**	3.54	0.447	3.21	**0.429**	2.02	**0.424**	1.67

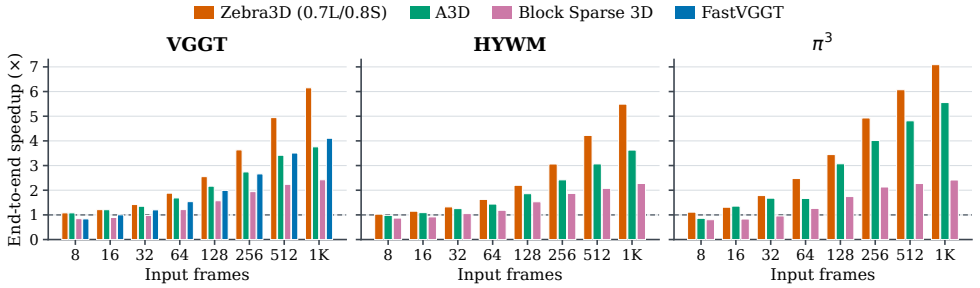


Figure 5: **End-to-end inference speedup relative to dense backbone.** At 1,024 frames, Zebra3D reaches 6.2× on VGGT, 5.5× on HYWM, and 7.1× on π^3, with gains increasing as global attention becomes the dominant cost. Speedups measured on a single MI355X GPU.

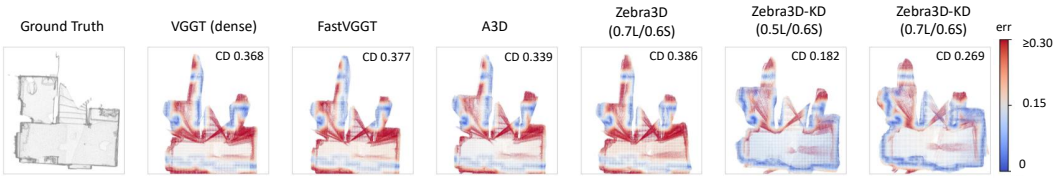


Figure 6: **Long-sequence reconstruction at** 1,000 **views on ScanNet-50.** Visualization of the views of the predicted point maps, coloured by distance to the ground-truth mesh.

Efficiency across backbones. Figure 5 extends the runtime comparison to all three backbones. At 1,024 views, ZEBRA3D (0.7L/0.8S) achieves 6.2×, 5.5×, and 7.1× end-to-end speedups over dense VGGT, HYWM, and π^3, respectively, and exceeds A3D and Block Sparse 3D, so its higher mean attention sparsity carries through to end-to-end latency.

Additional datasets and models. Across Tanks and Temples, ETH3D, and DTU at $N \in \{8, 16, 32, 64\}$, all 0.5L configurations remain within 3% of their dense backbones in NC1 and outperform A3D (Appendix E.3). We also compare with Fast3R, MUSt3R, and Spann3R (Yang et al., 2025a; Cabon et al., 2025; Wang & Agapito, 2025) in Appendix A.3. ZEBRA3D outperforms these models on point-map and camera-pose benchmarks and surpasses Fast3R and Spann3R in both ScanNet quality and speed.

4.3 ABLATION STUDIES

Layer-role placement. At a fixed 0.5L/0.8S budget, Table 3(d) shows that sensitivity-guided placement achieves the best pose AUC-30 on all three backbones, while uniform, random, and reverse-

Table 3: **Ablations.** **(a)** Linear-attention initialization; **(b)** sparse attention with landmark summaries; **(c)** KD objectives for VGGT; **(d)** layer placement at fixed budget. Additional ablations are reported in Appendix D.

(a) Linear attention (0.7L/0.8S) on π^3

Attention choice	RE10K AUC-30 ↑	7S Acc. ↓	NRGBD Acc. ↓
Frame attention	0.655	0.031	0.051
Linear, no fit	0.789	0.028	0.071
Linear, uncentered fit	0.811	0.024	0.065
Linear-centered + fit	**0.836**	**0.021**	**0.043**

(b) Sparse attention (0.5L/0.75S) on π^3

Attention choice	RE10K AUC-30 ↑	7S Acc. ↓	NRGBD Acc. ↓
SGA	0.847	**0.011**	0.017
Block sparse	0.808	0.024	0.058
Tile sparse	0.811	0.027	0.066
Block sparse + landmark	0.864	0.013	0.024
Tile sparse + landmark	**0.868**	0.012	**0.014**

(c) Knowledge distillation (VGGT at 0.5L/0.6S)

Training objective	7S Acc. ↓	NRGBD Acc. ↓	RE10K AUC-30 ↑
GT only	0.023	0.046	0.777
KD only	0.023	0.045	0.716
GT + output KD	0.023	0.041	0.736
GT + feature KD	0.029	0.047	0.756
RelKD	**0.015**	**0.024**	**0.805**

(d) Layer placement at fixed 0.5L/0.8S budget

Placement	VGGT	HYWM	π^3
Back-sparse	0.590	0.609	0.864
Front-sparse	0.208	0.200	0.033
Uniform	0.115	0.142	0.148
Reverse-sensitivity	0.114	0.137	0.033
Random, mean of 3 seeds	0.318	0.216	0.063
Sensitivity-guided	**0.730**	**0.850**	**0.866**

sensitivity assignments degrade substantially. Back-sparse nearly matches only on π^3 (0.864 vs. 0.866), because sensitive band for π^3 extends to the final stage.

Attention operators. Table 3(a,b) ablates the linear initialization and sparse attention operators on π^3. Centered affine fitting lifts Re10K AUC-30 to 0.836 (0.789 without fitting, 0.811 with an uncentered fit) and gives the best AUC-30 in all six backbone–assignment groups (Appendix D.1). Adding landmarks to tile-sparse attention raises AUC-30 from 0.811 to 0.868 and reduces Neural RGB-D accuracy error from 0.066 to 0.014, outperforming A3D's subsampled global attention (SGA) on both. Appendix D covers the other backbones and trained mixers.

Distillation. Table 3(c) compares refinement objectives at a fixed training budget. KD alone underperforms GT-only training on Re10K (0.716 vs. 0.777). The complete objective improves all three metrics over GT-only training, reaching 0.805 AUC-30 and reducing 7Scenes and Neural RGB-D accuracy errors from 0.023 to 0.015 and from 0.046 to 0.024. Full results are in (Appendix D.4).

Calibration-set stability. We see that across five different calibration datasets, the assigned layer rankings remain relatively stable. The Spearman correlation ρ between layer rankings ranges from 0.843–0.886. The resulting layer assignments give 81.5–88.9% of stages the same role as ours at 0.5L and 93.5–97.2% at 0.7L. Therefor, a fixed allocation needs no per-dataset recalibration (Appendix C.3).

5 Conclusion

We introduced ZEBRA3D , a sensitivity-guided framework for upcycling pretrained 3D reconstruction transformers into sparse–linear hybrids. Across VGGT, HYWM, and π^3, we identified depth-dependent sensitivity to linear replacement, with more sensitive stages tending to exhibit more concentrated attention. This observation motivates assigning tile-sparse attention to sensitive stages and linear-centered attention to tolerant stages, while landmark summaries preserve unselected context. Closed-form affine fitting enables training-free conversion, with optional post-conversion training incorporating task supervision and output and relational feature distillation. ZEBRA3D offers quality–efficiency trade-offs at 79–94% mean attention sparsity: conservative configurations preserve accuracy close to dense models, while more aggressive configurations prioritize speed. At 0.7L/0.8S, it delivers respective end-to-end speedups of $6.2\times$, $5.5\times$, and $7.1\times$ at $1{,}024$ views across the three backbones. Input-adaptive allocation and broader hardware support remain future work.

AI USE STATEMENT

Generative AI tools were used to assist with language editing and the preparation and debugging of code and tables. AI is used to write implementation details in depth to aid in reproduction by coding agents. The authors reviewed and verified all AI-assisted content and take full responsibility for the paper's claims, results, and presentation.

ETHICS STATEMENT

This work uses existing pretrained models and standard research datasets and does not involve human-subject experiments or the collection of new personal data. Scalable 3D reconstruction can nevertheless enable privacy-sensitive or surveillance applications. Deployments should respect consent, dataset licenses, privacy, and applicable legal requirements.

REPRODUCIBILITY STATEMENT

The paper and appendix specify the model conversions, layer assignments, calibration and training procedures, evaluation protocols, and ablations needed to reproduce the reported results. We will publicly release the code, configurations, custom kernels, and trained checkpoints upon publication.

REFERENCES

Henrik Aanæs, Rasmus Ramsbøl Jensen, George Vogiatzis, Engin Tola, and Anders Bjorholm Dahl. Large-scale data for multiple-view stereopsis. *International Journal of Computer Vision*, 120(2): 153–168, 2016.

Eduardo Arnold, Jamie Wynn, Sara Vicente, Guillermo Garcia-Hernando, Aron Monszpart, Victor Prisacariu, Daniyar Turmukhambetov, and Eric Brachmann. Map-free visual relocalization: Metric pose relative to a single image. In *European Conference on Computer Vision*, pp. 690–708. Springer, 2022.

Dejan Azinović, Ricardo Martin-Brualla, Dan B Goldman, Matthias Nießner, and Justus Thies. Neural rgb-d surface reconstruction. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6280–6291. IEEE, 2022.

Gilad Baruch, Zhuoyuan Chen, Afshin Dehghan, Tal Dimry, Yuri Feigin, Peter Fu, Thomas Gebauer, Brandon Joffe, Daniel Kurz, Arik Schwartz, et al. Arkitscenes: A diverse real-world dataset for 3d indoor scene understanding using mobile rgb-d data. *arXiv preprint arXiv:2111.08897*, 2021.

Yohann Cabon, Lucas Stoffl, Leonid Antsfeld, Gabriela Csurka, Boris Chidlovskii, Jerome Revaud, and Vincent Leroy. Must3r: Multi-view network for stereo 3d reconstruction. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1050–1060. IEEE, 2025.

Cesar Cadena, Luca Carlone, Henry Carrillo, Yasir Latif, Davide Scaramuzza, José Neira, Ian Reid, and John J Leonard. Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Transactions on robotics*, 32(6):1309–1332, 2016.

Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niessner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3d: Learning from rgb-d data in indoor environments. *arXiv preprint arXiv:1709.06158*, 2017.

Yutian Chen, Yuheng Qiu, Ruogu Li, Jay Patrikar, and Sebastian Scherer. Co-me: Confidence guided token merging for visual geometric transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14590–14599, 2026.

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.

Tri Dao and Albert Gu. Transformers are ssms: Generalized models and efficient algorithms through structured state space duality. *arXiv preprint arXiv:2405.21060*, 2024.

Jianing Deng, Yuanzhe Li, Jialu Wang, Song Wang, Tianlong Chen, Huanrui Yang, and Jingtong Hu. Saf3r: Dynamic sparse attention for feed-forward 3d reconstruction transformers. In *European Conference on Computer Vision*, pp. 42–59. Springer, 2026.

Sven Elflein, Ruilong Li, Sérgio Agostinho, Zan Gojcic, Laura Leal-Taixé, Qunjie Zhou, and Aljosa Osep. VGG-T^3: Offline feed-forward 3d reconstruction at scale. *arXiv preprint arXiv:2602.23361*, 2026.

Po-Han Huang, Kevin Matzen, Johannes Kopf, Narendra Ahuja, and Jia-Bin Huang. Deepmvs: Learning multi-view stereopsis. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2821–2830. IEEE, 2018.

HY HY-Team, Chenjie Cao, Xuhui Zuo, Zhenwei Wang, Yisu Zhang, Junta Wu, Zhenyang Liu, Yuning Gong, Yang Liu, Bo Yuan, et al. Hy-world 2.0: A multi-modal world model for reconstructing, generating, and simulating 3d worlds. *arXiv e-prints*, pp. arXiv–2604, 2026.

Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics (ToG)*, 36(4):1–13, 2017.

Zhengqi Li and Noah Snavely. Megadepth: Learning single-view depth prediction from internet photos. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2041–2050. IEEE, 2018.

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

Yifan Liu, Zhiyuan Min, Zhenwei Wang, Junta Wu, Tengfei Wang, Yixuan Yuan, Yawei Luo, and Chunchao Guo. Worldmirror: Universal 3d world reconstruction with any-prior prompting. *arXiv preprint arXiv:2510.10726*, 2025.

Eric Marchand, Hideaki Uchiyama, and Fabien Spindler. Pose estimation for augmented reality: a hands-on survey. *IEEE transactions on visualization and computer graphics*, 22(12):2633–2651, 2015.

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.

Jeremy Reizenstein, Roman Shapovalov, Philipp Henzler, Luca Sbordone, Patrick Labatut, and David Novotny. Common objects in 3d: Large-scale learning and evaluation of real-life 3d category reconstruction. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 10881–10891. IEEE, 2021.

Weining Ren, Xiao Tan, and Kai Han. Speed3r: Sparse feed-forward 3d reconstruction models. *arXiv preprint arXiv:2603.08055*, 2026.

Mike Roberts, Jason Ramapuram, Anurag Ranjan, Atulit Kumar, Miguel Angel Bautista, Nathan Paczan, Russ Webb, and Joshua M Susskind. Hypersim: A photorealistic synthetic dataset for holistic indoor scene understanding. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 10892–10902. IEEE, 2021.

Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4104–4113, 2016.

Johannes L Schönberger, Enliang Zheng, Jan-Michael Frahm, and Marc Pollefeys. Pixelwise view selection for unstructured multi-view stereo. In *European conference on computer vision*, pp. 501–518. Springer, 2016.

Thomas Schops, Johannes L Schonberger, Silvano Galliani, Torsten Sattler, Konrad Schindler, Marc Pollefeys, and Andreas Geiger. A multi-view stereo benchmark with high-resolution images and multi-camera videos. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3260–3269, 2017.

You Shen, Zhipeng Zhang, Yansong Qu, Xiawu Zheng, Jiayi Ji, Shengchuan Zhang, and Liujuan Cao. Fastvggt: Fast visual geometry transformer. In *The Fourteenth International Conference on Learning Representations*, 2026.

Jamie Shotton, Ben Glocker, Christopher Zach, Shahram Izadi, Antonio Criminisi, and Andrew Fitzgibbon. Scene coordinate regression forests for camera relocalization in rgb-d images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2930–2937, 2013.

Zhijian Shu, Cheng Lin, Tao Xie, Wei Yin, Ben Li, Zhiyuan Pu, Weize Li, Yao Yao, Xun Cao, Xiaoyang Guo, et al. Litevggt: Boosting vanilla vggt via geometry-aware cached token merging. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 36422–36432, 2026.

Xianbing Sun, Zhikai Zhu, Zhengyu Lou, Bo Yang, Jinyang Tang, Liqing Zhang, He Wang, and Jianfu Zhang. Avggt: Rethinking global attention for accelerating vggt. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 251–260, 2026.

Fabio Tosi, Yiyi Liao, Carolin Schmitt, and Andreas Geiger. Smd-nets: Stereo mixture density networks. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8938–8948. IEEE, 2021.

Chung-Shien Brian Wang, Christian Schmidt, Jens Piekenbrinck, and Bastian Leibe. Block-sparse global attention for efficient multi-view geometry transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14546–14555, 2026a.

Hengyi Wang and Lourdes Agapito. 3d reconstruction with spatial memory. In *2025 International Conference on 3D Vision (3DV)*, pp. 78–89. IEEE, 2025.

Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5294–5306. IEEE, 2025.

Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20697–20709. IEEE, 2024.

Weitian Wang, Lukas Meiner, Rai Shubham, Cecilia De La Parra, and Akash Kumar. Httm: Head-wise temporal token merging for faster vggt. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 26379–26388, 2026b.

Wenshan Wang, Delong Zhu, Xiangwei Wang, Yaoyu Hu, Yuheng Qiu, Chen Wang, Yafei Hu, Ashish Kapoor, and Sebastian Scherer. Tartanair: A dataset to push the limits of visual slam. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4909–4916. IEEE, 2020.

Yifan Wang, Jianjun Zhou, Haoyi Zhu, Wenzheng Chang, Yang Zhou, Zizun Li, Junyi Chen, Jiangmiao Pang, Chunhua Shen, and Tong He. pi^3: Permutation-equivariant visual geometry learning. In *International Conference on Learning Representations*, volume 2026, pp. 10481–10497, 2026c.

Zipeng Wang and Dan Xu. Flashvggt: Efficient and scalable visual geometry transformers with compressed descriptor attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21826–21835, 2026.

Hongchi Xia, Yang Fu, Sifei Liu, and Xiaolong Wang. Rgbd objects in the wild: Scaling real-world 3d object learning from rgb-d videos. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22378–22389. IEEE, 2024.

Jianing Yang, Alexander Sax, Kevin J Liang, Mikael Henaff, Hao Tang, Ang Cao, Joyce Chai, Franziska Meier, and Matt Feiszli. Fast3r: Towards 3d reconstruction of 1000+ images in one forward pass. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21924–21935. IEEE, 2025a.

Songlin Yang, Jan Kautz, and Ali Hatamizadeh. Gated delta networks: Improving mamba2 with delta rule. In *International Conference on Learning Representations*, volume 2025, pp. 29687–29707, 2025b.

Yao Yao, Zixin Luo, Shiwei Li, Jingyang Zhang, Yufan Ren, Lei Zhou, Tian Fang, and Long Quan. Blendedmvs: A large-scale dataset for generalized multi-view stereo networks. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1787–1796. IEEE, 2020.

Jinhao You, Shuo Lyu, Zhuohang Lyu, Tanxuan Li, Zibo Zhao, Jiaxiang Hu, Kai Tang, and Yichen Guo. Regimevggt: Layer-wise spatially preserving redundancy removal for visual geometry grounded transformer. *arXiv preprint arXiv:2606.18439*, 2026.

Haoyu Zhang, Zeyu Zhang, Zedong Zhou, Yang Zhao, and Hao Tang. Lite3r: A model-agnostic framework for efficient feed-forward 3d reconstruction. *arXiv preprint arXiv:2605.11354*, 2026a.

Jintao Zhang, Haoxu Wang, Kai Jiang, Shuo Yang, Kaiwen Zheng, Haocheng Xi, Ziteng Wang, Hongzhou Zhu, Min Zhao, Ion Stoica, et al. Sla: Beyond sparsity in diffusion transformers via fine-tunable sparse–linear attention. In *International Conference on Learning Representations*, volume 2026, pp. 138288–138305, 2026b.

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

A Additional Background and Related Work

A.1 Inference in 3D Reconstruction Models

Feed-forward 3D reconstruction models infer scene geometry directly from a set of images. VGGT, HunyuanWorld-Mirror (HYWM), and π^3 differ in their tokens and prediction heads, but follow the same broad computation (Wang et al., 2025; Liu et al., 2025; Wang et al., 2026c). A visual encoder, typically DINOv2, converts each input view into patch embeddings, which are concatenated with any model-specific tokens:

$$\boldsymbol{T}_i = E_\theta(\boldsymbol{I}_i) \in \mathbb{R}^{P_i \times d}, \qquad \boldsymbol{X}^0 = \mathrm{Concat}_{i=1}^{F}[\boldsymbol{T}_i; \boldsymbol{S}_i], \tag{9}$$

where $\boldsymbol{S}_i$ denotes camera, register, or geometric-conditioning tokens and $N = \sum_i (P_i + |\boldsymbol{S}_i|)$ is the sequence length. The resulting tokens pass through an aggregator that alternates within-view processing and scene-level mixing:

$$\boldsymbol{X}^{\ell+\frac{1}{2}} = \mathrm{FrameAttn}_\ell(\boldsymbol{X}^\ell), \qquad \boldsymbol{X}^{\ell+1} = \mathrm{GlobalAttn}_\ell(\boldsymbol{X}^{\ell+\frac{1}{2}}), \tag{10}$$

Task-specific heads then decode camera parameters and dense outputs such as depth, point maps, normals, and tracking features. VGGT appends one camera token and four register tokens to every view; π^3 removes reference-specific embeddings to preserve permutation equivariance; and World-Mirror can encode camera, intrinsic, and depth priors.

Global attention is the only stage in this pipeline that allows unrestricted communication across all views. For head h of width d_h,

$$\boldsymbol{Q}_h = \boldsymbol{X}\boldsymbol{W}_h^Q, \qquad \boldsymbol{K}_h = \boldsymbol{X}\boldsymbol{W}_h^K, \qquad \boldsymbol{V}_h = \boldsymbol{X}\boldsymbol{W}_h^V, \tag{11}$$

and multi-head global attention is

$$\boldsymbol{A}_h = \mathrm{softmax}\left(\frac{\boldsymbol{Q}_h \boldsymbol{K}_h^\top}{\sqrt{d_h}}\right), \qquad \mathrm{GlobalAttn}(\boldsymbol{X}) = \mathrm{Concat}_{h=1}^{H}(\boldsymbol{A}_h \boldsymbol{V}_h)\boldsymbol{W}^O, \tag{12}$$

with a row-wise softmax. The dense matrix $\boldsymbol{A}_h \in \mathbb{R}^{N \times N}$ supports long-range correspondence and global geometric consistency, but costs $\mathcal{O}(N^2 d)$ computation. At fixed image resolution, compute grows quadratically with the number of views. Prior work identifies this step as the main obstacle to scaling feed-forward reconstruction (Shen et al., 2026; Sun et al., 2026; Wang et al., 2026a;b).

A.2 Extended Related Work

Pairwise and direct multi-view reconstruction. DUSt3R regresses dense point maps from image pairs without requiring known camera calibration or poses, but reconstructing an image collection requires selecting pairs and aligning their local predictions in a common frame (Wang et al., 2024). The quadratic growth in candidate pairs and the subsequent global optimization limit scaling to long sequences. MUSt3R extends this paradigm to symmetric multi-view prediction in a shared coordinate frame and introduces a multi-layer memory mechanism for scalable offline and online reconstruction (Cabon et al., 2025). Fast3R also generalizes pairwise reconstruction to unordered image sets: a fusion transformer processes all views jointly and predicts local and global point maps in one forward pass, avoiding external global alignment (Yang et al., 2025a). Its train-short, test-long design and distributed inference enable sequences exceeding $1{,}000$ images. These methods redesign the reconstruction architecture or inference pathway for direct multi-view processing; ZEBRA3D instead upcycles existing feed-forward 3D backbones by replacing their dense global-attention operators.

Token reduction. Token merging reduces both dimensions of the attention matrix while leaving softmax attention itself unchanged. FastVGGT merges redundant tokens while protecting the reference view, salient regions, and spatial coverage (Shen et al., 2026). Its safeguards reflect a central difficulty in 3D reconstruction: tokens that appear redundant locally may still carry cross-view evidence needed for correspondence or camera estimation. LiteVGGT therefore selects geometry-aware anchors and caches merge assignments across layers, reducing repeated selection overhead while maintaining spatial structure (Shu et al., 2026). HTTM performs temporal merging separately for each attention head so that head-specific interaction patterns are not forced into one shared token partition (Wang et al., 2026b). Co-Me instead learns a confidence predictor and preferentially merges low-confidence tokens (Chen et al., 2026). These methods reduce both query and context lengths, so their savings propagate through subsequent layers, but merging is lossy and changes the token set consumed by downstream prediction heads.

Key/value and context compression. Another strategy keeps every query but compresses the context. We refer to AVGGT and its architecture-matched HYWM and π^3 variants collectively as A3D. A3D replaces early global layers with frame attention, then spatially subsamples patch keys and values while preserving queries and special tokens (Sun et al., 2026). This preserves a prediction for every query token but reduces the evidence available to each query. FlashVGGT summarizes each frame with descriptor tokens and uses cross-attention from image tokens to those descriptors; caching descriptors enables chunk-recursive inference without reprocessing the complete history (Wang & Xu, 2026). RegimeVGGT combines saliency-aware merging with protected key/value downsampling and varies compression across depth, acknowledging that redundancy is not uniform through the network (You et al., 2026). Speed3R similarly combines a compressed context with selected full-resolution interactions (Ren et al., 2026). Context-compression methods preserve the query grid, but their sampling or summary policy determines which cross-view evidence remains individually accessible.

Sparse and linear replacements. The closest methods replace global attention itself. We refer to Block-Sparse VGGT and its architecture-matched variants as Block Sparse 3D. Block Sparse 3D uses sparse kernels to evaluate adaptively selected query–key blocks, exploiting the concentration of attention around geometric correspondences (Wang et al., 2026a). Selected blocks retain exact softmax interactions, while unselected blocks make no contribution. SAF3R introduces dynamic sparse attention with head-specific sparse-pattern assignments (Deng et al., 2026). Lite3R instead combines sparse linear attention with partial attention distillation, replacing quadratic interactions with a model-agnostic efficient operator (Zhang et al., 2026a). VGG-T^3 takes a different route: it replaces the variable-length scene representation with a fixed-size MLP adapted at test time, yielding linear scaling in the number of views (Elflein et al., 2026).

ZEBRA3D differs along both operator selection and information preservation. Rather than applying one approximation throughout the stack or prescribing a fixed schedule, it profiles the pretrained model and assigns linear or tile-sparse attention according to measured stage sensitivity. Tile-sparse stages preserve exact softmax interactions for selected spatial regions and represent every omitted region with a landmark under the same normalization. Linear-centered stages retain the inherited

projections and use a closed-form fit, enabling training-free conversion. Optional relational distillation then refines the same hybrid architecture without changing its stage assignment.

A.3 Comparison with Direct Multi-View Reconstruction Models

Tables 4 and 5 compare ZEBRA3D with three architectures designed for direct multi-view reconstruction. MUSt3R predicts views in a shared coordinate frame using a multi-layer memory (Cabon et al., 2025); Spann3R incrementally maintains an external spatial memory (Wang & Agapito, 2025); and Fast3R jointly processes an unordered image set in one forward pass (Yang et al., 2025a).

MUSt3R provides the highest long-sequence speedup but substantially higher reconstruction error. At 1,000 frames, ZEBRA3D-KD 0.7L/0.8S reaches $5.60\times$ speedup with CD 0.460, while the 0.7L/0.6S configuration achieves the best CD (0.447) at $3.54\times$. Fast3R and Spann3R improve accuracy over MUSt3R but remain less accurate than dense VGGT and both ZEBRA3D-KD operating points.

Table 5 extends the comparison to the main reconstruction and pose benchmarks. Every ZEBRA3D 0.5L/0.6S backbone outperforms the three direct multi-view models on all reported 7Scenes and NRGBD point-map metrics and on all CO3Dv2 AUC thresholds. On RE10K, ZEBRA3D-HYWM and ZEBRA3D-Pi3 also exceed MUSt3R in AUC-30 and RTA-30, while MUSt3R remains competitive in RRA-30. Zebra3D-π^3 gives the strongest overall pose results, reaching 0.871 RE10K AUC-30 and 0.900 CO3Dv2 AUC-30.

A.4 ScanNet-50 Quality–Speed Trade-off

Figure 7 visualizes the 1,000-frame results from Table 2, augmented with Fast3R and Spann3R from Table 4. The two ZEBRA3D-KD 0.7L configurations define complementary operating points: 0.7L/0.6S achieves the lowest Chamfer distance (0.447) at $3.54\times$ speedup, while 0.7L/0.8S reaches $5.60\times$ speedup with a Chamfer distance of 0.460.

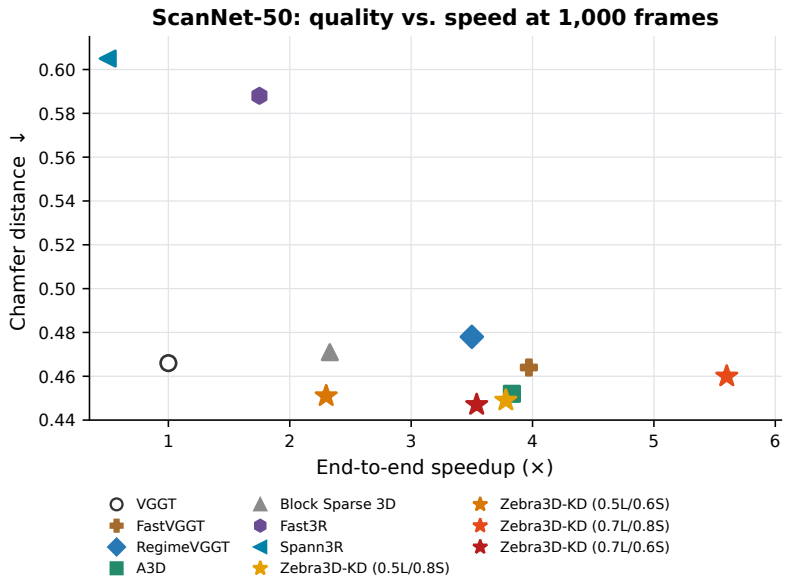


Figure 7: **ScanNet-50 quality–speed trade-off at** $1{,}000$ **frames.** Chamfer distance versus end-to-end speedup relative to dense VGGT. Lower Chamfer distance and higher speedup are preferred.

Table 4: **Direct multi-view reconstruction on ScanNet-50.** Chamfer distance (CD; ↓) and end-to-end speedup relative to dense VGGT on one MI355X. Bold/underline denote the best/second-best non-dense results. Spann3R* was evaluated without `torch.compile`.

Method	1,000 frames		500 frames		300 frames		100 frames	
	CD ↓	Speedup ↑	CD ↓	Speedup ↑	CD ↓	Speedup ↑	CD ↓	Speedup ↑
VGGT	0.466	1.00	0.465	1.00	0.451	1.00	0.440	1.00
MUSt3R	0.958	**15.7**	0.937	**7.07**	0.922	**3.85**	0.897	0.91
Spann3R*	0.605	0.50	0.604	0.27	0.588	0.18	0.567	0.09
Fast3R	0.588	1.75	0.558	1.64	0.547	1.52	0.524	<u>1.38</u>
Zebra3D-KD (0.7L/0.8S)	<u>0.460</u>	<u>5.60</u>	<u>0.449</u>	<u>4.59</u>	<u>0.447</u>	<u>2.51</u>	<u>0.439</u>	**1.86**
Zebra3D-KD (0.7L/0.6S)	**0.447**	3.54	**0.447**	3.21	**0.429**	2.02	**0.424**	1.67

Table 5: **Direct multi-view comparison on the main benchmarks.** Point-map accuracy, completeness, and NC1 on 7Scenes and NRGBD; camera-pose metrics on RE10K and CO3Dv2. Accuracy and completeness are lower-is-better; all other metrics are higher-is-better. Zebra3D uses the training-free 0.5L/0.6S configuration for each backbone.

Method	7Scenes			NRGBD			RE10K			CO3Dv2			
	Acc. ↓	Comp. ↓	NC1 ↑	Acc. ↓	Comp. ↓	NC1 ↑	AUC-30 ↑	RRA-30 ↑	RTA-30 ↑	AUC-3 ↑	AUC-5 ↑	AUC-15 ↑	AUC-30 ↑
Fast3R	0.035	0.043	0.522	0.074	0.030	0.539	0.605	0.984	0.825	0.226	0.357	0.642	0.767
MUSt3R	0.045	0.321	0.547	0.070	0.127	0.654	0.796	1.000	0.934	0.356	0.496	0.742	0.839
Spann3R	0.124	0.110	0.553	0.248	0.173	0.578	0.292	0.991	0.542	0.006	0.025	0.208	0.412
Zebra3D-VGGT (0.5L, 0.6S)	0.017	0.027	0.606	0.018	0.013	0.746	0.737	0.999	0.919	0.554	0.665	0.834	0.897
Zebra3D-HYWM (0.5L, 0.6S)	0.013	0.025	0.608	0.019	0.014	0.744	0.856	0.999	0.953	0.400	0.541	0.774	0.862
Zebra3D-Pi3 (0.5L, 0.6S)	0.012	0.018	0.606	0.014	0.009	0.731	0.871	1.000	0.955	0.563	0.673	0.839	0.900

B Implementation and Evaluation Details

B.1 Backbones and conversion

We use the released checkpoints `facebook/VGGT-1B` (Wang et al., 2025), `tencent/HunyuanWorld-Mirror` (Liu et al., 2025), and `yyfz233/Pi3` (Wang et al., 2026c). Conversion preserves every compatible pretrained parameter, including the input encoder, QKV and output projections, normalization layers, positional encoding, and prediction heads. We rank global stages by the single-stage linear-conversion error described in Equation 1, then assign the requested fraction to linear-centered attention; the remaining stages use tile-sparse attention. Table 6 lists the resulting zero-based 0.5L and 0.7L assignments, which match the released configurations. Every global-attention stage is replaced by either linear-centered or tile-sparse attention.

In the xL/yS shorthand of Section 4, the selector retains a key-tile fraction $\rho = 1 - y$; the four reported operating points are

Shorthand	x	y	$\rho{=}1-y$
0.5L/0.6S	0.5	0.6	0.4
0.5L/0.8S	0.5	0.8	0.2
0.7L/0.6S	0.7	0.6	0.4
0.7L/0.8S	0.7	0.8	0.2

B.2 Baselines

The dense baselines are the unmodified released VGGT, HYWM, and π^3 models (Wang et al., 2025; Liu et al., 2025; Wang et al., 2026c). We evaluate FastVGGT at its `m0` operating point (Shen et al., 2026) and SAF3R at its `t4` operating point using the published head-to-pattern assignments (Deng et al., 2026). A3D denotes AVGGT-f4 on VGGT and the same frame-attention/key–value subsampling operator ported without architectural changes to HYWM and π^3 (Sun et al., 2026). A3D converts the first global stages to frame attention and runs the rest as subsampled global attention (SGA): every query is kept, but it attends only to the special tokens, the intact reference frame, a spa-

Table 6: Layer-role assignments for the all-efficient Zebra3D variants.

Backbone	Variant	Linear-centered stages	Tile-sparse stages
VGGT	0.5L	1–6, 8, 18, 20–23	0, 7, 9–17, 19
	0.7L	0–9, 17–23	10–16
HYWM	0.5L	1–2, 4–6, 8–9, 18, 20–23	0, 3, 7, 10–17, 19
	0.7L	0–10, 17–18, 20–23	11–16, 19
π^3	0.5L	0–2, 4–9	3, 10–17
	0.7L	0–10, 16–17	11–15

tially subsampled grid of patch keys and values from the other frames, and one mean-fill token that averages the dropped keys and values. AVGGT releases no code, so A3D is our reimplementation of the published f4 setting; the same operator is applied unchanged to HYWM and π^3. Block Sparse 3D similarly denotes the published adaptive block-sparse operator and our architecture-matched ports at the sp70 operating point (Wang et al., 2026a). We evaluate RegimeVGGT at its released canonical AB2 setting on VGGT and, because HYWM has the same 24-stage alternating aggregator, apply the operator unchanged to HYWM; the latter is an architecture-matched extension not reported by the RegimeVGGT paper (You et al., 2026). Lite3R is evaluated from its released configuration (Zhang et al., 2026a). We report a method only on backbones that its released implementation or our architecture-matched operator supports. Every baseline is evaluated through the same data loaders, frame selections, alignment, and metric code as ZEBRA3D .

B.3 LINEAR INITIALIZATION

For each linear stage, we first run the frozen dense model on 13 held-out multi-view calibration scenes. Teacher softmax outputs and converted linear outputs are accumulated across scenes, tokens, heads, and views. We then fit the centered affine correction in Equation 4 by a regularized least-squares solve in float64. The fit requires one teacher forward pass and one solve per converted stage; it does not update inherited weights by gradient descent. These fitted projections, together with copied dense-model weights, define the training-free Zebra3D models.

B.4 TRITON IMPLEMENTATION OF HYBRID ATTENTION

We implement Zebra3D's linear-centered and tile-sparse operators in Triton to reduce intermediate memory traffic and execute their computations efficiently on the GPU. For linear-centered attention, we fuse Q/K normalization, positional rotation, and channel-softmax feature mapping into the kernels that accumulate the linear-attention state and compute the readout. This avoids materializing intermediate normalized Q/K tensors between separate preprocessing and attention kernels.

Sparse stages partition each image into 8×8-patch tiles, or 64 tokens per full tile. We pack the full key/value sequence into tile-major order before selecting key blocks independently for each head and query block. The retention parameter is ρ=0.2 or 0.4, corresponding to y=0.8 or 0.6. Unselected packed blocks contribute one key–value landmark each; special-token keys remain visible, special-token query rows retain dense attention, and every patch query retains its self key. Training and evaluation use the packed `triton` implementation with 64-token query and key blocks, head-level selection, one view per selection group.

The operator first constructs a permutation from the model's view-major token layout to tile-major order. Complete spatial tiles occupy aligned 64-token blocks; tiles truncated at the image boundary are concatenated in a ragged region and may share or span packed blocks. Camera and register tokens are appended as special tokens. The permutation, its inverse, the tile-to-block map, and special-token indices are cached for each frame count and patch grid. This avoids rebuilding layout-dependent index tensors and repeating the device synchronizations needed to construct them. At inference, a fused packing launch makes Q contiguous and permutation-packs K and V together. The attention kernel can then read repeatedly used K/V blocks with coalesced memory accesses, while query and output rows are mapped through the cached permutation.

A fused statistics kernel reads BF16 Q, K, and V and accumulates in FP32. For each packed block and head, it emits the mean query and key used by the selector, a mean key–value landmark over non-special tokens, and the log occupancy. These blocks coincide with spatial tiles in the aligned region but can combine boundary-tile tokens in the ragged region. Each query block ranks key blocks by the mean-query/mean-key dot product and stores the selected indices together with a chosen-block mask. The selection count is $\lceil \rho N/64 \rceil$, capped by the number of eligible non-special blocks and kept at least one when an eligible block exists.

The main Triton program implements FlashAttention-style online softmax. It streams four contributions without constructing an $N \times N$ score matrix: non-special tokens in selected key blocks, every special-token key, the query's own key when its block was not selected, and one occupancy-weighted landmark for each unselected block. The landmark logit receives a $\log n_b$ offset, where n_b is the number of valid non-special tokens represented by that landmark. The chosen-block mask excludes selected blocks from the landmark tail, while special-token masks exclude special keys from the selected-block stream. When an explicit self key is added, it is not subtracted from its block's landmark summary. Softmax state and output accumulators remain in FP32; BF16 matrix products use the accelerator's matrix instructions. The kernel works in base-two exponent units, folding $\log_2 e$ into the attention scale and using the hardware `exp2` instruction. The shipped forward launch uses two warps, two pipeline stages, and a 64-column landmark tile. Special query rows are evaluated exactly by a separate dense-row kernel and written into the same output tensor. This kernel distributes key tiles across workers and combines their partial softmax results after rescaling to a common maximum. On supported GPU layouts, it uses four warps, two pipeline stages, and explicit base-two exponential evaluation while retaining attention over every key.

Training uses a custom autograd bridge for BF16 $[B, H, N, 64]$ $Q/K/V$, 64-token blocks, per-head selection, and a landmark tail. The forward retains the selected-block indices, chosen masks, landmark statistics, output, and row-wise log-sum-exp. Separate Triton kernels compute dQ and the owner-accumulated dK/dV in FP32, followed by dedicated gradient kernels for exact special-query rows; gradients are cast back to the input dtype at the boundary. The sparse call is an explicit compiler graph break so that raw Triton launches are not captured or rewritten, while the surrounding backbone remains compilable. The reported 3,000-step KD runs use this autograd bridge, whereas inference calls the forward kernel directly.

B.5 Relational knowledge distillation

All ZEBRA3D-KD models are initialized from the corresponding training-free 0.5L or 0.7L conversion (Table 6), using copied dense weights and the float64 linear-centered projection fit (Appendix B.3). Each student is refined for 3,000 optimizer steps with all parameters trainable. The original unconverted checkpoint serves as a frozen FP32 teacher, while the student uses BF16 mixed precision.

Distillation loss definitions. For the output modalities shared by teacher and student, we use

$$\mathcal{L}_{\text{out}} = \sum_{u \in \mathcal{U}} \alpha_u \frac{\left\| M_u \odot (y_S^{(u)} - y_T^{(u)}) \right\|_1}{\left\| M_u \odot y_T^{(u)} \right\|_1 + \epsilon}, \tag{13}$$

where M_u masks invalid entries. Each geometric-map term has weight 1, and each included camera term has weight 0.5; the terms available for each backbone are listed below. Intermediate feature matching uses

$$\mathcal{L}_1 = \frac{1}{K} \sum_{k=1}^{K} \frac{\|h_S^{(k)} - h_T^{(k)}\|_1}{\|h_T^{(k)}\|_1 + \epsilon}. \tag{14}$$

For cosine and relational matching at sampled linear-centered stages, we first remove special tokens and normalize patch features separately for each view:

$$\overline{h}_v = \frac{1}{T} \sum_{j=1}^{T} h_{v,j}, \qquad z_{v,i} = \frac{h_{v,i} - \overline{h}_v}{\|h_{v,i} - \overline{h}_v\|_2 + \epsilon}. \tag{15}$$

The cosine term is $\mathcal{L}_{\cos} = \mathbb{E}_{v,i}[1 - z_{S,v,i}^\top z_{T,v,i}]$. Both cosine and relational losses are averaged over the sampled stages.

Output matching. Equation 13 is evaluated only for modalities emitted by both models; missing heads are omitted. For HYWM, output KD matches `pts3d`, `depth`, and `normals` with unit weight and the 9D pose encoding `camera_params` with weight 0.5. For VGGT, it matches `point_map` and `depth_map` with unit weight and `camera_params`, `extrinsic`, and `intrinsic`, each with weight 0.5; the latter two are decoded from `camera_params`. For π^3, it matches `point_map` and `depth_map` with unit weight and the 3×4 world-to-camera `extrinsic` with weight 0.5; `camera_params` and `intrinsic` are absent.

Feature taps. For HYWM, feature-L1 matching (Equation 14) uses the DPT `token_list` at `intermediate_idxs` $4, 11, 17, 23$. For VGGT, it uses all 24 global-block outputs; for π^3, it uses all 18 global-stage outputs, including tile-sparse stages. All feature-L1 taps are evaluated on every microbatch.

Centered cosine and relational matching instead use only the linear-centered stages specified in Table 6. At optimizer step t, we sample $m = 4$ such stages without replacement using a deterministic permutation seeded by $10007 + t$. Both losses are averaged over the sampled stages; tile-sparse stages are never candidates. For π^3, global stage i corresponds to decoder block $2i + 1$. The four accumulated microbatches share the same step-seeded stage subset, while token pairs are resampled on each microbatch.

Cross-view pair sampling. At each sampled stage, features are reshaped to $[B, V, T, d]$, camera and register tokens are removed, and patch features are normalized as in Equation 15. For each batch item and sampled stage, we sample $A = 192$ patch-token anchors. For each anchor, $P = 64$ candidate indices are sampled independently. A same-view draw is deterministically remapped by one view stride, and the teacher retains the $k = 8$ most similar candidates from this pool. We additionally draw $R = 24$ candidates by the same procedure, without deduplicating them against the top-k set. Equation 7 averages Smooth-L1 over the resulting 32 pairs per anchor. The teacher selects the high-similarity candidates, and teacher and student relations are evaluated on identical token pairs. We use PyTorch Smooth-L1 with transition parameter $\beta_{\text{Huber}} = 1$; if $V < 2$, the relation term is zero. We do not distill attention matrices, query/key alignments, or tile-selection scores.

Distillation schedule. The distillation weight decays linearly from 1.0 to 0.6 over the $3{,}000$ optimizer updates:

$$\lambda_{\text{KD}}(t) = 1.0 - 0.4\,\frac{t}{3000}, \qquad 0 \le t < 3000, \tag{16}$$

where t counts optimizer updates.

B.6 Training data and optimization

Data and augmentation. All KD runs use the same 20% preprocessed subset of the HunyuanWorld-Mirror training mixture. The constituent datasets and their sampling weights are: HyperSim (Roberts et al., 2021) (400), TartanAir (Wang et al., 2020) (300), Matterport3D (Chang et al., 2017) (250), DL3DV-10K (Ling et al., 2024) (250), CO3D training split (Reizenstein et al., 2021) (250), BlendedMVS (Yao et al., 2020) (200), UnrealStereo4K (Tosi et al., 2021) (150), MVS-Synth (Huang et al., 2018) (150), WildRGB-D (Xia et al., 2024) (100), ARKitScenes (Baruch et al., 2021) (200), MapFree (Arnold et al., 2022) (150), and MegaDepth (Li & Snavely, 2018) (150). Each GPU receives up to 16 views per microbatch. Images have aspect ratios in $[0.5, 2.0]$ and resolutions of 100k–250k pixels. Spatial cropping and photometric augmentation follow the common multi-view data pipeline. Color jitter is applied with probability 0.9, with brightness, contrast, and saturation magnitudes of 0.5 and hue magnitude of 0.1; the pipeline also includes random grayscale augmentation.

Calibration scenes. The reference stage assignment and the closed-form projection fit use 13 multi-view scenes shipped with the HunyuanWorld-Mirror examples. No gradient update is run on them, and none of them enters a reported metric table. Appendix C.3 repeats the sensitivity profiling on 7Scenes, DTU, RealEstate10K, NRGBD, and CO3Dv2 and compares those rankings with this fixed reference. It does not change the assignment.

Task supervision. Task supervision in Equation 8 is backbone-specific, because the three backbones do not predict the same quantities.

HYWM and VGGT use the shared camera, point-map, and depth objectives with top-level weights 5.0, 1.0, and 1.0, respectively. These weights are inherited unchanged from the released HYWM and VGGT training configurations rather than tuned for this work. In raw magnitude the camera term is 4–5$\times$ smaller than the point term, so the factor of 5.0 aligns the two contributions rather than prioritizing pose: over early training the weighted camera and point gradients stay within 1.06–$1.18\times$ of each other, and this ratio is nearly identical for the dense and hybrid runs, so the weighting does not confound the dense–hybrid comparisons of Section 4. We depart from the released VGGT configuration in one respect: it disables the point loss by default, whereas we enable all three terms. Point and depth losses use $\gamma = 1$, $\alpha = 0.2$, and a valid-range setting of 0.98. The point loss uses a normal-based gradient penalty, while the depth loss uses a spatial-gradient penalty. Normal-map and Gaussian-splat rendering task losses are disabled; for HYWM, normal predictions nevertheless remain in output distillation with unit weight, and the normal head is not frozen.

For π^3 we use its own published objective, because it predicts local point maps and camera-to-world transforms rather than the world-coordinate points, nine-vector pose encoding, and depth confidence that the shared losses expect. Local point maps are supervised by an inverse-depth-weighted L1 against local ground truth after a scale $s^\star$ obtained by a detached weighted alignment on $4{,}096$ resampled valid points. A geometric normal term built from local point neighborhoods is enabled on the high- and mid-quality datasets of the mixture. Cameras are supervised on all relative pose pairs, using a Huber term on translations scaled by the same $s^\star$ and a rotation-angle term; the effective weights are 1 on the point terms, 10 on translations, and 0.1 on rotations. Confidence supervision is disabled, as are the shared camera, point, and depth losses.

Optimization. Training uses eight AMD MI355X GPUs with DistributedDataParallel, BF16 student mixed precision, synchronized batch normalization, and four-step gradient accumulation. Each GPU processes at most 16 views per microbatch; the dynamic sequence length sets the sequence batch size to $\lfloor 16/V \rfloor$. Thus 512 views per optimizer update is an upper bound attained only when every GPU microbatch fills the 16-view budget. Training sequences contain at most 16 views; evaluations on longer sequences test generalization beyond this training sequence length.

We use AdamW with $(\beta_1, \beta_2) = (0.9, 0.95)$, $\epsilon = 10^{-8}$, weight decay 0.1, and gradient-norm clipping at 1.0. The optimizer's `pretrained`, `backbone`, and `new` parameter groups use peak learning rates 10^{-5}, 3×10^{-5}, and 10^{-4}, respectively. The `new` group contains proj_ℓ and any mixer-only tensors. All groups are trainable. A 250-step linear warmup is followed by cosine decay to 10% of each group's peak learning rate by step $3{,}000$. We use random seed 42, save checkpoints every 500 steps, and perform no in-loop validation. Jobs run in the ROCm environment with Triton 3.7.1.

Fine-tuned dense control. The VGGT + GT-FT reference reported in the main tables is produced by this same procedure applied to the unconverted dense VGGT checkpoint: the same 20% HunyuanWorld-Mirror subset, the same augmentation pipeline, the same $3{,}000$ steps with 250-step warmup and cosine decay, the same optimizer groups and peak learning rates, and the same seed. The only differences are that no attention operator is replaced and that the objective contains the task term alone, with all distillation terms disabled; no frozen teacher is instantiated. This control therefore matches the Zebra3D-KD training budget and supervision exactly and differs only in architecture and in the presence of the relational terms, which is what makes it usable both as a supervision-matched reference for the main tables and as the dense counterpart of the GT-only row in Table 3(c).

B.7 Evaluation datasets

All indoor point-map evaluations use 518×378 inputs and fixed sequence maps. For 7Scenes (Shotton et al., 2013) and NRGBD (Azinović et al., 2022), stride 10 selects every tenth frame; the maps contain 18 and 9 test sequences, respectively. Depth, validity masks, intrinsics, and camera poses are read from each benchmark's official representation before the shared crop-and-resize transform.

RealEstate10K (Zhou et al., 2018) pose evaluation uses a fixed seed-42 map containing 1,719 test clips and ten frames per clip at 518×378. CO3Dv2 (Reizenstein et al., 2021) follows the VGGT ten-frame protocol: sequences are selected with NumPy seed 0 from 41 categories, images are resized to width 518 with aspect-preserving center crop and white padding, and results are macro-averaged over categories.

B.8 ScanNet-50 long-sequence protocol

Table 2 follows the FastVGGT ScanNet-50 setting: the same 50 scenes and the official frame-index maps at 1,000, 500, 300, and 100 frames. Images are resized to width 518 with the height snapped to the patch size of 14, which gives 518×392 inputs for ScanNet. Chamfer distance is computed against the ground-truth mesh after voxel downsampling at 0.05, clipping at 0.5, and sampling 100k points, and is normalized by the bounding-box diagonal. For each scene we take the median over repeated forwards, then the mean over the 50 scenes. Chamfer distance uses these 518×392 inputs. Speedup is measured once per mixer configuration at 518×378 with the harness in Appendix B.10, and the same number is reported for the refined and training-free students that share that configuration. Figure 5 uses this harness at 1,024 frames.

B.9 Metrics and alignment

For point maps, we first estimate a similarity transform with Umeyama alignment on valid corresponding pixels and refine it with ICP. We then report prediction-to-ground-truth distance as accuracy, ground-truth-to-prediction distance as completeness, and normal consistency in the accuracy direction as NC1; both means and medians are computed per scene before dataset aggregation. For RealEstate10K, relative rotation and translation accuracies are computed over all frame pairs at thresholds of 5°, 15°, and 30°. AUC-30 integrates the cumulative distribution of the larger rotation or translation error up to 30°. CO3Dv2 uses the same relative-pose construction and reports AUC at 3°, 5°, 15°, and 30°. Pose arithmetic and metric accumulation use float64.

B.10 Runtime measurement

Each end-to-end measurement uses one MI355X with BF16 inference and standard Inductor compilation, with graph breaks and eager fallback enabled. Each method and frame count is measured in a fresh process with a separate compiler cache. We use five warmup runs and 20 device-event measurements, except at exactly 1,024 frames, where we use two warmup runs and five measurements. We report compute speedup by dividing the separately measured dense median by the method median for the same backbone and frame count. The measured call includes the model forward pass and all enabled geometry-prediction heads, but excludes input loading and preprocessing. HYWM Gaussian-splat outputs are disabled for every method, and no teacher computation is present at inference.

B.11 Mean Attention Sparsity

Let F be the number of frames, N the total number of tokens, and C_ℓ the number of query–key pairs evaluated by layer ℓ. We define the mean attention sparsity over L layers as

$$\bar{s} = 1 - \frac{1}{L}\sum_{\ell=1}^{L}\frac{C_\ell}{N^2}. \tag{17}$$

Thus, dense, frame, and linear attention have sparsities 0, $1 - 1/F$, and 1, respectively. Special-token terms may be omitted in a simplified patch-dominated approximation when their contribution is negligible; all numerical examples and performance–sparsity plots reported here retain them.

For Zebra3D , a sparse layer evaluates the keys in the selected spatial tiles, one landmark for each unselected tile, all special-token keys, and an unselected query's self key; special-token query rows remain dense. If $\mathcal{Q}_{\mathrm{sp}}$ is the set of special-token queries and $\mathcal{K}_\ell(q)$ is the resulting key set for query q, then

$$C_\ell^{\mathrm{Zebra}} = |\mathcal{Q}_{\mathrm{sp}}|N + \sum_{q\notin\mathcal{Q}_{\mathrm{sp}}}|\mathcal{K}_\ell(q)|, \qquad \bar{s}_{\mathrm{Zebra}} = 1 - \frac{1}{L}\sum_{\ell=1}^{L}\frac{C_\ell^{\mathrm{Zebra}}}{N^2}. \tag{18}$$

For A3D, let E of its D global stages use global-to-frame attention, and let V_σ be the number of keys retained by an SGA stage at subsampling factor σ. Here V_σ includes all special-token keys, the intact first frame when used, the subsampled patch grid, and the mean-fill token. Its mean sparsity is

$$\bar{s}_{\mathrm{A3D}} = 1 - \frac{1}{D}\left(\frac{E}{F} + (D-E)\frac{V_\sigma}{N}\right). \tag{19}$$

For RegimeVGGT, let M_ℓ be the number of query tokens remaining after the layer-ℓ merge and K_ℓ the number of those tokens retained on the key–value axis by its phase-shifted grid, including the reference-frame anchor and all special tokens. Merge-only middle layers have $K_\ell = M_\ell$; unmerging occurs after attention and therefore adds no query–key pairs. Hence

$$C_\ell^{\mathrm{Regime}} = M_\ell K_\ell, \qquad \bar{s}_{\mathrm{Regime}} = 1 - \frac{1}{D}\sum_{\ell=1}^{D}\frac{M_\ell K_\ell}{N^2}. \tag{20}$$

The geometry-salient top-α protection set is input dependent, so its overlap with the one-per-2×2 merge-destination grid can change M_ℓ and K_ℓ slightly across scenes. Dataset results use the exact counts accumulated during inference. A configuration-only calculation uses the expectation under uniformly located protected patches while retaining the reference frame, ragged grid boundaries, special tokens, and all four released layer groups.

Example. For 300-frame VGGT at 518×378 resolution, each frame has 999 patch and 5 special tokens, so $N = 301{,}200$. For ZEBRA3D 0.5L/0.8S (20% retention), a sparse layer attends 942 selected 64-token tile units, 3,741 landmarks, all 1,500 special tokens, and the self key. Including dense special-query rows gives $C_\ell/N^2 = 0.2215$. Since 0.5L uses 12 sparse and 12 linear layers,

$$\bar{s}_{\mathrm{Zebra}} = 1 - \frac{12(0.2215) + 12(0)}{24} = 0.8893. \tag{21}$$

For A3D-f4, each non-reference frame retains $\lceil 37/2\rceil\lceil 27/2\rceil = 266$ patches. Including the intact reference frame, special tokens, and one mean-fill token gives $V_4 = 82{,}034$. With $E = 9$ G2F stages among $D = 24$ stages,

$$\bar{s}_{\mathrm{A3D}} = 1 - \frac{9/300 + 15(82{,}034/301{,}200)}{24} = 0.8285. \tag{22}$$

For the canonical RegimeVGGT AB2 setting at $F = 10$, $N = 10{,}040$. The expected (M_ℓ, K_ℓ) counts are $(4046, 2345.3)$ for the ten shallow layers, $(5228.7, 5228.7)$ for the four merge-only rank-peak layers, $(5228.7, 3139.5)$ for the remaining four middle layers, and $(4046, 2038.7)$ for the six deep layers. Therefore

$$\begin{aligned}\bar{s}_{\mathrm{Regime}} = 1 - \frac{1}{24(10{,}040)^2}\big[&10(4046)(2345.3) + 4(5228.7)^2 \\ &+ 4(5228.7)(3139.5) + 6(4046)(2038.7)\big] = 0.8680.\end{aligned} \tag{23}$$

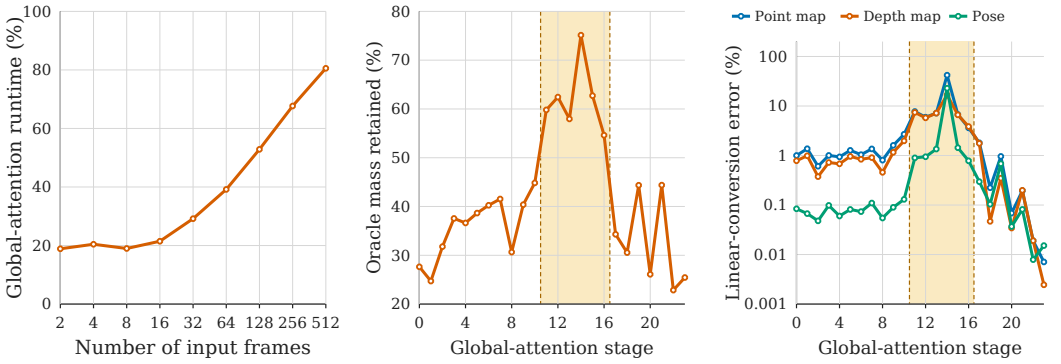
C Layer Structure and Allocation Analysis

C.1 Depth-Wise Structure Across Backbones

Figure 8 extends the VGGT measurements in Figure 2 to HYWM and Pi3 using the same measurement protocol. Across all three backbones, global attention dominates long-sequence inference, oracle block selection recovers only part of the teacher attention mass at a 20% block budget, and linear-centered conversion is most disruptive in a narrow middle band of global-attention stages. Figure 9 shows the corresponding tile-level teacher attention for eight VGGT stages across both linear bands and the tile-sparse band.

C.2 Layer-Role Placement Ablation

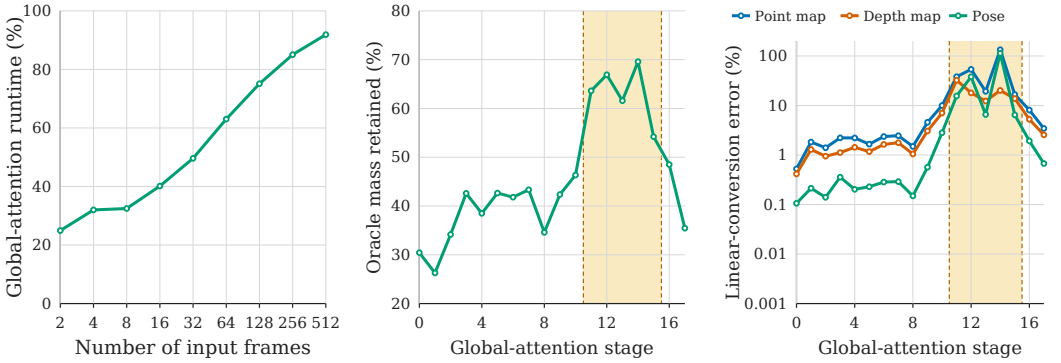
Table 7 reports every camera-pose metric for the placement study summarized in Table 3(d). All rows use the 0.5L/0.8S budget, so each backbone keeps the same number of linear and tile-sparse stages and the same closed-form initialization; only the set of stages given the linear role changes. No row receives gradient updates, so the sensitivity-guided rows coincide with the training-free 0.5L/0.8S entries of Table 1.



(a) HYWM: Global-attention runtime share.

(b) HYWM: Oracle top-20% block mass.

(c) HYWM: Per-stage teacher deviation.



(d) Pi3: Global-attention runtime share.

(e) Pi3: Oracle top-20% block mass.

(f) Pi3: Per-stage teacher deviation.

Figure 8: Additional background measurements for HYWM (top) and Pi3 (bottom). Global-attention runtime reaches 80.6% for HYWM and 91.9% for Pi3 at 512 frames. The middle column reports attention mass retained by oracle selection of the highest-mass 20% of 64×64 token blocks. The right column reports relative final-output deviation after replacing one global-attention stage with fitted linear-centered attention, averaged over 11 held-out validation scenes. Shading marks the sensitive middle bands: stages 11–16 for HYWM and 11–15 for Pi3.

C.3 Calibration-Set Stability of Layer Rankings

We test whether the stage assignment depends on the data used to measure conversion sensitivity. For each backbone, the single-stage profiling of Equation 1 is repeated with each of five calibration datasets, and the resulting stage ranking and operator assignment are compared with the reference Zebra3D ones (Tables 8 and 9). Rankings remain strongly correlated with the reference, with Spearman's ρ ranging from 0.7765 to 0.9365 and dataset means from 0.8425 to 0.8865. Mean assignment agreement ranges from 81.5% to 88.9% at 0.5L and from 93.5% to 97.2% at 0.7L, so the more linear configuration is especially stable. The division into linear-tolerant and sensitive stages is therefore not specific to one calibration collection.

D Operator and Refinement Ablations

D.1 Zero-Shot Linear-Attention Ablation

Tables 10–12 report the zero-shot linear-attention ablations across all three backbones and the 0.5L and 0.7L assignments. Lavender marks linear-centered attention, and the best result within each backbone–assignment group is bold. Linear-centered attention achieves the best RealEstate10K AUC-30 in every backbone–assignment group, with especially large gains over frame attention at

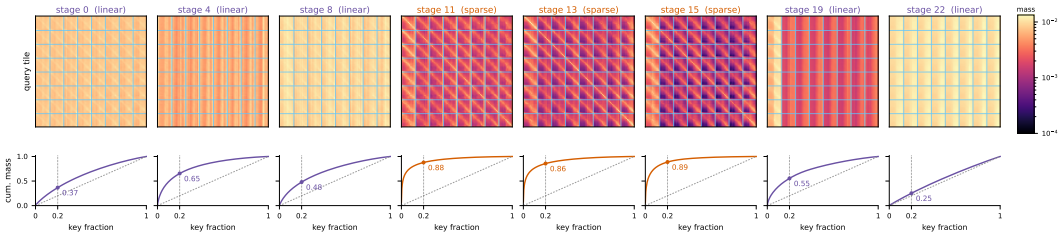


Figure 9: Tile-level teacher attention for eight VGGT global stages under the protocol of Figure 3, spanning both linear bands and the tile-sparse band. The transition into concentrated, stripe-structured attention at stage 11 and out of it after stage 15 matches the 0.7L assignment in Table 6.

Table 7: Layer-role placement ablation on RealEstate10K and CO3Dv2 camera pose at 0.5L/0.8S. Best per backbone in **bold**.

		RealEstate10K Camera Pose							CO3Dv2 Camera Pose			
Backbone	**Placement**	**RRA-5 ↑**	**RRA-15 ↑**	**RRA-30 ↑**	**RTA-5 ↑**	**RTA-15 ↑**	**RTA-30 ↑**	**AUC-30 ↑**	**AUC-3 ↑**	**AUC-5 ↑**	**AUC-15 ↑**	**AUC-30 ↑**
VGGT	**Sensitivity-guided (ours)**	**0.935**	**0.995**	**0.999**	**0.508**	**0.832**	**0.916**	**0.730**	**0.535**	**0.650**	**0.827**	**0.892**
	Back-sparse	0.901	0.984	0.993	0.313	0.676	0.831	0.590	0.052	0.111	0.355	0.553
	Front-sparse	0.541	0.878	0.961	0.041	0.218	0.453	0.208	0.000	0.000	0.004	0.023
	Uniform	0.485	0.842	0.948	0.017	0.116	0.294	0.115	0.000	0.000	0.006	0.030
	Reversed	0.383	0.755	0.910	0.016	0.119	0.312	0.114	0.000	0.000	0.002	0.014
	Random (seed 101)	0.783	0.947	0.983	0.080	0.330	0.585	0.311	0.002	0.009	0.089	0.220
	Random (seed 202)	0.884	0.977	0.990	0.242	0.613	0.791	0.534	0.017	0.047	0.252	0.471
	Random (seed 303)	0.377	0.752	0.907	0.016	0.114	0.303	0.108	0.000	0.000	0.003	0.017
HYWM	**Sensitivity-guided (ours)**	**0.984**	**0.997**	**0.999**	**0.758**	**0.907**	**0.951**	**0.850**	**0.376**	**0.521**	**0.763**	**0.854**
	Back-sparse	0.924	0.984	0.994	0.357	0.684	0.835	0.609	0.031	0.076	0.308	0.511
	Front-sparse	0.574	0.857	0.951	0.041	0.212	0.442	0.200	0.000	0.000	0.006	0.029
	Uniform	0.490	0.813	0.932	0.027	0.150	0.340	0.142	0.000	0.000	0.010	0.043
	Reversed	0.506	0.815	0.933	0.023	0.143	0.340	0.137	0.000	0.000	0.003	0.020
	Random (seed 101)	0.506	0.810	0.931	0.030	0.167	0.384	0.159	0.000	0.000	0.005	0.023
	Random (seed 202)	0.825	0.969	0.992	0.109	0.386	0.621	0.356	0.005	0.021	0.202	0.415
	Random (seed 303)	0.367	0.705	0.869	0.028	0.148	0.337	0.132	0.000	0.000	0.010	0.043
π^3	**Sensitivity-guided (ours)**	**0.988**	**0.998**	**1.000**	**0.792**	**0.917**	**0.953**	**0.866**	**0.542**	**0.657**	**0.831**	**0.895**
	Back-sparse	0.987	**0.998**	**1.000**	0.786	**0.917**	**0.953**	0.864	0.541	0.656	0.830	**0.895**
	Front-sparse	0.111	0.417	0.659	0.005	0.043	0.158	0.033	0.000	0.000	0.001	0.012
	Uniform	0.518	0.827	0.926	0.023	0.146	0.357	0.148	0.001	0.004	0.052	0.158
	Reversed	0.107	0.404	0.649	0.005	0.044	0.160	0.033	0.000	0.000	0.002	0.012
	Random (seed 101)	0.225	0.640	0.842	0.004	0.038	0.140	0.040	0.000	0.000	0.001	0.012
	Random (seed 202)	0.189	0.528	0.748	0.011	0.081	0.240	0.067	0.000	0.000	0.005	0.028
	Random (seed 303)	0.401	0.831	0.951	0.007	0.067	0.226	0.081	0.000	0.000	0.003	0.023

Table 8: Spearman correlation between layer-sensitivity rankings obtained from each calibration dataset and the reference Zebra3D rankings.

Calibration dataset	**VGGT**	**HYWM**	π^3	**Mean**
7Scenes	0.9322	0.8965	0.8308	0.8865
DTU	0.8852	0.9278	0.8369	0.8833
Re10K	0.9183	0.7765	0.8328	0.8425
NRGBD	0.8878	0.8635	0.8225	0.8579
CO3Dv2	0.9365	0.9235	0.7874	0.8825

0.7L. Point-map results are more metric-dependent, but fitted linear attention generally avoids the severe 0.7L degradation of frame attention; centering provides the strongest overall robustness across reconstruction and pose.

D.2 Training-Based Linear-Attention Ablation

Tables 13 and 14 compare HYWM mixers trained for 3,000 steps with the same data, optimization, and distillation recipe. The comparison covers Mamba-2 (Dao & Gu, 2024), linear-centered attention, Gated Delta Net (Yang et al., 2025b), and Sparse Linear Attn (Zhang et al., 2026b). Lavender marks the linear-centered operator used by Zebra3D, and bold marks the best result in each column within each dataset. No mixer dominates every metric: Sparse Linear Attn performs best on NRGBD, while Gated Delta Net leads several 7Scenes metrics. Linear-centered attention achieves

Table 9: Agreement with the reference 0.5L and 0.7L operator assignments, measured as the fraction of stages receiving the same linear or tile-sparse label.

	0.5L				0.7L			
Calibration dataset	**VGGT**	**HYWM**	π^3	**Mean**	**VGGT**	**HYWM**	π^3	**Mean**
7Scenes	91.7%	83.3%	88.9%	88.0%	91.7%	100.0%	100.0%	97.2%
DTU	83.3%	91.7%	77.8%	84.3%	91.7%	91.7%	100.0%	94.4%
Re10K	91.7%	75.0%	88.9%	85.2%	91.7%	100.0%	100.0%	97.2%
NRGBD	83.3%	83.3%	100.0%	88.9%	91.7%	100.0%	100.0%	97.2%
CO3Dv2	83.3%	83.3%	77.8%	81.5%	91.7%	100.0%	88.9%	93.5%

Table 10: Linear-attention ablation on 7Scenes.

Configuration	**Attention choice**	**Acc. mean ↓**	**Acc. med. ↓**	**Comp. mean ↓**	**Comp. med. ↓**	**NC1 mean ↑**	**NC1 med. ↑**	**NC2 mean ↑**	**NC2 med. ↑**
VGGT (0.5L/0.8S)	Frame attention	0.019	0.007	**0.025**	**0.009**	0.605	0.664	0.600	0.654
	Linear, no fit	0.018	0.007	0.027	0.012	**0.606**	**0.665**	0.602	0.658
	Linear, uncentered fit	0.018	0.006	0.028	0.012	0.605	0.665	0.604	0.661
	Linear-centered + fit	**0.018**	**0.006**	0.027	0.011	0.605	0.664	**0.604**	**0.661**
VGGT (0.7L/0.8S)	Frame attention	0.099	0.062	0.033	0.010	**0.609**	**0.674**	0.591	0.641
	Linear, no fit	0.018	0.007	0.025	0.010	0.606	0.666	0.601	0.656
	Linear, uncentered fit	**0.016**	**0.006**	0.026	0.011	0.607	0.667	**0.606**	**0.664**
	Linear-centered + fit	0.017	0.006	**0.025**	**0.008**	0.607	0.669	0.603	0.660
HYWM (0.5L/0.8S)	Frame attention	0.014	0.006	**0.022**	**0.007**	0.607	0.667	0.602	0.659
	Linear, no fit	0.013	0.005	0.025	0.010	0.608	0.669	0.607	0.666
	Linear, uncentered fit	0.013	0.005	0.025	0.010	**0.608**	**0.669**	0.608	0.668
	Linear-centered + fit	**0.013**	**0.005**	0.024	0.010	0.607	0.668	**0.608**	**0.668**
HYWM (0.7L/0.8S)	Frame attention	0.098	0.059	0.031	0.009	**0.614**	**0.683**	0.598	0.652
	Linear, no fit	**0.012**	0.005	0.023	0.008	0.610	0.672	0.609	0.669
	Linear, uncentered fit	0.013	0.005	**0.023**	**0.008**	0.610	0.672	**0.609**	**0.669**
	Linear-centered + fit	0.013	**0.005**	0.023	0.009	0.609	0.671	0.609	0.669
π^3 (0.5L/0.8S)	Frame attention	0.013	0.005	**0.017**	**0.006**	0.606	0.666	0.598	0.652
	Linear, no fit	0.012	0.004	0.019	0.008	0.606	0.666	0.601	0.657
	Linear, uncentered fit	0.012	**0.004**	0.019	0.007	**0.606**	**0.666**	**0.602**	**0.658**
	Linear-centered + fit	**0.012**	0.004	0.018	0.007	0.606	0.665	0.601	0.657
π^3 (0.7L/0.8S)	Frame attention	0.031	0.014	**0.018**	0.006	**0.625**	**0.698**	**0.585**	**0.632**
	Linear, no fit	0.028	0.013	0.021	0.006	0.620	0.691	0.579	0.622
	Linear, uncentered fit	0.024	0.010	0.019	**0.006**	0.619	0.689	0.582	0.626
	Linear-centered + fit	**0.021**	**0.008**	0.020	0.006	0.613	0.679	0.585	0.630

the best AUC-30 on both pose benchmarks (0.852 on RealEstate10K and 0.849 on CO3Dv2), making it the strongest balanced choice across reconstruction and pose.

D.3 Sparse-Attention Ablation

Tables 15–17 report the sparse-attention ablations across all three backbones at 0.5L and 0.7L, both with 0.75S. Lavender marks the operator selected for Zebra3D, and the best result within each backbone–assignment group is bold. Overall, the landmark tail improves sparse-attention quality, and tile sparse with landmarks provides the strongest aggregate trade-off. It generally outperforms block sparse and surpasses the SGA operator used by A3D (Sun et al., 2026) on NRGBD and RealEstate10K, although SGA remains competitive on some 7Scenes metrics.

D.4 Knowledge-Distillation Ablation

Tables 18 and 19 ablate the 3,000-step training objective for VGGT with the 0.5L/0.6S configuration. "GT only" uses task supervision alone, and "KD only" uses teacher distillation without ground-truth supervision. "GT + output KD" adds final-prediction matching to the task loss, while "GT + feature KD" adds intermediate-feature matching. RelKD combines task supervision with output, feature-L1, token-cosine, and cross-view relational losses and is the complete Zebra3D-KD objective. Lavender marks the reported Zebra3D-KD objective, and bold marks the best result in each column within each dataset.

Table 11: Linear-attention ablation on Neural RGB-D.

Configuration	Attention choice	Acc. mean ↓	Acc. med. ↓	Comp. mean ↓	Comp. med. ↓	NC1 mean ↑	NC1 med. ↑	NC2 mean ↑	NC2 med. ↑
VGGT (0.5L/0.8S)	Frame attention	0.020	0.013	**0.010**	**0.003**	0.730	0.881	0.713	0.866
	Linear, no fit	0.019	0.011	0.012	0.003	0.737	0.886	0.745	0.899
	Linear, uncentered fit	**0.017**	**0.010**	0.011	0.003	**0.743**	**0.891**	**0.757**	**0.906**
	Linear-centered + fit	0.021	0.013	0.014	0.004	0.739	0.883	0.744	0.881
VGGT (0.7L/0.8S)	Frame attention	0.201	0.143	0.041	0.011	0.672	0.829	0.695	0.857
	Linear, no fit	0.036	0.020	0.019	0.007	0.719	0.875	0.722	0.878
	Linear, uncentered fit	**0.024**	**0.013**	**0.015**	**0.004**	**0.733**	**0.883**	**0.741**	**0.894**
	Linear-centered + fit	0.029	0.017	0.018	0.005	0.721	0.872	0.720	0.867
HYWM (0.5L/0.8S)	Frame attention	0.021	0.014	**0.012**	**0.004**	0.720	0.868	0.708	0.858
	Linear, no fit	0.020	0.013	0.014	0.005	0.732	0.874	0.744	0.890
	Linear, uncentered fit	0.020	0.013	0.014	0.005	0.734	0.876	0.751	0.898
	Linear-centered + fit	**0.019**	**0.012**	0.013	0.005	**0.740**	**0.880**	**0.762**	**0.906**
HYWM (0.7L/0.8S)	Frame attention	0.201	0.150	0.043	0.012	0.681	0.848	0.695	0.859
	Linear, no fit	**0.022**	**0.013**	**0.012**	**0.003**	0.721	0.866	0.728	0.876
	Linear, uncentered fit	0.029	0.019	0.014	0.005	0.720	0.865	0.728	0.877
	Linear-centered + fit	0.022	0.015	0.014	0.005	**0.729**	**0.871**	**0.744**	**0.895**
π^3 (0.5L/0.8S)	Frame attention	0.016	0.009	0.010	0.003	0.723	0.860	0.713	0.854
	Linear, no fit	0.015	0.008	0.010	0.002	0.725	0.861	0.718	0.857
	Linear, uncentered fit	0.015	0.008	0.010	0.002	0.725	0.861	**0.720**	**0.858**
	Linear-centered + fit	**0.015**	**0.008**	**0.010**	**0.002**	**0.726**	**0.862**	0.718	0.858
π^3 (0.7L/0.8S)	Frame attention	0.051	0.032	**0.012**	0.004	**0.710**	**0.871**	0.677	0.824
	Linear, no fit	0.071	0.044	0.023	0.005	0.700	0.857	**0.680**	**0.828**
	Linear, uncentered fit	0.065	0.040	0.021	0.005	0.702	0.857	0.675	0.820
	Linear-centered + fit	**0.043**	**0.027**	0.015	**0.004**	0.708	0.862	0.678	0.822

Table 12: Linear-attention ablation on RealEstate10K camera pose.

Configuration	Attention choice	RRA-5 ↑	RRA-15 ↑	RRA-30 ↑	RTA-5 ↑	RTA-15 ↑	RTA-30 ↑	AUC-30 ↑
VGGT (0.5L/0.8S)	Frame attention	0.932	0.994	0.999	0.481	0.818	0.907	0.714
	Linear, no fit	0.928	0.994	0.999	0.471	0.822	0.912	0.714
	Linear, uncentered fit	0.928	0.995	**0.999**	0.475	0.820	0.910	0.713
	Linear-centered + fit	**0.935**	**0.995**	**0.999**	**0.508**	**0.832**	**0.916**	**0.730**
VGGT (0.7L/0.8S)	Frame attention	0.909	0.994	0.999	0.309	0.682	0.846	0.596
	Linear, no fit	0.921	0.994	0.999	0.440	0.819	0.915	0.707
	Linear, uncentered fit	0.920	0.993	0.999	0.457	0.821	0.914	0.710
	Linear-centered + fit	**0.932**	**0.995**	**1.000**	**0.459**	**0.828**	**0.918**	**0.718**
HYWM (0.5L/0.8S)	Frame attention	0.979	0.996	0.998	0.708	0.890	0.942	0.823
	Linear, no fit	0.984	**0.997**	0.999	0.741	0.903	0.948	0.840
	Linear, uncentered fit	0.984	**0.997**	0.999	0.745	0.903	0.948	0.842
	Linear-centered + fit	**0.984**	0.997	**0.999**	**0.758**	**0.907**	**0.951**	**0.850**
HYWM (0.7L/0.8S)	Frame attention	0.959	0.986	0.993	0.486	0.766	0.877	0.690
	Linear, no fit	0.977	0.997	0.999	0.685	0.885	0.939	0.811
	Linear, uncentered fit	0.977	0.997	0.999	0.692	0.887	0.939	0.814
	Linear-centered + fit	**0.983**	**0.998**	**1.000**	**0.739**	**0.903**	**0.948**	**0.839**
π^3 (0.5L/0.8S)	Frame attention	0.986	0.997	0.999	0.783	0.916	0.952	0.862
	Linear, no fit	**0.988**	0.998	1.000	0.788	0.917	0.952	0.864
	Linear, uncentered fit	0.988	0.998	**1.000**	0.791	0.917	0.952	0.865
	Linear-centered + fit	0.988	**0.998**	1.000	**0.792**	**0.917**	**0.953**	**0.867**
π^3 (0.7L/0.8S)	Frame attention	0.977	0.994	0.997	0.443	0.732	0.841	0.655
	Linear, no fit	0.980	0.997	1.000	0.641	0.866	0.931	0.789
	Linear, uncentered fit	0.983	0.997	0.999	0.682	0.883	0.936	0.811
	Linear-centered + fit	**0.985**	**0.998**	**1.000**	**0.725**	**0.901**	**0.948**	**0.836**

E Extended Results and Wide-Baseline Evaluation

E.1 Full Main-Benchmark Results

The following tables report the complete reconstruction and camera-pose metric breakdowns summarized by Table 1: Table 20 reports reconstruction results, and Table 21 reports camera-pose results.

E.2 Additional Performance–Sparsity Results

Figure 10 reports the corresponding 300-frame trade-offs for VGGT and π^3. The sparsity calculation and plotting protocol are identical across all three backbones.

Table 13: Training-based linear-attention ablation on 7Scenes (Shotton et al., 2013) and Neural RGB-D (Azinović et al., 2022).

Mixer	Acc. mean ↓	Acc. med. ↓	Comp. mean ↓	Comp. med. ↓	NC1 mean ↑	NC1 med. ↑	NC2 mean ↑	NC2 med. ↑
7Scenes								
Mamba-2	0.014	0.006	0.023	0.009	0.609	0.670	0.604	0.662
Linear-centered	**0.013**	0.005	0.023	0.009	0.607	0.667	0.606	0.664
Gated Delta Net	0.014	0.005	**0.022**	**0.008**	**0.609**	**0.670**	**0.607**	**0.666**
Sparse Linear Attn	0.013	**0.005**	0.024	0.009	0.608	0.668	0.606	0.665
Neural RGB-D								
Mamba-2	0.019	0.012	0.013	0.005	0.741	0.881	0.749	0.895
Linear-centered	0.019	0.012	0.014	0.005	0.746	0.885	0.769	0.910
Gated Delta Net	0.020	0.012	0.014	0.005	0.742	0.881	0.751	0.896
Sparse Linear Attn	**0.016**	**0.010**	**0.013**	**0.004**	**0.749**	**0.888**	**0.775**	**0.916**

Table 14: Training-based linear-attention ablation on RealEstate10K (Zhou et al., 2018) and CO3Dv2 (Reizenstein et al., 2021) camera pose.

	RealEstate10K							CO3Dv2			
Mixer	RRA-5 ↑	RRA-15 ↑	RRA-30 ↑	RTA-5 ↑	RTA-15 ↑	RTA-30 ↑	AUC-30 ↑	AUC-3 ↑	AUC-5 ↑	AUC-15 ↑	AUC-30 ↑
Mamba-2	0.985	0.998	**1.000**	0.746	0.907	0.952	0.845	**0.345**	**0.496**	**0.752**	0.848
Linear-centered	0.985	0.998	1.000	**0.762**	**0.912**	**0.953**	**0.852**	0.330	0.482	0.748	**0.849**
Gated Delta Net	0.984	**0.998**	**1.000**	0.711	0.900	0.948	0.828	0.305	0.461	0.735	0.839
Sparse Linear Attn	**0.987**	0.998	1.000	0.741	0.898	0.944	0.838	0.325	0.478	0.745	0.846

Table 15: Sparse-attention ablation on 7Scenes.

Configuration	Attention choice	Acc. mean ↓	Acc. med. ↓	Comp. mean ↓	Comp. med. ↓	NC1 mean ↑	NC1 med. ↑	NC2 mean ↑	NC2 med. ↑
VGGT (0.5L/0.75S)	SGA	**0.016**	**0.005**	**0.026**	0.010	0.605	0.664	**0.605**	**0.664**
	Block sparse	0.018	0.006	0.026	**0.009**	0.606	**0.666**	0.603	0.660
	Tile sparse	0.018	0.006	0.027	0.010	0.604	0.664	0.604	0.661
	Block sparse + landmark	0.018	0.006	0.027	0.010	**0.606**	0.665	0.603	0.660
	Tile sparse + landmark	0.017	0.006	0.027	0.011	0.605	0.664	0.605	0.662
VGGT (0.7L/0.75S)	SGA	**0.016**	**0.006**	**0.023**	**0.007**	0.607	0.668	**0.604**	**0.661**
	Block sparse	0.018	0.006	0.025	0.008	0.607	0.668	0.602	0.658
	Tile sparse	0.017	0.006	0.025	0.008	0.606	0.667	0.602	0.659
	Block sparse + landmark	0.017	0.006	0.025	0.008	**0.608**	**0.670**	0.603	0.660
	Tile sparse + landmark	0.017	0.006	0.025	0.008	0.607	0.669	0.604	0.661
HYWM (0.5L/0.75S)	SGA	0.013	0.005	0.024	0.009	0.608	0.669	0.609	0.669
	Block sparse	0.014	0.005	0.024	**0.008**	0.606	0.667	0.606	0.665
	Tile sparse	0.013	0.005	0.025	0.009	0.605	0.664	0.606	0.665
	Block sparse + landmark	0.013	0.005	**0.024**	0.009	**0.609**	**0.670**	**0.609**	**0.669**
	Tile sparse + landmark	**0.013**	**0.005**	0.025	0.010	0.608	0.668	0.608	0.668
HYWM (0.7L/0.75S)	SGA	0.013	0.005	**0.022**	0.008	0.609	0.671	0.609	0.669
	Block sparse	0.013	0.005	0.023	**0.008**	0.609	0.671	0.608	0.669
	Tile sparse	0.013	0.005	0.023	0.008	0.609	0.670	0.609	0.669
	Block sparse + landmark	**0.012**	**0.005**	0.023	0.008	**0.610**	**0.673**	0.609	0.669
	Tile sparse + landmark	0.013	0.005	0.023	0.009	0.609	0.671	**0.609**	**0.670**
π^3 (0.5L/0.75S)	SGA	**0.011**	**0.004**	0.016	0.005	0.605	0.665	0.600	0.655
	Block sparse	0.024	0.010	0.017	0.004	0.625	0.700	0.586	0.632
	Tile sparse	0.027	0.013	**0.014**	**0.004**	**0.628**	**0.705**	0.590	0.639
	Block sparse + landmark	0.013	0.005	0.017	0.005	0.610	0.673	0.595	0.647
	Tile sparse + landmark	0.012	0.004	0.018	0.007	0.606	0.665	**0.602**	**0.658**
π^3 (0.7L/0.75S)	SGA	0.022	0.009	**0.018**	**0.005**	**0.617**	**0.686**	**0.585**	**0.631**
	Block sparse	0.024	0.010	0.021	0.006	0.613	0.678	0.583	0.628
	Tile sparse	**0.020**	0.008	0.019	0.006	0.615	0.683	0.585	0.631
	Block sparse + landmark	0.025	0.010	0.022	0.006	0.610	0.674	0.582	0.627
	Tile sparse + landmark	0.020	**0.008**	0.020	0.006	0.614	0.680	0.585	0.631

E.3 Wide-Baseline Point-Map Reconstruction

The benchmarks of Section 4 are dominated by video-like capture: 7Scenes, NRGBD, and ScanNet-50 are handheld sequences, and RealEstate10K and CO3Dv2 supply densely sampled trajectories. Consecutive views therefore share a narrow baseline, the setting in which attention is most likely to concentrate on a few neighboring frames. This appendix evaluates the converted models outside that setting, on Tanks and Temples (Knapitsch et al., 2017) (large-scale outdoor and indoor scenes) and ETH3D (Schops et al., 2017) (indoor and outdoor DSLR captures), with DTU (Aanæs et al., 2016) retained as an object-centric turntable control. Table 22 averages over scenes and over view counts $N \in \{8, 16, 32, 64\}$, so each entry spans both sparse and moderately dense coverage of the same scene. All rows share the inputs, preprocessing, and metric definitions of Appendix B, and

Table 16: Sparse-attention ablation on Neural RGB-D.

Configuration	Attention choice	Acc. mean ↓	Acc. med. ↓	Comp. mean ↓	Comp. med. ↓	NC1 mean ↑	NC1 med. ↑	NC2 mean ↑	NC2 med. ↑
VGGT (0.5L/0.75S)	SGA	0.029	0.022	0.030	0.009	0.713	0.845	0.720	0.859
	Block sparse	0.038	0.023	0.016	0.004	0.708	0.853	0.714	0.864
	Tile sparse	0.039	0.034	0.020	0.006	0.708	0.847	0.707	0.854
	Block sparse + landmark	0.034	0.020	0.015	0.004	0.734	0.884	0.747	**0.898**
	Tile sparse + landmark	**0.018**	**0.011**	**0.013**	**0.003**	**0.743**	**0.888**	**0.750**	0.892
VGGT (0.7L/0.75S)	SGA	0.040	0.030	0.028	0.009	0.716	0.864	0.704	0.843
	Block sparse	0.029	0.018	**0.016**	0.004	0.714	0.867	0.729	0.882
	Tile sparse	0.030	0.017	0.018	0.005	0.718	0.865	0.719	0.863
	Block sparse + landmark	0.031	0.018	0.017	**0.004**	0.717	**0.872**	**0.731**	**0.889**
	Tile sparse + landmark	**0.029**	**0.017**	0.018	0.005	**0.722**	0.871	0.719	0.866
HYWM (0.5L/0.75S)	SGA	0.020	0.012	0.014	**0.004**	0.738	0.879	0.758	0.906
	Block sparse	0.036	0.023	0.022	0.006	0.700	0.839	0.719	0.861
	Tile sparse	0.032	0.021	0.018	0.005	0.701	0.846	0.701	0.849
	Block sparse + landmark	0.022	0.014	0.014	0.005	0.736	0.879	0.753	0.904
	Tile sparse + landmark	**0.019**	**0.012**	**0.013**	0.005	**0.741**	**0.881**	**0.764**	**0.907**
HYWM (0.7L/0.75S)	SGA	0.024	0.015	**0.013**	**0.003**	0.726	0.872	0.743	**0.895**
	Block sparse	0.031	0.020	0.016	0.005	0.719	0.863	0.728	0.879
	Tile sparse	**0.022**	**0.014**	0.013	0.004	0.719	0.861	0.729	0.879
	Block sparse + landmark	0.032	0.020	0.016	0.005	0.721	0.866	0.734	0.884
	Tile sparse + landmark	0.023	0.015	0.014	0.005	**0.730**	**0.872**	**0.745**	0.895
π^3 (0.5L/0.75S)	SGA	0.017	0.010	0.010	0.003	0.717	0.857	0.686	0.822
	Block sparse	0.058	0.040	0.019	0.005	0.689	0.840	0.677	0.818
	Tile sparse	0.066	0.047	0.016	0.004	0.695	0.854	0.677	0.825
	Block sparse + landmark	0.024	0.014	0.012	0.003	0.721	**0.866**	0.702	0.844
	Tile sparse + landmark	**0.014**	**0.008**	**0.010**	**0.002**	**0.728**	0.864	**0.723**	**0.862**
π^3 (0.7L/0.75S)	SGA	0.043	0.028	**0.014**	0.004	0.708	0.864	0.675	0.820
	Block sparse	0.053	0.033	0.023	0.004	0.701	0.853	0.675	0.821
	Tile sparse	**0.042**	0.027	0.016	0.004	**0.709**	**0.864**	**0.677**	**0.823**
	Block sparse + landmark	0.054	0.031	0.022	0.004	0.702	0.855	0.675	0.819
	Tile sparse + landmark	0.043	**0.027**	0.015	**0.004**	0.708	0.863	0.677	0.821

each converted model is compared against that backbone's own dense model on the same scenes, which isolates the cost of conversion from the backbone's absolute quality.

Conversion cost at the 0.5L operating points. Normal consistency is the metric most sensitive to fine-grained correspondence, and it is where a widened baseline should hurt first. Across all three benchmarks and all three backbones, every 0.5L configuration stays within 3% of its dense backbone in NC1; the largest gap is 2.9% (HYWM 0.5L/0.8S on Tanks and Temples, 0.6790 versus 0.6992), and most are inside 2%. Point-map errors follow the same pattern and occasionally improve on dense: on ETH3D, every VGGT 0.5L configuration beats dense in both accuracy and completeness, with Zebra3D-KD 0.5L/0.6S reaching 0.2438 m accuracy and 0.1816 m completeness against 0.2898 m and 0.3321 m for dense VGGT. Tanks and Temples is the hardest of the three, and the damage is concentrated in completeness on HYWM ($+0.53$ m at 0.5L/0.6S, reduced to $+0.04$ m after distillation), while the same operating point stays within $+0.06$ m on VGGT and $+0.03$ m on π^3.

Where the budget does break. The 0.7L configurations are less forgiving here than on the video benchmarks, and π^3 degrades most: 0.7L/0.8S loses 0.076 NC1 on Tanks and Temples (10.4% relative) and 0.054 on ETH3D (6.4%), and on DTU its accuracy grows from 1.0550 mm to 1.9583 mm at 0.7L/0.6S. This is consistent with the placement analysis of Section 4.3: π^3 has the latest sensitive band, so raising the linear fraction encroaches on it soonest. We therefore regard 0.5L as the operating point that transfers to wide baselines, and 0.7L as a speed-first setting validated mainly on the video-like benchmarks.

Table 17: Sparse-attention ablation on RealEstate10K camera pose.

Configuration	Attention choice	RRA-5 ↑	RRA-15 ↑	RRA-30 ↑	RTA-5 ↑	RTA-15 ↑	RTA-30 ↑	AUC-30 ↑
VGGT (0.5L/0.75S)	SGA	**0.949**	**0.996**	**0.999**	**0.519**	**0.841**	**0.924**	**0.739**
	Block sparse	0.945	0.993	0.998	0.466	0.811	0.906	0.709
	Tile sparse	0.943	0.995	0.999	0.492	0.824	0.910	0.722
	Block sparse + landmark	0.939	0.993	0.998	0.487	0.825	0.912	0.722
	Tile sparse + landmark	0.938	0.995	0.999	0.516	0.834	0.917	0.733
VGGT (0.7L/0.75S)	SGA	**0.943**	**0.996**	**1.000**	**0.471**	**0.836**	**0.925**	**0.727**
	Block sparse	0.939	0.995	0.999	0.442	0.819	0.914	0.710
	Tile sparse	0.939	0.996	0.999	0.465	0.830	0.918	0.720
	Block sparse + landmark	0.935	0.994	0.999	0.437	0.817	0.913	0.708
	Tile sparse + landmark	0.933	0.995	1.000	0.464	0.830	0.919	0.721
HYWM (0.5L/0.75S)	SGA	**0.985**	**0.998**	**1.000**	0.745	0.903	0.949	0.843
	Block sparse	0.984	0.997	0.998	0.727	0.897	0.944	0.833
	Tile sparse	0.985	0.998	1.000	0.721	0.892	0.942	0.828
	Block sparse + landmark	0.982	0.996	0.999	0.746	0.906	0.949	0.844
	Tile sparse + landmark	0.985	0.998	0.999	**0.763**	**0.908**	**0.951**	**0.852**
HYWM (0.7L/0.75S)	SGA	**0.985**	0.998	**1.000**	0.722	0.896	0.945	0.830
	Block sparse	0.982	0.996	0.999	0.713	0.895	0.942	0.826
	Tile sparse	0.984	**0.998**	1.000	0.716	0.893	0.943	0.826
	Block sparse + landmark	0.981	0.996	0.999	0.721	0.900	0.946	0.831
	Tile sparse + landmark	0.984	0.998	1.000	**0.744**	**0.904**	**0.949**	**0.841**
π^3 (0.5L/0.75S)	SGA	0.986	0.998	1.000	0.767	0.908	0.947	0.847
	Block sparse	0.987	**0.998**	**1.000**	0.689	0.880	0.933	0.808
	Tile sparse	0.988	0.998	1.000	0.693	0.878	0.935	0.811
	Block sparse + landmark	0.987	0.998	1.000	0.784	**0.921**	**0.954**	0.864
	Tile sparse + landmark	**0.988**	0.998	1.000	**0.795**	0.919	0.954	**0.868**
π^3 (0.7L/0.75S)	SGA	0.969	0.993	0.997	0.621	0.826	0.896	0.759
	Block sparse	0.978	0.998	1.000	0.684	0.888	0.944	0.815
	Tile sparse	0.986	**0.998**	**1.000**	0.677	0.878	0.934	0.807
	Block sparse + landmark	0.981	0.997	1.000	0.709	**0.902**	**0.952**	0.831
	Tile sparse + landmark	**0.986**	**0.998**	**1.000**	**0.727**	0.901	0.948	**0.837**

Table 18: Knowledge-distillation ablation on 7Scenes (Shotton et al., 2013) and Neural RGB-D (Azinović et al., 2022).

Training objective	Acc. mean ↓	Acc. med. ↓	Comp. mean ↓	Comp. med. ↓	NC1 mean ↑	NC1 med. ↑	NC2 mean ↑	NC2 med. ↑
7Scenes								
GT only	0.023	0.008	0.023	0.007	0.612	0.678	0.597	0.650
KD only	0.023	0.010	**0.020**	**0.004**	0.611	0.674	0.580	0.623
GT + output KD	0.023	0.010	0.021	0.005	**0.620**	**0.690**	0.596	0.648
GT + feature KD	0.029	0.012	0.023	0.007	0.616	0.683	0.592	0.641
RelKD	**0.015**	**0.005**	0.026	0.011	0.608	0.668	**0.606**	**0.665**
Neural RGB-D								
GT only	0.046	0.027	0.018	0.005	0.699	0.848	0.685	0.828
KD only	0.045	0.031	0.014	**0.004**	0.696	0.839	0.687	0.825
GT + output KD	0.041	0.029	0.015	**0.004**	0.701	0.858	0.695	0.847
GT + feature KD	0.047	0.031	**0.011**	**0.004**	0.705	0.866	0.693	0.848
RelKD	**0.024**	**0.013**	0.018	0.005	**0.751**	**0.888**	**0.762**	**0.896**

Table 19: Knowledge-distillation ablation on RealEstate10K camera pose (Zhou et al., 2018).

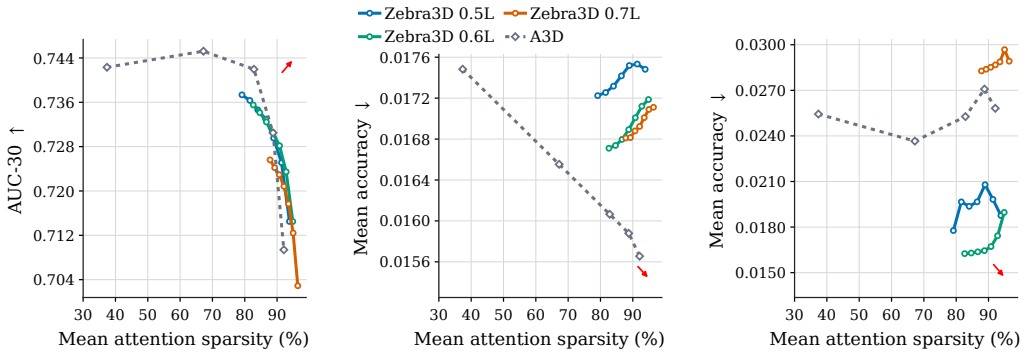
Training objective	RRA-5 ↑	RRA-15 ↑	RRA-30 ↑	RTA-5 ↑	RTA-15 ↑	RTA-30 ↑	AUC-30 ↑
GT only	**0.980**	0.997	1.000	0.586	0.870	0.939	0.777
KD only	0.961	0.998	1.000	0.449	0.823	0.916	0.716
GT + output KD	0.973	0.997	0.999	0.488	0.838	0.922	0.736
GT + feature KD	0.977	0.998	**1.000**	0.542	0.854	0.929	0.756
RelKD	0.975	**0.998**	1.000	**0.649**	**0.895**	**0.950**	**0.805**

Table 20: Full quantitative point-map reconstruction results on 7Scenes and NRGBD at stride 10. Dense backbones are excluded when marking the best (**bold**) and second-best (underlined) non-dense results. In the shorthand xL/yS, x is the fraction of model layers using linear attention and y is the sparsity of attention rows in the remaining layers. VGGT + GT-only FT is dense VGGT fine-tuned with the same data, schedule, and ground-truth supervision as Zebra3D-KD.

Method	Mean Attn. Sparsity ↑	7Scenes						NRGBD					
		Acc ↓		Comp ↓		NC1 ↑		Acc ↓		Comp ↓		NC1 ↑	
		Mean	*Med.*	*Mean*	*Med.*	*Mean*	*Med.*	*Mean*	*Med.*	*Mean*	*Med.*	*Mean*	*Med.*
VGGT	0%	0.017	0.005	0.027	0.011	0.607	0.667	0.016	0.009	0.011	0.003	0.761	0.901
VGGT + GT-only FT	0%	0.024	0.008	0.031	0.014	0.604	0.662	0.024	0.013	0.018	0.006	0.754	0.890
FastVGGT	88%	**0.015**	**0.005**	0.025	0.009	0.607	0.667	0.020	0.013	**0.013**	0.004	0.731	0.871
SAF3R	84%	0.018	0.008	**0.022**	**0.005**	0.611	0.675	0.054	0.034	0.022	0.005	0.694	0.847
RegimeVGGT	90%	0.016	**0.005**	0.026	0.010	0.606	0.667	0.024	0.014	0.016	0.005	0.736	0.874
Block Sparse 3D	70%	0.017	0.006	0.025	0.007	0.608	0.670	0.042	0.029	0.016	0.004	0.696	0.837
Lite3R	70%	0.039	0.020	0.023	**0.005**	**0.619**	**0.688**	0.068	0.041	0.018	0.005	0.689	0.843
A3D	83%	0.016	**0.005**	0.026	0.010	0.607	0.668	0.025	0.014	0.017	0.005	0.735	0.877
Zebra3D (0.5L/0.6S)	79%	0.017	0.006	0.027	0.012	0.606	0.665	**0.018**	**0.011**	**0.013**	**0.003**	0.746	**0.891**
Zebra3D-KD (0.5L/0.6S)	79%	**0.015**	**0.005**	0.026	0.011	0.608	0.668	0.024	0.013	0.018	0.005	**0.751**	0.888
Zebra3D-KD (0.7L/0.8S)	**94%**	0.016	0.006	0.025	0.011	0.608	0.669	0.025	0.015	0.019	0.006	0.739	0.879
HYWM	0%	0.013	0.005	0.024	0.010	0.608	0.668	0.018	0.011	0.014	0.004	0.750	0.890
Block Sparse 3D	70%	0.014	0.006	0.024	**0.007**	0.606	0.667	0.041	0.026	0.023	0.006	0.688	0.828
A3D	83%	**0.013**	**0.005**	0.024	0.009	**0.608**	**0.669**	0.019	**0.011**	**0.014**	**0.004**	0.740	0.882
RegimeVGGT	90%	**0.013**	**0.005**	0.024	0.009	0.607	0.667	0.019	0.011	**0.014**	0.004	0.734	0.873
Zebra3D (0.5L/0.6S)	79%	**0.013**	**0.005**	0.025	0.010	**0.608**	**0.669**	0.019	0.012	**0.014**	0.005	0.744	0.883
Zebra3D-KD (0.5L/0.6S)	79%	**0.013**	**0.005**	0.023	0.009	**0.608**	**0.669**	**0.018**	**0.011**	**0.014**	0.005	**0.749**	**0.887**
Zebra3D-KD (0.7L/0.8S)	**94%**	0.014	**0.005**	**0.022**	0.008	**0.608**	0.668	0.019	0.013	**0.014**	0.005	0.737	0.877
π^3	0%	0.012	0.004	0.017	0.007	0.607	0.668	0.014	0.007	0.009	0.003	0.737	0.872
Block Sparse 3D	70%	0.020	0.008	**0.016**	**0.004**	**0.622**	**0.695**	0.049	0.033	0.017	0.004	0.693	0.844
A3D	88%	**0.012**	**0.004**	**0.016**	0.005	0.607	0.667	0.018	0.010	0.010	0.003	0.716	0.857
Zebra3D (0.5L/0.6S)	79%	**0.012**	**0.004**	0.018	0.007	0.606	0.666	**0.014**	**0.007**	**0.009**	**0.002**	**0.731**	**0.866**
Zebra3D (0.7L/0.8S)	**94%**	0.021	0.008	0.020	0.006	0.613	0.679	0.043	0.027	0.015	0.004	0.708	0.863
Zebra3D-KD (0.5L/0.6S)	79%	**0.012**	**0.004**	0.018	0.007	0.604	0.663	0.015	0.008	**0.009**	**0.002**	0.728	0.861
Zebra3D-KD (0.7L/0.8S)	**94%**	0.013	0.005	0.018	0.006	0.605	0.665	0.017	0.009	0.010	0.003	0.718	0.854

Table 21: Full camera-pose estimation results on RealEstate10K and CO3Dv2 using 10 input frames. Higher is better. Dense backbones are excluded when marking the best (**bold**) and second-best (underlined) non-dense results; dashes denote unavailable measurements. In the shorthand xL/yS, x is the fraction of model layers using linear attention and y is the sparsity of attention rows in the remaining layers. VGGT + GT-only FT is dense VGGT fine-tuned with the same data, schedule, and ground-truth supervision as Zebra3D-KD.

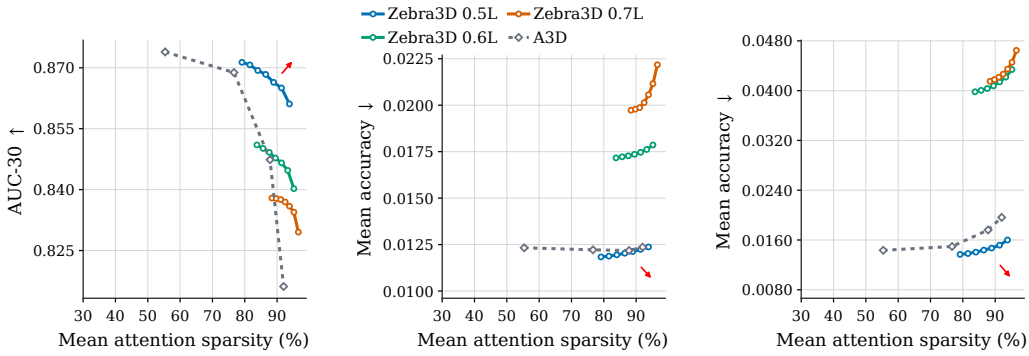
Method	Mean Attn. Sparsity ↑	RealEstate10K Camera Pose							CO3Dv2 Camera Pose			
		RRA-5 ↑	RRA-15 ↑	RRA-30 ↑	RTA-5 ↑	RTA-15 ↑	RTA-30 ↑	AUC-30 ↑	AUC-3 ↑	AUC-5 ↑	AUC-15 ↑	AUC-30 ↑
VGGT	0%	0.947	0.996	0.999	0.543	0.843	0.920	0.744	0.566	0.675	0.839	0.900
VGGT + GT-only FT	0%	0.980	0.998	1.000	0.674	0.901	0.951	0.816	0.474	0.609	0.812	0.885
FastVGGT	83%	0.936	0.994	0.999	0.476	0.808	0.904	0.708	0.412	0.555	0.784	0.867
SAF3R	78%	0.946	0.995	0.999	0.493	0.809	0.902	0.714	0.504	0.626	0.814	0.883
RegimeVGGT	87%	0.939	0.995	0.999	0.498	0.824	0.911	0.722	0.445	0.582	0.797	0.876
Block Sparse 3D	70%	0.937	0.994	0.998	0.379	0.745	0.869	0.649	0.207	0.390	0.721	0.835
Lite3R	70%	0.933	0.996	0.999	0.253	0.666	0.843	0.578	0.078	0.220	0.621	0.779
A3D	75%	0.952	0.996	0.999	0.536	0.839	0.919	0.742	0.468	0.602	0.806	0.881
Zebra3D (0.5L/0.6S)	79%	0.941	0.996	0.999	0.526	0.839	0.919	0.737	**0.554**	**0.665**	**0.834**	**0.897**
Zebra3D-KD (0.5L/0.6S)	79%	**0.975**	**0.998**	1.000	0.649	**0.895**	**0.950**	0.805	0.458	0.598	0.808	0.883
Zebra3D-KD (0.7L/0.8S)	**94%**	0.975	**0.998**	**1.000**	**0.654**	0.894	0.947	**0.805**	0.403	0.556	0.789	0.872
HYWM	0%	0.986	0.998	1.000	0.774	0.915	0.955	0.860	0.403	0.544	0.777	0.865
Block Sparse 3D	70%	0.981	0.996	0.998	0.573	0.810	0.890	0.735	0.085	0.199	0.590	0.761
A3D	75%	0.984	0.998	0.999	0.740	0.900	0.947	0.839	0.323	0.474	0.743	0.844
RegimeVGGT	87%	0.985	**0.998**	**1.000**	0.741	0.902	0.947	0.840	0.315	0.468	0.740	0.843
Zebra3D (0.5L/0.6S)	79%	**0.986**	0.997	0.999	**0.769**	0.911	0.953	**0.856**	**0.400**	**0.541**	**0.774**	**0.862**
Zebra3D-KD (0.5L/0.6S)	79%	0.986	0.998	1.000	0.766	**0.913**	**0.954**	0.854	0.332	0.481	0.748	0.849
Zebra3D-KD (0.7L/0.8S)	**94%**	0.984	0.998	0.999	0.752	0.907	0.949	0.847	0.295	0.447	0.731	0.838
π^3	0%	0.988	0.998	1.000	0.809	0.926	0.958	0.877	0.582	0.687	0.844	0.903
Block Sparse 3D	70%	0.987	**0.998**	**1.000**	0.655	0.855	0.918	0.786	0.213	0.387	0.720	0.835
A3D	82%	0.985	**0.998**	**1.000**	0.765	0.909	0.950	0.847	0.457	0.593	0.804	0.880
Zebra3D (0.5L/0.6S)	79%	**0.988**	**0.998**	**1.000**	**0.800**	**0.921**	**0.955**	**0.871**	**0.563**	**0.673**	**0.839**	**0.900**
Zebra3D (0.7L/0.8S)	**94%**	0.985	**0.998**	**1.000**	0.726	0.901	0.948	0.836	0.256	0.425	0.731	0.841
Zebra3D-KD (0.5L/0.6S)	79%	**0.988**	**0.998**	**1.000**	0.773	0.907	0.948	0.855	0.527	0.649	0.829	0.895
Zebra3D-KD (0.7L/0.8S)	**94%**	0.987	**0.998**	**1.000**	0.751	0.901	0.945	0.844	0.440	0.581	0.797	0.876



(a) VGGT: RealEstate10K camera pose.

(b) VGGT: 7Scenes point maps.

(c) VGGT: Neural RGB-D point maps.



(d) π^3: RealEstate10K camera pose.

(e) π^3: 7Scenes point maps.

(f) π^3: Neural RGB-D point maps.

Figure 10: Performance versus mean attention sparsity for VGGT and π^3 and mean attention sparsity is computed at 300 input frames. Zebra3D curves vary sparse retention within the 0.5L, 0.6L, and 0.7L placements; gray A3D curves vary the subsampling configuration. Higher is better for RealEstate10K AUC-30, while lower is better for 7Scenes and Neural RGB-D point-map accuracy error. The gold arrow points toward each panel's preferred corner.

Table 22: Wide-baseline point-map accuracy on Tanks and Temples (large-scale outdoor and indoor scenes) and ETH3D (indoor and outdoor DSLR captures), with DTU (object-centric turntable rig) retained as a narrow-baseline control. Each entry is averaged over scenes and view counts $N \in \{8, 16, 32, 64\}$. Accuracy and completeness are lower-is-better, in m for Tanks and Temples and ETH3D and in mm for DTU; NC1 is higher is better. Dense rows are excluded when marking the best (**bold**) result in each column within a backbone block.

	Tanks and Temples			ETH3D			DTU		
Method	**Acc ↓**	**Comp ↓**	**NC1 ↑**	**Acc ↓**	**Comp ↓**	**NC1 ↑**	**Acc ↓**	**Comp ↓**	**NC1 ↑**
HYWM									
Dense	0.9865	0.4437	0.6992	0.2591	0.1857	0.8198	0.9622	1.4920	0.6549
A3D	1.4735	1.7308	0.6681	0.4634	0.4119	0.7723	0.9990	**1.4659**	0.6512
Zebra3D (0.5L/0.6S)	1.2380	0.9749	0.6817	**0.2437**	**0.1681**	**0.8191**	**0.9563**	1.5110	**0.6565**
Zebra3D (0.5L/0.8S)	1.3271	0.9866	0.6790	0.2629	0.1830	0.8151	0.9766	1.5020	0.6559
Zebra3D (0.7L/0.6S)	**1.1272**	0.8316	0.6795	0.2829	0.1857	0.8059	1.0430	1.4766	0.6557
Zebra3D (0.7L/0.8S)	1.4817	1.6450	0.6678	0.3073	0.2013	0.8034	1.0691	1.5129	0.6551
Zebra3D-KD (0.5L/0.6S)	1.1364	**0.4831**	**0.6889**	0.2989	0.2242	0.8101	1.1988	1.8062	0.6549
Zebra3D-KD (0.7L/0.8S)	1.4398	0.7431	0.6779	0.3017	0.2159	0.8052	1.2152	1.7885	0.6543
VGGT									
Dense	0.7303	0.6909	0.7231	0.2898	0.3321	0.8056	1.1153	1.9545	0.6768
A3D	0.8312	0.8278	0.6965	0.4995	0.5108	0.7657	1.1285	1.9411	0.6675
Zebra3D (0.5L/0.6S)	**0.6977**	0.7481	**0.7184**	0.2605	0.2606	0.8097	1.1165	1.9084	**0.6739**
Zebra3D (0.5L/0.8S)	0.7732	0.8653	0.7069	0.2513	0.2307	0.8109	**1.1162**	1.8913	0.6720
Zebra3D (0.7L/0.6S)	0.7083	**0.7074**	0.7062	0.2540	0.2302	0.7994	1.2368	1.8959	0.6610
Zebra3D (0.7L/0.8S)	0.7441	0.7740	0.6993	0.2499	0.2235	0.7996	1.2465	1.8868	0.6600
Zebra3D-KD (0.5L/0.6S)	0.7878	0.8131	0.7060	**0.2438**	**0.1816**	**0.8222**	1.1562	**1.8787**	0.6703
Zebra3D-KD (0.7L/0.8S)	0.7954	0.7609	0.6934	0.2595	0.1942	0.8173	1.1552	1.8844	0.6606
π^3									
Dense	0.7478	0.4910	0.7311	0.1281	0.0811	0.8472	1.0550	1.5950	0.6463
A3D	1.1478	0.8858	0.6832	0.3332	0.2416	0.7807	1.3521	1.6462	0.6530
Zebra3D (0.5L/0.6S)	**0.7488**	0.5141	**0.7276**	**0.1322**	**0.0843**	**0.8424**	**1.1458**	1.6189	0.6502
Zebra3D (0.5L/0.8S)	0.7812	0.5679	0.7187	0.1458	0.0957	0.8356	1.1807	1.6009	0.6505
Zebra3D (0.7L/0.6S)	0.9643	0.6009	0.6627	0.2401	0.1484	0.7956	1.9583	1.7168	0.6595
Zebra3D (0.7L/0.8S)	1.1376	0.9781	0.6551	0.2448	0.1526	0.7935	2.0262	1.7319	**0.6607**
Zebra3D-KD (0.5L/0.6S)	0.9205	**0.4656**	0.7141	0.1463	0.0941	0.8335	1.1493	1.6234	0.6454
Zebra3D-KD (0.7L/0.8S)	1.1929	0.7778	0.6890	0.1966	0.1246	0.8196	1.2652	**1.5940**	0.6516

F Qualitative Results and Limitations

F.1 Additional Qualitative Reconstructions

Figure 11 extends Figure 6 to every scene used for qualitative inspection. Scenes were selected from the per-scene Chamfer distances already recorded for Table 2, before any render was produced, so the choice is made on the reported metric rather than on appearance.

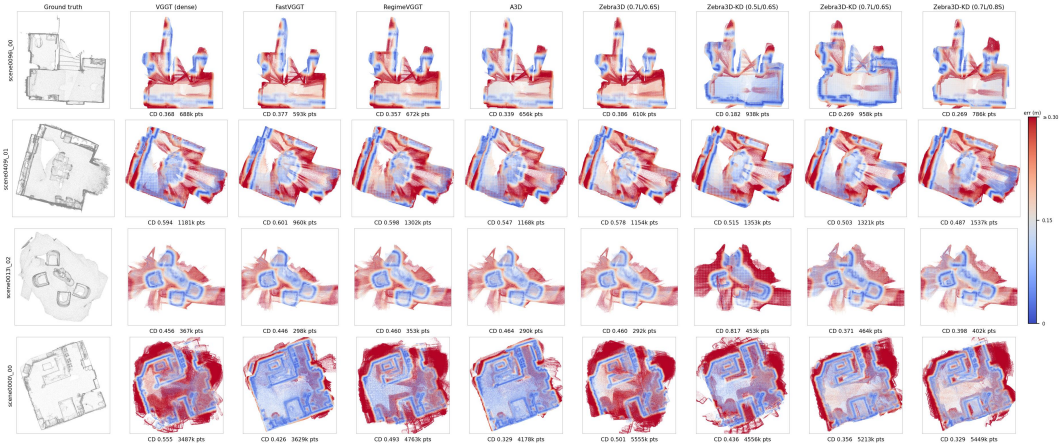


Figure 11: Long-sequence reconstruction at 1,000 views on four ScanNet-50 scenes, coloured by distance to the ground-truth mesh (blue 0 m, red ≥ 0.30 m). Protocol is identical to Figure 6.

The renders re-fit the evaluation's similarity transform on 1–99 percentile extents, identically for every method, so that a few unprojection outliers cannot shrink the room toward the centre of the frame; no points are removed, and every reported Chamfer distance uses the evaluation's own bounding-box alignment.

A lower Chamfer distance can in principle be bought by smoothing: points spread around the true surface score well under a two-sided mean distance, while the render looks softer. That is not what happens here. Averaged over the four scenes of Figure 11, Zebra3D -KD (0.7L/0.6S) improves the median per-point error from 0.217 to 0.174 m, the 90th percentile from 0.536 to 0.420 m, and the share of points beyond 0.30 m from 33.0% to 23.1%, so the gain is present at every quantile rather than only in the mean. Local surface thickness, estimated as the median distance from a point to the best-fit plane of its 16 nearest neighbours, is 0.0114 m against 0.0112 m for dense VGGT: the reconstructions are no thicker. What does differ is coverage, since more of the predicted depth clears the shared confidence threshold, which is why point counts are annotated in the figures.

Point clouds are exported from the evaluation path itself, and the Chamfer distance recomputed from each exported cloud reproduces the recorded value to ± 0.0000 for dense VGGT, A3D and both Zebra3D -KD configurations. FastVGGT and RegimeVGGT differ by up to 0.005, which is expected: their token-merging decisions depend on runtime state and are not bit-reproducible across runs. No ordering in Table 2 changes.

Figure 12 adds four scenes from outside the training data: an outdoor KITTI driving sequence and three Neural RGB-D rooms. Unlike the ScanNet figures, the clouds are coloured by the input images rather than by error, since this comparison is qualitative only.

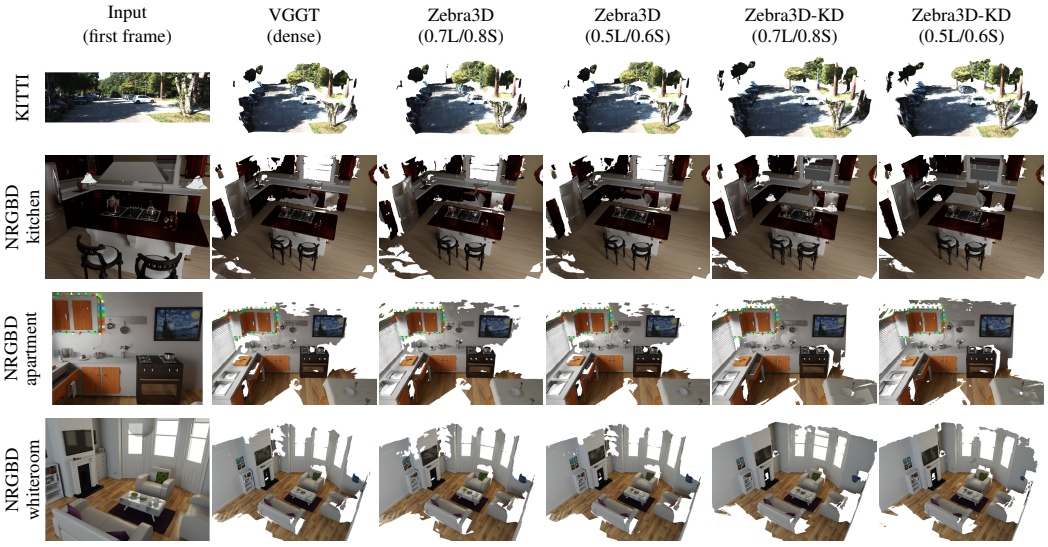


Figure 12: **Reconstructions on scenes outside the training data.** From top: 40 consecutive frames of the KITTI driving sequence `2011_09_26_drive_0023` (camera 02); 40 frames each of the Neural RGB-D scenes `complete_kitchen`, `morning_apartment` and `whiteroom`. None of these is in the training data of any model shown; VGGT's includes only the synthetic Virtual KITTI. The first column is the first input frame, and every method is rendered from one virtual camera placed behind and above it. The same lowest-confidence 40% and farthest 3% of pixels are removed from every method, so each keeps nearly the same number of points, and each point is drawn as a disc sized to the surface its source pixel covers.

F.2 Limitations

ZEBRA3D primarily benefits long sequences, where global attention dominates inference cost; speedups remain modest at 8–16 views (Figure 5). As attention becomes cheaper, the image encoder and prediction heads impose an increasing upper bound on end-to-end acceleration. Our stage allocation is fixed per backbone and derived from marginal, single-stage conversion sensitivity, so it does not model interactions among converted stages or adapt to individual inputs. Although the allocation is stable across our calibration datasets, broader generalization to substantially different architectures and domains remains to be established. Finally, our optimized kernels and benchmarks target Triton on AMD MI355X GPUs; portability and performance on other accelerators require further evaluation. Optional distillation also requires a frozen dense teacher during refinement, although it adds no inference-time cost.