MOSAICATTENTION: DUAL-POOLED SPARSE ATTENTION FOR ACCELERATING MULTI-VIEW 3D GEOMETRY TRANSFORMERS

Anonymous authors
Paper under double-blind review

ABSTRACT

Feed-forward multi-view geometry Transformers recover cameras and scene geometry from images in a single forward pass by aggregating cross-view evidence through global attention. However, the global attention scales quadratically with the number of input views, making long-sequence inference extremely expensive. Block-sparse attention here provides an intuitive way to accelerate global attention. Existing methods typically treat visual tokens as a flattened 1D sequence and represent each block through only average statistics with uniform sparsity across all layers, which overlooks the 2D spatial layout, salient variations within each block, and layer-wise varying sensitivity to block removal, leading to unreliable block-importance estimation and potential performance loss. These limitations motivate us to introduce MosaicAttention, a sparse attention framework that jointly exploits both 2D average- and maximum-pooled views for more reliable block routing. Our theoretical analysis shows that Avg preserves mean regional logits while Max captures additional within-region variation, motivating dual pooling over 2D visual regions as coarse-grained proxies of token interactions. We then calibrate these dual-pooled proxies against dense value-weighted output contributions to obtain reliable block priorities, and allocate computation according to both block importance and layer-wise sparsification sensitivity. We further bound regional logit approximation and sparse-attention output errors and characterize when calibration preserves block selection. Experiments on multiple feed-forward geometry Transformers and 3D prediction tasks show that MosaicAttention achieves a favorable accuracy-efficiency trade-off over previous methods. Especially, on inputs with 128 views, it achieves up to 2.89× end-to-end inference speedup while maintaining competitive performance.

1 INTRODUCTION

Recovering cameras and scene geometry from multiple images is a fundamental problem in 3D vision (Hartley & Zisserman, 2003). Traditional reconstruction pipelines (Agarwal et al., 2011; Schonberger & Frahm, 2016) combine specialized modules for correspondence estimation, camera recovery, and dense reconstruction with iterative geometric optimization. However, these modular pipelines remain slow and fragile: each stage is optimized independently, so errors accumulate across the chain, and the repeated iterative optimization scales poorly with the number of views. Recent learning-based approaches address this by progressively unifying these stages, moving from pairwise point-map regression in DUSt3R and MASt3R (Wang et al., 2024; Leroy et al., 2024) toward joint multi-view inference in VGGT (Wang et al., 2025b), π^3 (Wang et al., 2026b), and Depth Anything 3 (Lin et al., 2025). These feed-forward geometry Transformers directly predict cameras

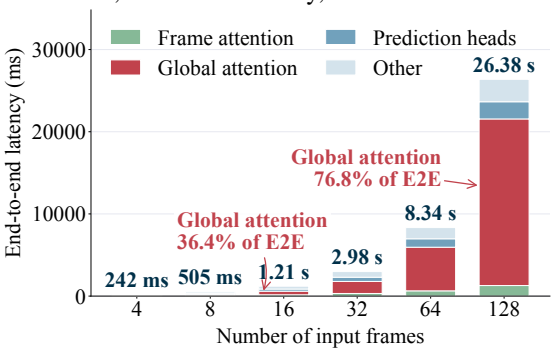


Figure 1: **Scaling bottleneck:** Global attention dominates inference latency with more input views.

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.

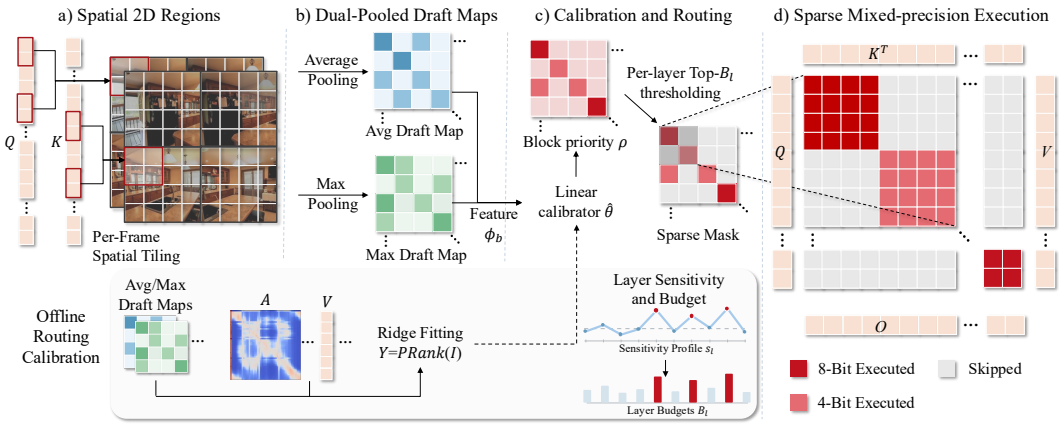


Figure 2: **Overview of MosaicAttention.** (a) Visual tokens are partitioned into compact 2D regions within each frame, forming spatially coherent units for block routing. (b) Average and maximum pooling summarize each region into two complementary low-resolution draft maps. (c) Offline calibration fits a frozen affine calibrator from proxy features ϕ_b to dense contribution ranks, and profiles layer-wise sparsification sensitivity to fix per-layer budgets B_ℓ. At inference, the predicted priorities ρ guide cross-head selection under budget B_ℓ, with one patch block reserved for each active query row. (d) A mixed-precision sparse kernel computes token-level attention on the selected patch blocks and always-retained interactions.

and scene structure from image collections, with global cross-view attention providing a shared mechanism for aggregating visual evidence across views.

However, the global attention that enables joint multi-view reasoning also limits its scalability. As shown in Figure 1, the token sequence grows linearly with the number of views, making the quadratically growing global attention a major computation bottleneck during inference. Token-reduction approaches alleviate this by merging visual tokens (Bolya et al., 2023; Shen et al., 2025b) or compressing them into compact descriptors (Wang & Xu, 2026). However, these methods modify token representations before global interaction, which may lead to the loss of fine-grained visual captures.

Meanwhile, block-sparse attention (Child et al., 2019; Beltagy et al., 2020) offers a complementary strategy: it retains original tokens while selectively removing interactions between queries, keys, and values. This strategy relies on accurate block importance estimation to skip unnecessary computations without compromising model performance (Jiang et al., 2024; Tang et al., 2024). Existing sparse variants of multi-view geometry Transformers follow two representative strategies. SparseVGGT (Wang et al., 2025a) adopts a 1D frame-major sequence layout with average pooling to downsample each query or key block into a compact representative for efficient sparse routing. SAF3R (Deng et al., 2026) instead profiles head-specific sparse kernels offline and adapts selected key indices online using kernel-dependent cues. Despite these advances, three limitations remain. (i) Contiguous 1D segments do not explicitly preserve the 2D neighborhood structure of image patches and can split spatially coherent regions at sequence boundaries. (ii) Existing routing relies on average-pooled proxy or predefined pattern-specific selection rules, without directly calibrating the importance of individual query–key blocks against their internal feature distributions and dense output contributions. (iii) A uniform sparsity ratio across all layers assumes equal tolerance to block removal, overlooking the varying importance of attention layers in geometric inference. These limitations motivate a routing mechanism that operates on spatially structured regions, leverages complementary within-region statistics, and adapts to layer-wise sparsification sensitivity.

Based on the above insights, we introduce MosaicAttention, a dual-pooled block-sparse attention framework for feed-forward multi-view geometry Transformers. Figure 2 summarizes its complete pipeline. Tokens are first reorganized into a spatially structured layout to enable coherent regional downsampling. MosaicAttention then summarizes each region using average and maximum pooling, where average pooling captures the mean regional response and maximum pooling provides a complementary view of within-region variation. Interactions between the pooled query and key representations form lightweight dual-pooled proxy maps (i.e., draft maps), providing coarse estimates of block importance without materializing the full token-level attention map. A conditional calibrator further maps these proxy features to unified block priorities aligned with dense value-

weighted output contributions. Given the calibrated priorities, offline layer-wise sensitivity determines the computation budget for each layer, while blocks across heads compete for the shared layer budget subject to row-wise connectivity constraints. Only the selected blocks undergo full token-level attention computation, allowing unimportant interactions to be skipped for efficient inference. Our analysis bounds average-pooled logit approximation error through regional feature variation and gives a score-gap condition under which calibration preserves block selection. It then bounds sparse-attention output error by the contributions and attention mass of omitted blocks.

We evaluate MosaicAttention on camera-pose estimation and dense geometry prediction across VGGT, π^3, and Depth Anything 3. Under controlled sparsity and execution-precision settings, MosaicAttention achieves a favorable accuracy–efficiency trade-off relative to average-only block routing, head-adaptive sparse attention, and token merging. On 128-view inputs, it delivers up to 2.89 $\times$ end-to-end inference speedup on π^3 at 75% sparsity, while maintaining competitive geometric accuracy. Ablation studies isolate the contributions of spatial dual pooling, priority calibration, hierarchical allocation, and mixed-precision. Our contributions are summarized as follows:

- **Spatial dual-pooled block selection.** We construct lightweight interaction proxies over compact 2D token regions using average and maximum pooling. Our theoretical analysis demonstrates that they retain complementary statistics for estimating block importance.
- **Contribution-calibrated block routing.** We introduce a lightweight conditional calibrator that maps dual-pooled features to block priorities aligned with dense-output contribution. These priorities guide joint cross-head selection under sensitivity-aware layer budgets and row-wise connectivity constraints.
- Extensive experiments demonstrate that MosaicAttention consistently pushes the accuracy-efficiency trade-off frontier compared with existing efficient baselines.

2 Related Work

Feed-forward multi-view geometry Transformers. Traditional multi-view reconstruction combines correspondence estimation, camera recovery, and bundle adjustment (Schonberger & Frahm, 2016). DUSt3R and MASt3R (Wang et al., 2024; Leroy et al., 2024) simplify this pipeline with pairwise point-map prediction, yet still require global alignment over image collections. Feed-forward models such as VGGT (Wang et al., 2025b) and its successors (Wang et al., 2026b; Lin et al., 2025; Wang et al., 2026a) instead predict the geometry of all views in a single forward pass. However, their cost of dense cross-view attention grows steeply with the number of input frames.

Efficient inference for geometry Transformers. Recent works accelerate these models through quantization (QVGGT (Pan et al., 2026)), token merging (FastVGGT (Shen et al., 2025b)), context compression for streaming inference (OVGGT (Lu et al., 2026)), and descriptor-based global aggregation (FlashVGGT (Wang & Xu, 2026)). These methods reduce cost by altering either the token set or the numerical precision. MosaicAttention instead retains all original tokens and exploits block-level sparsity within global attention, evaluating only critical query–key interactions.

Block-sparse and low-precision attention. Block-sparse attention (Child et al., 2019; Beltagy et al., 2020) reduces computation by executing only selected query–key block interactions. Recent methods predict sparse masks using lightweight approximations: learned gates (SeerAttention (Gao et al., 2024)), low-resolution attention drafts (DraftAttention (Shen et al., 2025a)), or training-free pooled representations (SpargeAttention (Zhang et al., 2025c)). These works motivate efficient draft-based routing, while leaving the design of block representations for multi-view geometric reasoning underexplored. Complementary to sparsification, low-precision kernels such as SageAttention (Zhang et al., 2025b;a) reduce the arithmetic cost of the retained computations. Within multi-view geometry Transformers, SparseVGGT (Wang et al., 2025a) builds block-sparse masks using average-pooled proxy over a flattened token sequence. SAF3R (Deng et al., 2026) selects head-specific sparse kernels through offline output-error profiling and adapts key selection to each input. MosaicAttention instead calibrates complementary Avg/Max proxies over compact 2D regions into contribution-aware block priorities for shared-budget routing, with low-precision execution providing further acceleration.

3 MOSAICATTENTION

Global attention (Vaswani et al., 2017) incurs quadratic cost as the number of input views grows. Block-sparse attention (Child et al., 2019) reduces this cost by skipping query–key blocks, but requires estimating their importance before dense token-level interactions are evaluated. MosaicAttention uses spatially pooled features to estimate block priorities and allocates a fixed computation budget according to these priorities and layer-wise sensitivity to sparsification.

3.1 FRAMEWORK OVERVIEW

We propose **MosaicAttention**. Tokens are grouped into blocks, assigned lightweight block priorities, and sparsely routed under a computation budget. A low-precision sparse kernel evaluates the selected blocks and skips the remainder. The pretrained geometry Transformer remains frozen. Model-specific special tokens are always retained, with the selectable family limited only to visual patch-patch blocks.

Block priority. We estimate block priority in two stages. First, *spatial dual-pooled proxies* summarize regional visual interactions using complementary Avg and Max features. Second, *block contribution calibration* maps these inexpensive proxy features to priorities aligned with dense value-weighted output contributions.

Sparse routing. Given calibrated block priorities, MosaicAttention allocates computation at two levels. Layer-wise sensitivity determines how much budget each layer receives, while blocks across heads compete for the shared budget within each layer, subject to row-wise connectivity constraints.

Offline calibration. A small calibration set is used to estimate the layer sensitivity profile and fit the block contribution calibrator. Both are computed offline and fixed during inference, introducing no online profiling or dense-attention overhead.

3.2 SPATIAL DUAL-POOLED PROXIES

Visual tokens are grouped into spatially coherent regions within each frame and mapped to the blocks consumed by the sparse kernel. Selected blocks retain the original token-level attention computation, whereas unselected blocks are skipped; the exact sparse-attention operator and layout are given in Appendix A.2. We write token interactions for one head, omitting batch and layer indices; routing candidates retain their head identity and share a layer budget.

Let $q_t, k_s, v_s \in \mathbb{R}^d$ denote query, key, and value vectors, with logits $\ell_{ts} = q_t^\top k_s/\sqrt{d}$, attention probabilities $A_{ts} = \exp(\ell_{ts})/\sum_r \exp(\ell_{tr})$, and dense output $O_t^{\mathrm{d}} = \sum_s A_{ts} v_s$. Sparse routing requires useful interaction estimates before dense token-pair computation is available. Thus, we adopt spatial pooling to summarize local visual features at low cost, including average and maximum pooling. Average pooling captures the mean regional response, while maximum pooling retains channel-wise extrema that averaging can suppress. Regions with the same mean can therefore have different maximum descriptors, motivating their use as complementary proxies for block routing.

Definition 3.1 (Dual-Pooled Regional Proxy)**.** *For query and key regions R_i and R_j, define the average descriptors $\bar{q}_i = |R_i|^{-1}\sum_{t\in R_i} q_t$ and $\bar{k}_j = |R_j|^{-1}\sum_{s\in R_j} k_s$. Their channel-wise maximum descriptors are $q_{i,r}^{\max} = \max_{t\in R_i} q_{t,r}$ and $k_{j,r}^{\max} = \max_{s\in R_j} k_{s,r}$. The corresponding regional interactions are*

$$\ell_{ij}^{\mathrm{avg}} = \frac{\bar{q}_i^\top \bar{k}_j}{\sqrt{d}}, \qquad \ell_{ij}^{\max} = \frac{(q_i^{\max})^\top k_j^{\max}}{\sqrt{d}}. \tag{1}$$

Lemma 3.2 (Dual-Pooling Relation)**.** *Let $\Delta q_i = q_i^{\max} - \bar{q}_i$ and $\Delta k_j = k_j^{\max} - \bar{k}_j$. Avg preserves the mean token-level logit, while Max adds variation-dependent terms:*

$$\ell_{ij}^{\mathrm{avg}} = \frac{1}{|R_i||R_j|}\sum_{t\in R_i, s\in R_j} \ell_{ts}, \qquad \ell_{ij}^{\max} - \ell_{ij}^{\mathrm{avg}} = \frac{\bar{q}_i^\top \Delta k_j + \Delta q_i^\top \bar{k}_j + \Delta q_i^\top \Delta k_j}{\sqrt{d}}. \tag{2}$$

The offsets Δq_i and Δk_j expose channel-wise extrema that regional means alone do not determine, motivating dual-pooled calibration. To quantify the fidelity of the Avg representation, let $r_i^q = \max_{t \in R_i} \|q_t - \bar{q}_i\|_2$ and $r_j^k = \max_{s \in R_j} \|k_s - \bar{k}_j\|_2$.

Theorem 3.3 (Regional Logit Approximation). *For every $t \in R_i$ and $s \in R_j$,*

$$\left|\ell_{ts} - \ell_{ij}^{\text{avg}}\right| \le \frac{\|\bar{q}_i\|_2 r_j^k + \|\bar{k}_j\|_2 r_i^q + r_i^q r_j^k}{\sqrt{d}}. \tag{3}$$

When features vary little within each region, Avg closely approximates its token-level logits. This motivates pooling over compact spatial regions, while Max supplies an additional view of within-region variation. Proofs are given in Appendix A.3. For $c \in \{\text{avg}, \max\}$, row-wise softmax gives the draft map $D_{ij}^c = \exp(\ell_{ij}^c) / \sum_{j'} \exp(\ell_{ij'}^c)$. Both maps supply the calibration features.

3.3 Block Contribution Calibration

The dual-pooled draft maps characterize regional query–key interactions. Block importance depends on the values aggregated by these interactions. We therefore use dense-output contributions as an offline calibration target, learning to predict block priorities from the inexpensive draft features. For each selectable block b, we construct a compact feature vector ϕ_b from its relative Avg/Max proxy scores, their agreement, block occupancy, and coarse input- and per-head statistics. All components are obtained from the pooled draft maps and layout metadata and therefore require no dense attention at inference time. Let block b connect query region R_i to key-token block $\mathcal{B}_b$. We apply contribution calibration to the layers equipped with the spatial calibrated router; see Appendix A.4 for details.

Definition 3.4 (Block Contribution Calibration). *For block b, define its dense-output contribution as $I_b = |R_i|^{-1} \sum_{t \in R_i} \left\| \sum_{s \in \mathcal{B}_b} A_{ts} v_s \right\|_2$. Its calibration target is $Y_b = \text{PRank}(I_b) \in [0, 1]$, the percentile rank among eligible blocks in the same layer and calibration sample. In the multi-head router, this ranking is computed jointly over eligible blocks from all heads.*

We standardize each feature dimension using calibration-set statistics and, with a slight abuse of notation, continue to denote the standardized feature by ϕ_b. Given N calibration blocks, we fit an affine ridge calibrator,

$$(\hat{\theta}, \hat{\beta}) = \arg\min_{\theta, \beta} \sum_{n=1}^{N} \left(\theta^\top \phi_n + \beta - Y_n\right)^2 + \lambda \|\theta\|_2^2, \qquad \rho_b = \hat{\theta}^\top \phi_b + \hat{\beta}, \tag{4}$$

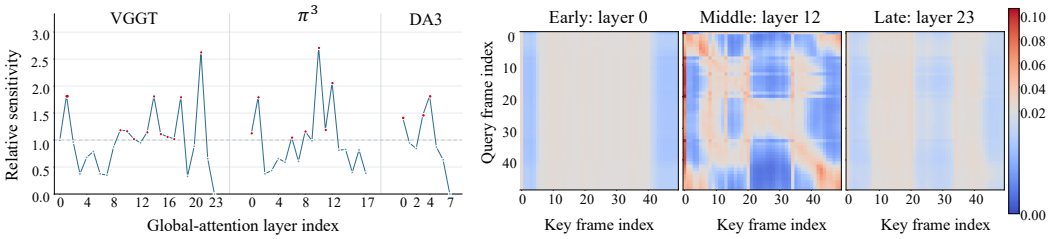
where $\lambda > 0$ and the intercept is not regularized. At inference time, the normalization is folded into an equivalent affine scorer on the raw features. Percentile ranking removes sample-dependent scale while preserving the ordering relevant to sparse selection. Appendix A.4 details the features, normalization, and rank convention, and bounds calibration error from the ridge solution. This bound connects calibration accuracy to routing stability.

3.4 Sparse Routing under Layer-Wise Budget Allocation

Layer-wise budget allocation. Block priorities determine *where* computation is spent within a layer, while the layer-wise budget determines *how much* computation each layer receives. We measure each layer's sensitivity by sparsifying that layer alone at a common reference sparsity and evaluating the resulting geometry distortion. As shown in Figure 3, global-attention layers exhibit substantially different tolerance to sparsification.

To convert this sensitivity profile into per-layer computation budgets, we first derive each layer's budget through an offline calibration dataset. The sensitivity scores are then passed through an automated offline allocator that assigns larger retained fractions to more sensitive layers and smaller fractions to less sensitive ones. The allocation respects a target global sparsity and fixed per-layer bounds, so that the resulting schedule is feasible under the row-wise reservation constraint.

The sensitivity profile and resulting budgets for each model are fixed after calibration and introduce no online profiling cost. The exact sensitivity measurement and allocation procedure are described in Appendix A.5.



(a) Layer-wise sensitivity to sparsification. (b) Cross-view interactions across network depth.

Figure 3: **Layer-wise sparsification behavior.** Non-uniform sensitivity motivates different layer budgets, while changing interaction patterns motivate adaptive routing within each layer.

Sparse routing. Given B_ℓ, candidate blocks compete globally across heads according to their calibrated priorities ρ_b. To guarantee a valid sparse support, the highest-priority patch block is first reserved for every active head/query row. Let $\mathcal{C}_\ell$ denote the eligible patch-block candidates and $\mathcal{R}_\ell$ the resulting reserved set. The allocator guarantees $B_\ell \geq |\mathcal{R}_\ell|$.

For a candidate set $\mathcal{C}$, let $\mathrm{TopK}_B(\rho; \mathcal{C}) \subseteq \mathcal{C}$ denote the B blocks with largest priorities ρ_b. The remaining layer budget is allocated as

$$\widehat{\mathcal{S}}_\ell = \mathcal{R}_\ell \cup \mathrm{TopK}_{B_\ell - |\mathcal{R}_\ell|}(\rho; \mathcal{C}_\ell \setminus \mathcal{R}_\ell). \tag{5}$$

For each Top-K stage $(\mathcal{C}, B)$, define $\mathcal{S}_B^\star = \mathrm{TopK}_B(Y; \mathcal{C})$ and $\widehat{\mathcal{S}}_B = \mathrm{TopK}_B(\rho; \mathcal{C})$.

Theorem 3.5 (Top-K Routing Stability)**.** *Suppose $1 \leq B < |\mathcal{C}|$ and $|\rho_b - Y_b| \leq \varepsilon$ for all $b \in \mathcal{C}$. Then $\widehat{\mathcal{S}}_B = \mathcal{S}_B^\star$ if $Y_{(B)} - Y_{(B+1)} > 2\varepsilon$, where $Y_{(r)}$ is the r-th largest target score in $\mathcal{C}$.*

Calibration can shrink a selected–unselected score gap by at most 2ε. The condition applies first to row-wise reservations and, once these agree, to the remaining global selection. Appendix A.5 proves the result; Lemma A.1 bounds ε.

Output error. For one head and query region R_i sharing a sparse support, let $\mathcal{C}_i$ be its disjoint selectable key blocks and $\mathcal{S} \subseteq \mathcal{C}_i$ the retained set. Define the average dense attention mass $M_b = |R_i|^{-1} \sum_{t \in R_i} \sum_{s \in \mathcal{B}_b} A_{ts}$. Let O_t^{s} be attention renormalized over the retained blocks and all always-retained interactions.

Lemma 3.6 (Blockwise Sparsification Error)**.** *If every query retains a nonempty support and $\|v_s\|_2 \leq V_{\max}$ for every valid key token, then*

$$\frac{1}{|R_i|} \sum_{t \in R_i} \left\| O_t^{\mathrm{d}} - O_t^{\mathrm{s}} \right\|_2 \leq \sum_{b \in \mathcal{C}_i \setminus \mathcal{S}} (I_b + V_{\max} M_b). \tag{6}$$

These terms bound omitted contributions and renormalization. At each nontrivial ranking stage, Theorem 3.5 ensures that routing recovers the contribution-ranked support, whose omitted mass bounds the error since $I_b \leq V_{\max} M_b$. Appendix A.6 gives the proof and combined bound.

Remark 3.7 (Low-Precision Execution)**.** *If reduced-precision execution contributes at most ε_q under the same region-averaged output metric, the RHS of Equation* (13) *increases additively by ε_q.*

3.5 Low-Precision Sparse Execution

Block sparsification reduces the number of executed interactions, while low-precision execution reduces the arithmetic cost of each retained interaction. We decouple these two sources of efficiency: the dual-pooled proxies, calibrated priorities, and sparse masks are all computed in FP16, and quantization is applied only to the retained blocks after $\widehat{\mathcal{S}}_\ell$ has been determined.

Retained blocks are evaluated with hardware-efficient quantized kernels. Bit-widths follow the calibrated priorities: within each layer, the top half of the retained blocks are executed in 8-bit and the remaining half in 4-bit, concentrating quantization error on the least important interactions. By Remark 3.7, this error enters the output bound only as an additive term ε_q, while the omitted-attention-mass term remains controlled by routing.

Table 1: Camera pose estimation on CO3Dv2 (Reizenstein et al., 2021). **Bold**: better than SparseVGGT (Wang et al., 2025a) at matched sparsity; highlight: best among all efficient methods.

Model	Sparsity	Method	10 Frames		30 Frames		100 Frames	
			AUC @30↑	Latency (s) (Speedup)	AUC @30↑	Latency (s) (Speedup)	AUC @30↑	Latency (s) (Speedup)
VGGT	0%	Dense	90.03	0.529 (1.000×)	90.47	2.290 (1.000×)	90.67	12.21 (1.000×)
	50%	Ours	**88.66**	0.505 (1.048×)	**89.21**	**1.603(1.429×)**	**89.60**	**5.619 (2.17×)**
		SparseVGGT	86.12	**0.478 (1.107×)**	86.48	1.626 (1.408×)	87.81	6.566 (1.859×)
	75%	Ours	**85.95**	0.502 (1.054×)	**87.34**	**1.511 (1.516×)**	87.91	**4.937 (2.473×)**
		SparseVGGT	82.50	**0.456 (1.160×)**	83.64	1.520 (1.506×)	86.23	6.344 (1.924×)
	85%	SAF3R	88.25	0.683 (0.775×)	89.18	1.974 (1.161×)	89.52	6.193 (1.971×)
	90% (Merge)	FastVGGT	85.79	0.656 (0.808×)	87.45	2.066 (1.107×)	88.63	6.618 (1.845×)
π^3	0%	Dense	90.38	0.491 (1.000×)	90.72	2.203 (1.000×)	91.10	12.10 (1.000×)
	50%	Ours	**89.59**	0.485 (1.012×)	**89.97**	**1.457 (1.512×)**	**90.59**	**5.106 (2.370×)**
		SparseVGGT	86.80	**0.477 (1.029×)**	87.21	1.687 (1.306×)	87.26	7.165 (1.689×)
	75%	Ours	80.62	0.470 (1.045×)	**83.06**	**1.378 (1.599×)**	**84.95**	**4.327 (2.796×)**
		SparseVGGT	**81.49**	**0.467 (1.051×)**	82.87	1.491 (1.478×)	83.48	5.478 (2.209×)
	85%	SAF3R	89.48	0.469 (1.047×)	89.37	1.721 (1.280×)	90.38	5.084 (2.380×)
	90% (Merge)	FastVGGT	85.66	0.561 (0.875×)	88.65	1.899 (1.160×)	89.16	6.338 (1.909×)
DA3	0%	Dense	87.57	0.383 (1.000×)	88.25	1.287 (1.000×)	88.56	5.439 (1.000×)
	50%	Ours	**86.84**	0.366 (1.046×)	**87.69**	**1.087 (1.184×)**	**88.47**	**3.677 (1.479×)**
		SparseVGGT	85.42	**0.353 (1.085×)**	86.40	1.171 (1.098×)	87.59	4.295 (1.266×)
	75%	Ours	**82.44**	0.362 (1.058×)	**84.92**	**1.045 (1.232×)**	**86.80**	**3.145 (1.729×)**
		SparseVGGT	81.48	**0.351 (1.091×)**	84.83	1.105 (1.165×)	86.54	3.594 (1.513×)
	90% (Merge)	FastVGGT	84.51	0.389 (0.984×)	87.20	1.234 (1.043×)	87.58	3.961 (1.373×)

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Models and tasks. We evaluate MosaicAttention on three representative feed-forward multi-view geometry Transformers, including VGGT-1B (Wang et al., 2025b), π^3 (Wang et al., 2026b), and DA3-Large (Depth Anything 3) (Lin et al., 2025). All backbone parameters remain frozen, and MosaicAttention is applied only to their global-attention modules. Each backbone is evaluated on both camera-pose estimation and dense geometry prediction. See Appendix B for full details.

Evaluation settings. Camera-pose estimation is evaluated on CO3Dv2 (Reizenstein et al., 2021) with 10, 30, and 100 input frames, reporting AUC@30. Dense geometry prediction is assessed on 7 Scenes (Shotton et al., 2013), NRGBD (Azinović et al., 2022), and ETH3D (Schops et al., 2017), reporting depth inlier ratio δ., point-cloud accuracy (Acc.) and completeness (Comp.).

Baselines. We compare the dense FP16 models and three efficient inference methods: SparseVGGT (Wang et al., 2025a), building block-sparse masks from 1D average-pooled descriptors; SAF3R (Deng et al., 2026), selecting head-specific sparse attention policies via offline profiling and online adaptation; and FastVGGT (Shen et al., 2025b), reducing attention cost via token merging. MosaicAttention and SparseVGGT adopt an 8-bit kernel for a fair comparison, while FastVGGT retains FP16 execution and SAF3R retains BF16 execution following their official implementations. FlashAttention-2 (Dao, 2024) is enabled for all supported dense-attention operations.

4.2 MAIN RESULTS

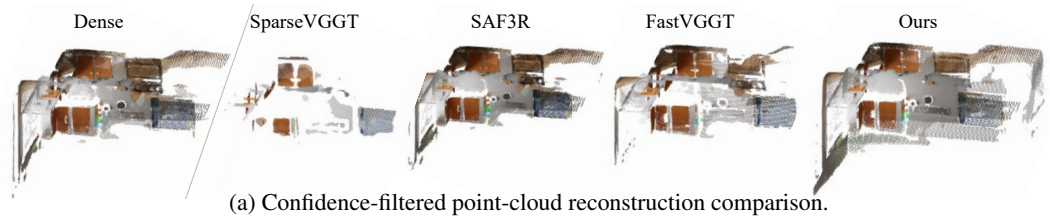
Camera-pose estimation. We first evaluate MosaicAttention on camera-pose estimation across different backbones, input lengths, and sparsity in Table 1. Under matched sparsity and 8-bit execution, MosaicAttention consistently preserves pose accuracy better than SparseVGGT, confirming the benefit of spatial dual-pooled and contribution-calibrated routing. At 50% sparsity, it consistently outperforms SAF3R at 85% sparsity across all VGGT and π^3 configurations, with lower latency on longer sequences. Token merging requires an aggressive merge ratio for meaningful end-to-end acceleration, so we compare FastVGGT at its recommended 0.9 merge ratio. Even at this operating point, MosaicAttention (50% sparsity) achieves higher AUC with lower end-to-end latency across all evaluated model–frame configurations. Raising the sparsity to 75% further reduces latency, while the accuracy degradation remains moderate for VGGT and DA3 and becomes more pronounced for

Table 2: Depth estimation and dense reconstruction on 7 Scenes (Shotton et al., 2013), NRGBD (Azinović et al., 2022) and ETH3D (Schops et al., 2017).

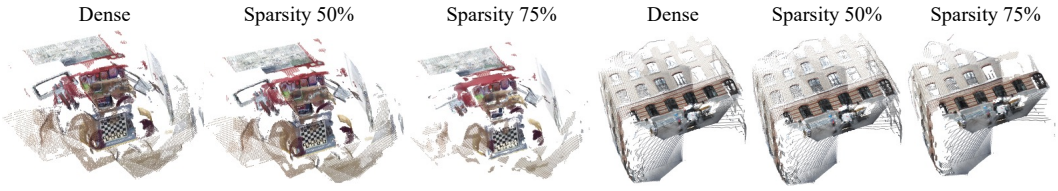
Model	Sparsity	Method	7Scenes			NRGBD			ETH3D		
			$\delta\uparrow$	Acc.↓	Comp.↓	$\delta\uparrow$	Acc.↓	Comp.↓	$\delta\uparrow$	Acc.↓	Comp.↓
VGGT	50%	Ours	**95.97**	**0.0187**	**0.0246**	**99.27**	**0.0158**	**0.0123**	**98.94**	**0.1465**	**0.1087**
		SparseVGGT	95.96	0.0193	0.0261	99.19	0.0197	0.0153	98.56	0.1974	0.1285
	75%	Ours	**96.64**	**0.0191**	**0.0242**	**99.21**	**0.0225**	**0.0162**	**98.52**	**0.2124**	**0.1402**
		SparseVGGT	95.01	0.0205	0.0271	99.20	0.0231	0.0170	98.47	0.2645	0.2104
	85%	SAF3R	96.03	0.0189	0.0267	99.18	0.0198	0.0155	98.58	0.1497	0.1048
	90% (Merge)	FastVGGT	96.07	0.0189	0.0244	99.23	0.0252	0.0173	98.91	0.1905	0.1032
π^3	50%	Ours	**98.76**	**0.0131**	**0.0139**	**99.22**	**0.0247**	**0.0129**	**99.76**	**0.0966**	**0.0500**
		SparseVGGT	98.67	0.0185	0.0163	99.18	0.0413	0.0193	99.69	0.1569	0.0918
	75%	Ours	**98.68**	**0.0256**	**0.0162**	**99.17**	**0.0586**	**0.0252**	**99.76**	**0.2374**	0.1238
		SparseVGGT	98.68	0.0282	0.0184	99.13	0.0738	0.0319	99.64	0.2559	**0.1235**
	85%	SAF3R	98.65	0.0141	0.0146	99.19	0.0176	0.0157	99.71	0.1475	0.0552
	90% (Merge)	FastVGGT	98.69	0.0139	0.0138	99.20	0.0197	0.0123	99.65	0.1334	0.0767
DA3	50%	Ours	**97.76**	**0.0187**	**0.0210**	**99.19**	**0.0394**	**0.0802**	**99.32**	**0.1496**	**0.0702**
		SparseVGGT	97.67	0.0207	0.0219	99.13	0.0545	0.0897	99.21	0.1703	0.1035
	75%	Ours	**97.77**	**0.0214**	**0.0206**	**99.13**	**0.0369**	**0.0528**	**99.19**	**0.1804**	**0.0874**
		SparseVGGT	97.68	0.0226	0.0217	99.12	0.0465	0.0565	99.17	0.1852	0.1029
	90% (Merge)	FastVGGT	97.59	0.0203	0.0221	99.12	0.0327	0.0247	98.83	0.1521	0.0734

π^3. Beyond the 8-bit comparison, the retained blocks can be executed with mixed FP4/FP8 sparse kernels for further acceleration, particularly on long input sequences, as shown in Figure 5.

Dense geometry prediction. We further evaluate depth estimation and point-map reconstruction on the indoor 7 Scenes and NRGBD and the outdoor ETH3D benchmarks in Table 2. Under matched sparsity and 8-bit execution, MosaicAttention improves nearly all metrics over SparseVGGT across all three backbones. Following Table 1, FastVGGT and SAF3R are evaluated at their official implementations for comparable acceleration. They remain competitive on selected metrics in 16-bit, notably on NRGBD with π^3. Nevertheless, we achieve higher depth accuracy in most configurations and improve more consistently over the matched baseline, indicating that calibrated routing better preserves geometry-relevant interactions under limited attention budgets.



(a) Confidence-filtered point-cloud reconstruction comparison.



(b) Qualitative results across sparsity levels.

Figure 4: **Visualization results.** (a) At 75% sparsity, our method achieves the best performance. (b) Our method preserves both scene layout and fine structures as sparsity increases from 50% to 75%.

Visualization results. Qualitative reconstructions in Figure 4 are consistent with these quantitative results. See Appendix E for more visualization results.

4.3 Ablation Study

2D Regional routing and contribution calibration. Table 3 ablates the regional proxy and the contribution calibration. Max-only collapses across all tasks, showing that within-region extremes alone cannot characterize block importance, while Avg-only preserves the attention output much better; fusing the two nevertheless improves pose accuracy over Avg-only, confirming that the two

statistics are complementary. Naively fusing them—ranking blocks directly by the combination of Avg and Max draft scores without calibration—improves pose accuracy over Avg-only, revealing a misalignment between pooled interaction scores and actual block contributions. Pooling over 2D regions yields a more faithful proxy, yet still fails to convert this faithfulness into better routing. Calibrating the dual-pooled features against dense-output contribution ranks closes this gap, yielding the best pose and reconstruction accuracy together with the lowest attention-output error.

Table 3: Ablation of regional routing proxies and block contribution calibration. All variants evaluated on VGGT-1B with 10 input frames, 50% sparsity.

Method	CO3Dv2 AUC@30 ↑	7Scenes Acc.↓	NRGBD Acc.↓	ETH3D Acc.↓	Attn. Output Relative ℓ_2 ↓
Avg-Only	86.84	0.0193	0.0197	0.1944	0.0956
Max-Only	75.56	0.0492	0.0664	0.4045	0.2581
1D Naive Fusion	88.12	0.0192	0.0173	0.1661	0.1165
2D Naive Fusion	87.79	0.0192	0.0179	0.1699	0.0936
Ours	**88.66**	**0.0187**	**0.0158**	**0.1465**	**0.0785**

Table 4: Ablation of layer-wise budget allocation under the same total block budget.

Method	CO3Dv2 AUC@30 ↑	7Scenes Acc.↓	NRGBD Acc.↓	ETH3D Acc.↓
Uniform	87.58	0.0199	0.0188	0.1673
Concentration	87.71	0.0197	0.0190	0.1698
Sensitivity	**88.66**	**0.0187**	**0.0158**	**0.1465**

Table 5: Ablation of execution-precision.

Precision	CO3Dv2 AUC@30 ↑	7Scenes Acc.↓	NRGBD Acc.↓	ETH3D Acc.↓	Latency
FP16	88.73	0.0185	0.0156	0.1462	0.533
FP8	88.66	0.0187	0.0158	0.1465	0.505
50%FP4+ 50%FP8	88.45	0.0188	0.0158	0.1468	0.491

Layer-wise budget allocation. Table 4 compares allocation strategies under an identical total block budget. Uniform allocation ignores heterogeneous layer sensitivities, and attention concentration does not reflect a layer's actual reliance on the retained tokens. Sensitivity-aware allocation directly measures each layer's tolerance to block removal and consistently outperforms both baselines.

Execution precision. Table 5 evaluates the effect of sparse-attention precision under fixed routing and block budgets. FP8 reduces latency with negligible accuracy loss, while mixed FP4/FP8 further improves efficiency with only minor degradation. These results motivate the mixed-precision setting used in our full-system evaluation. More results are provided in Appendix C.

4.4 Efficiency Analysis

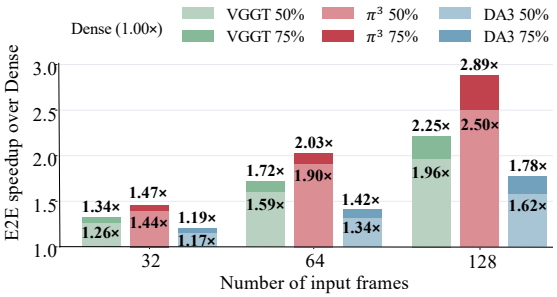


Figure 5: End-to-end speedup of MosaicAttention with mixed 4/8-bit sparse execution over the dense FP16 backbone at different numbers of input views.

Figure 5 reports the end-to-end speedup of the complete MosaicAttention system, combining sparse routing with mixed 4/8-bit execution, over the dense FP16 backbones. The speedup increases with the number of input views across all evaluated backbones and sparsity levels, since global attention accounts for a growing share of the total runtime as shown in Figure 1. This scaling behavior confirms that MosaicAttention is most beneficial exactly in the long-sequence regime where feed-forward geometry Transformers are the most expensive.

5 Conclusion

We presented MosaicAttention, a block-sparse attention framework that constructs complementary average- and maximum-pooled lightweight routing proxies over compact 2D visual regions, and calibrates these proxy features against value-weighted output contributions. A hierarchical allocator distributes a fixed computational budget according to layer-wise sparsification sensitivity and block priorities shared across attention heads. Experiments across multiple geometry Transformers and camera-pose, depth, and pointmap prediction tasks show that MosaicAttention consistently outperforms average-only block routing under matched sparsity and precision, and achieves both higher accuracy and lower latency than token merging. Its practical advantage becomes increasingly pronounced for long multi-view sequences, where global attention dominates inference cost.

AI USE STATEMENT

The authors used AI-based tools to assist with language polishing of the manuscript and with code development tasks such as debugging. All AI-assisted code was reviewed and validated by the authors. The research ideas, method design, experimental protocols, analysis, and conclusions are entirely the authors' own, and the authors take full responsibility for the content of this paper.

REFERENCES

Sameer Agarwal, Yasutaka Furukawa, Noah Snavely, Ian Simon, Brian Curless, Steven M Seitz, and Richard Szeliski. Building rome in a day. *Communications of the ACM*, 2011.

Dejan Azinović, Ricardo Martin-Brualla, Dan B Goldman, Matthias Nießner, and Justus Thies. Neural rgb-d surface reconstruction. In *CVPR*, 2022.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. Longformer: The long-document transformer. *arXiv:2004.05150*, 2020.

Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. Token merging: Your ViT but faster. In *ICLR*, 2023.

Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. Generating long sequences with sparse transformers. *arXiv preprint arXiv:1904.10509*, 2019.

Tri Dao. FlashAttention-2: Faster attention with better parallelism and work partitioning. In *ICLR*, 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. *arXiv preprint arXiv:2607.03612*, 2026.

Yizhao Gao, Zhichen Zeng, Dayou Du, Shijie Cao, Hayden Kwok-Hay So, Ting Cao, Fan Yang, and Mao Yang. Seerattention: Learning intrinsic sparse attention in your llms. *arXiv preprint arXiv:2410.13276*, 2024.

Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2003.

Huiqiang Jiang, Yucheng Li, Chengruidong Zhang, Qianhui Wu, Xufang Luo, Surin Ahn, Zhenhua Han, Amir H. Abdi, Dongsheng Li, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. MInference 1.0: Accelerating pre-filling for long-context LLMs via dynamic sparse attention. In *NeurIPS*, 2024.

Vincent Leroy, Yohann Cabon, and Jerome Revaud. Grounding image matching in 3d with mast3r. In *ECCV*, 2024.

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

Si-Yu Lu, Po-Ting Chen, Hui-Che Hsu, Sin-Ye Jhong, Wen-Huang Cheng, and Yung-Yao Chen. Ovggt: O(1) constant-cost streaming visual geometry transformer. *arXiv preprint arXiv:2603.05959*, 2026.

Zhizhen Pan, Hesong Wang, and Huan Wang. Qvggt: Post-training quantized visual geometry grounded transformer. In *CVPR*, 2026.

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 *ICCV*, 2021.

Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *CVPR*, 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 *CVPR*, 2017.

Xuan Shen, Chenxia Han, Yufa Zhou, Yanyue Xie, Yifan Gong, Quanyi Wang, Yiwei Wang, Yanzhi Wang, Pu Zhao, and Jiuxiang Gu. Draftattention: Fast video diffusion via low-resolution attention guidance. *arXiv preprint arXiv:2505.14708*, 2025a.

You Shen, Zhipeng Zhang, Yansong Qu, and Liujuan Cao. Fastvggt: Training-free acceleration of visual geometry transformer. *arXiv preprint arXiv:2509.02560*, 2025b.

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 *CVPR*, 2013.

Jiaming Tang, Yilong Zhao, Kan Zhu, Guangxuan Xiao, Baris Kasikci, and Song Han. Quest: Query-aware sparsity for efficient long-context llm inference. In *ICML*, 2024.

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

Chung-Shien Brian Wang, Christian Schmidt, Jens Piekenbrinck, and Bastian Leibe. Block-Sparse Global Attention for Efficient Multi-View Geometry Transformers. *arXiv preprint arXiv:2509.07120*, 2025a.

Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *CVPR*, 2025b.

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

Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *CVPR*, 2024.

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

Zipeng Wang and Dan Xu. Flashvggt: Efficient and scalable visual geometry transformers with compressed descriptor attention. In *CVPR*, 2026.

Jintao Zhang, Haofeng Huang, Pengle Zhang, Jia Wei, Jun Zhu, and Jianfei Chen. Sageattention2: Efficient attention with thorough outlier smoothing and per-thread int4 quantization. In *ICML*, 2025a.

Jintao Zhang, Jia Wei, Pengle Zhang, Jun Zhu, and Jianfei Chen. Sageattention: Accurate 8-bit attention for plug-and-play inference acceleration. In *ICLR*, 2025b.

Jintao Zhang, Chendong Xiang, Haofeng Huang, Jia Wei, Haocheng Xi, Jun Zhu, and Jianfei Chen. Spargeattn: Accurate sparse attention accelerating any model inference. In *ICML*, 2025c.

A Theory and Implementation Details

A.1 Main Symbols

Table 6: Main notation used in the theoretical analysis.

Symbol	Description
q_t, k_s, v_s	Query, key, and value vectors
A_{ts}	Dense attention probability
$O_t^{\mathrm{d}}, O_t^{\mathrm{s}}$	Dense and sparse attention outputs
R_i	Spatial routing region
$b, \mathcal{B}_b$	Selectable block and its key-token support
ψ_b, ϕ_b	Raw augmented and standardized routing features
ρ_b	Calibrated block priority
I_b	Dense-output contribution score
Y_b	Percentile-ranked calibration target
s_ℓ	Layer sparsification sensitivity
B_ℓ	Layer block budget
M_b	Dense attention mass assigned to block b
$\mathcal{C}_i$	Selectable key-block candidates for query region R_i
$\mathcal{S}$	Retained selectable-block set

A.2 Block-Sparse Global Attention

The analysis is written for a single attention head and omits batch and layer indices. In the full multi-head implementation, each selectable block additionally carries its head identity, and routing candidates from all heads share the layer budget.

For query token t and key token s, dense attention is

$$\ell_{ts} = \frac{q_t^\top k_s}{\sqrt{d}}, \qquad A_{ts} = \frac{\exp(\ell_{ts})}{\sum_r \exp(\ell_{tr})}, \qquad O_t^{\mathrm{d}} = \sum_s A_{ts} v_s. \tag{7}$$

Given a subset Ω_t of the valid key-token support, sparse attention retains the original token-level interactions inside Ω_t and renormalizes over the retained support:

$$O_t^{\mathrm{s}} = \sum_{s \in \Omega_t} \frac{\exp(\ell_{ts})}{\sum_{r \in \Omega_t} \exp(\ell_{tr})} v_s. \tag{8}$$

Equation (8) defines the exact-arithmetic sparse operator on the predicted support, using the original token-level logits and values.

Sparse layout. Visual tokens are partitioned independently within each frame into spatially connected logical regions. Pooling is applied to the same normalized and position-encoded query and key vectors that enter the dense attention logits. Logical regions are then deterministically lowered to the physical blocks consumed by the sparse kernel. When one logical region maps to multiple physical key blocks, its regional proxy statistics are inherited by the corresponding physical candidates before calibration and routing. Padding tokens are excluded from both pooling and attention.

Routing is applied to selectable visual-patch interactions. Model-specific special queries remain dense, and special-key interactions are always retained. These fixed interactions are therefore outside the selectable block family analyzed below.

A.3 Dual-Pooled Regional Approximation

For query and key regions R_i and R_j, let $\bar{q}_i = |R_i|^{-1} \sum_{t \in R_i} q_t$ and $\bar{k}_j = |R_j|^{-1} \sum_{s \in R_j} k_s$. Their channel-wise maximum descriptors are $q_{i,r}^{\max} = \max_{t \in R_i} q_{t,r}$ and $k_{j,r}^{\max} = \max_{s \in R_j} k_{s,r}$.

Proof of Lemma 3.2. By bilinearity,

$$\frac{\bar{q}_i^\top \bar{k}_j}{\sqrt{d}} = \frac{1}{|R_i||R_j|} \sum_{t \in R_i} \sum_{s \in R_j} \frac{q_t^\top k_s}{\sqrt{d}},$$

which proves the first identity in Lemma 3.2.

For the second identity, write $q_i^{\max} = \bar{q}_i + \Delta q_i$ and $k_j^{\max} = \bar{k}_j + \Delta k_j$. Expanding their inner product gives

$$\ell_{ij}^{\max} = \ell_{ij}^{\text{avg}} + \frac{\bar{q}_i^\top \Delta k_j + \Delta q_i^\top \bar{k}_j + \Delta q_i^\top \Delta k_j}{\sqrt{d}},$$

which proves the second identity in Lemma 3.2. □

Proof of Theorem 3.3. For $t \in R_i$ and $s \in R_j$, write

$$q_t = \bar{q}_i + \epsilon_t^q, \qquad k_s = \bar{k}_j + \epsilon_s^k.$$

Then

$$\ell_{ts} - \ell_{ij}^{\text{avg}} = \frac{\bar{q}_i^\top \epsilon_s^k + (\epsilon_t^q)^\top \bar{k}_j + (\epsilon_t^q)^\top \epsilon_s^k}{\sqrt{d}}.$$

By definition, $\|\epsilon_t^q\|_2 \le r_i^q$ and $\|\epsilon_s^k\|_2 \le r_j^k$. Applying Cauchy–Schwarz to the three terms yields

$$\left|\ell_{ts} - \ell_{ij}^{\text{avg}}\right| \le \frac{\|\bar{q}_i\|_2 r_j^k + \|\bar{k}_j\|_2 r_i^q + r_i^q r_j^k}{\sqrt{d}},$$

which is Equation (3). □

The Avg and Max regional interaction maps are normalized independently over key regions before routing features are constructed. These results explain the roles of the pooled descriptors: Avg preserves the mean interaction, and Max adds information about within-region variation. The resulting draft maps provide the features used by the contribution calibrator.

A.4 Block Contribution Calibration

Routing features. For each candidate block b, the local block features contain Avg/Max percentile ranks within the query row and within the corresponding head, their agreement and joint rank, and the valid-token occupancy of the physical key block:

$$f_b = \left[r_b^{\text{avg,row}}, r_b^{\text{max,row}}, r_b^{\text{avg,head}}, r_b^{\text{max,head}}, a_b^{\text{agree}}, a_b^{\text{joint}}, a_b^{\text{occ}}\right]^\top,$$

where

$$a_b^{\text{agree}} = 1 - |r_b^{\text{avg,row}} - r_b^{\text{max,row}}|, \qquad a_b^{\text{joint}} = r_b^{\text{avg,row}} r_b^{\text{max,row}}.$$

The contextual features summarize input geometry, block occupancy, and the Avg/Max proxy distributions of the corresponding head. Their dynamic components are

$$c^{\text{dyn}} = \left[E^{\text{avg}}, E^{\max}, D^{\text{rank}}, T^{\text{avg}}, T^{\max}\right]^\top,$$

where E^c denotes normalized proxy entropy, D^{rank} denotes Avg–Max rank disagreement, and T^c measures concentration among the highest-scoring proxy entries. Static context contains normalized input-layout and occupancy statistics.

The conditional scorer is linear in the augmented raw feature

$$\psi_b = \left[f_b,\ c,\ f_b \otimes c^{\text{dyn}},\ \text{onehot}(\text{head}(b))\right]^\top.$$

Feature normalization and fitting. Let μ and σ denote the calibration-set mean and standard deviation of the augmented features, applied elementwise. Constant feature dimensions can be omitted after centering; we write the normalization for dimensions with $\sigma_j > 0$. The feature used by the ridge calibrator is

$$\phi_b = \mathrm{diag}(\sigma)^{-1}(\psi_b - \mu).$$

Thus the notation ϕ_b in the main text refers to the standardized calibration feature.

An unregularized intercept is fitted jointly with ridge-regularized feature weights, as in Equation (4). After calibration, the normalization can be folded algebraically into the stored affine scorer. Specifically, if $\rho_b = \hat{\theta}^\top \phi_b + \hat{\beta}$, then inference can equivalently evaluate

$$\rho_b = w^\top \psi_b + b_0, \qquad w = \mathrm{diag}(\sigma)^{-1}\hat{\theta}, \qquad b_0 = \hat{\beta} - w^\top \mu.$$

Hence feature standardization incurs no additional routing operation at inference.

The spatially calibrated router is applied only to the layers designated by the frozen deployment profile. Other global-attention layers retain their corresponding fixed routing rule under the same sensitivity-derived layer-budget schedule.

Percentile-rank target. For the eligible candidate set $\mathcal{C}$ of one layer and calibration sample, including eligible candidates from all heads, let $\mathrm{rank}_{\uparrow,\mathcal{C}}(I_b)$ denote the ascending rank of I_b under deterministic stable tie-breaking. For $|\mathcal{C}| > 1$,

$$\mathrm{PRank}_{\mathcal{C}}(I_b) = \frac{\mathrm{rank}_{\uparrow,\mathcal{C}}(I_b) - 1}{|\mathcal{C}| - 1}.$$

For $|\mathcal{C}| = 1$, we set $\mathrm{PRank}_{\mathcal{C}}(I_b) = 1$.

Thus the smallest contribution receives rank 0, and the largest receives rank 1 whenever $|\mathcal{C}| > 1$. With the same deterministic tie-breaking, PRank preserves the induced ordering of I_b exactly.

Ridge formulation. Let $\Phi \in \mathbb{R}^{N\times p}$ denote the standardized calibration design matrix whose n-th row is $\phi_n^\top$. Because each feature dimension is centered during standardization, $\mathbf{1}^\top \Phi = 0$. With an unregularized intercept, the optimal fitted intercept is therefore the mean target

$$\bar{Y} = \frac{1}{N}\sum_{n=1}^{N} Y_n.$$

Define the centered target vector $\widetilde{Y} = Y - \bar{Y}\mathbf{1}$. The feature weights then satisfy

$$\hat{\theta} = \left(\Phi^\top \Phi + \lambda I\right)^{-1} \Phi^\top \widetilde{Y}.$$

To characterize the prediction error, write $\widetilde{Y} = \Phi\theta_\star + \xi$ and define $G_\lambda = \Phi^\top\Phi + \lambda I$, with $\lambda > 0$.

Lemma A.1 (Calibration Prediction Error)**.** *For any candidate block b, suppose $Y_b - \bar{Y} = \phi_b^\top \theta_\star + \xi_b$. Then*

$$|\rho_b - Y_b| \le \|\phi_b\|_{G_\lambda^{-1}} \left(\|\Phi^\top \xi\|_{G_\lambda^{-1}} + \sqrt{\lambda}\|\theta_\star\|_2\right) + |\xi_b|, \tag{9}$$

where $\|x\|_M = \sqrt{x^\top M x}$.

Proof of Lemma A.1. Write

$$\widetilde{Y} = \Phi\theta_\star + \xi, \qquad G_\lambda = \Phi^\top\Phi + \lambda I.$$

Then

$$\hat{\theta} - \theta_\star = G_\lambda^{-1}\Phi^\top \xi - \lambda G_\lambda^{-1}\theta_\star.$$

For a candidate block b, suppose

$$Y_b - \bar{Y} = \phi_b^\top \theta_\star + \xi_b.$$

Since $\rho_b = \bar{Y} + \phi_b^\top \hat{\theta}$,

$$|\rho_b - Y_b| = \left|\phi_b^\top \hat{\theta} - (Y_b - \bar{Y})\right| \tag{10}$$

$$\leq \left|\phi_b^\top G_\lambda^{-1} \Phi^\top \xi\right| + \lambda \left|\phi_b^\top G_\lambda^{-1} \theta_\star\right| + |\xi_b| \tag{11}$$

$$\leq \|\phi_b\|_{G_\lambda^{-1}} \|\Phi^\top \xi\|_{G_\lambda^{-1}} + \lambda \|\phi_b\|_{G_\lambda^{-1}} \|\theta_\star\|_{G_\lambda^{-1}} + |\xi_b|. \tag{12}$$

Since $G_\lambda \succeq \lambda I$ for $\lambda > 0$,

$$\|\theta_\star\|_{G_\lambda^{-1}} \leq \frac{\|\theta_\star\|_2}{\sqrt{\lambda}}.$$

Substituting this relation gives

$$|\rho_b - Y_b| \leq \|\phi_b\|_{G_\lambda^{-1}} \left(\|\Phi^\top \xi\|_{G_\lambda^{-1}} + \sqrt{\lambda} \|\theta_\star\|_2 \right) + |\xi_b|,$$

which proves Equation (9). □

A.5 Layer-Wise Budget Allocation and Global Routing

Layer sensitivity. Let F denote the dense model and let F_ℓ^{ref} denote the model in which only global-attention layer ℓ is sparsified at a common reference sparsity. On calibration set $\mathcal{D}_{\mathrm{cal}}$, we measure

$$s_\ell = \frac{1}{|\mathcal{D}_{\mathrm{cal}}|} \sum_{x \in \mathcal{D}_{\mathrm{cal}}} d\big(F(x), F_\ell^{\mathrm{ref}}(x)\big),$$

where d measures distortion in the geometry predictions. Larger s_ℓ indicates lower tolerance to sparsification.

A monotone allocator converts the sensitivity profile into retained fractions and integer budgets satisfying $\sum_\ell B_\ell = B_{\mathrm{total}}$. More sensitive layers receive a larger retained fraction. The sensitivity profile and resulting budgets are frozen before deployment.

Constrained global routing. Within layer ℓ, let $\mathcal{C}_\ell$ denote the eligible patch-block candidates. For every active head/query row, the highest-priority patch block is first reserved; let $\mathcal{R}_\ell$ denote the resulting reserved set. The allocated layer budget satisfies $B_\ell \geq |\mathcal{R}_\ell|$.

For any candidate set $\mathcal{C}$, $\mathrm{TopK}_B(\rho; \mathcal{C})$ denotes the B blocks in $\mathcal{C}$ with largest priorities ρ_b. The remaining layer budget is allocated globally across heads:

$$\widehat{\mathcal{S}}_\ell = \mathcal{R}_\ell \cup \mathrm{TopK}_{B_\ell - |\mathcal{R}_\ell|}\left(\rho; \mathcal{C}_\ell \setminus \mathcal{R}_\ell\right).$$

Proof of Theorem 3.5. Let $\mathcal{C}$ denote the candidate set of a Top-K stage and assume $1 \leq B < |\mathcal{C}|$. For any $b \in \mathcal{S}_B^\star$ and $c \in \mathcal{C} \setminus \mathcal{S}_B^\star$,

$$Y_b - Y_c \geq Y_{(B)} - Y_{(B+1)} > 2\varepsilon.$$

By assumption, $|\rho_b - Y_b| \leq \varepsilon$ and $|\rho_c - Y_c| \leq \varepsilon$, hence

$$\rho_b - \rho_c \geq Y_b - Y_c - 2\varepsilon > 0.$$

Every target-selected block therefore remains above every unselected candidate, proving $\widehat{\mathcal{S}}_B = \mathcal{S}_B^\star$.

The row-wise reservation is the same argument with $B = 1$ applied independently to each row. Once the reserved sets agree, both routers have the same residual candidates and budget, so the argument applies to the residual global Top-K stage. Stages that select no candidates or all available candidates agree automatically. □

A.6 Blockwise Sparse-Attention Error

Fix a query region R_i whose tokens share the selected key blocks, and let $\mathcal{C}_i$ denote its selectable key-block candidates. The physical key-token supports $\{\mathcal{B}_b : b \in \mathcal{C}_i\}$ are pairwise disjoint. Let $\mathcal{S} \subseteq \mathcal{C}_i$ denote the retained selectable blocks. All valid interactions outside this selectable family, including model-specific always-retained interactions, remain in the sparse support.

For $b \in \mathcal{C}_i$, recall the dense-output contribution

$$I_b = \frac{1}{|R_i|} \sum_{t \in R_i} \left\| \sum_{s \in \mathcal{B}_b} A_{ts} v_s \right\|_2$$

and define its average dense attention mass as

$$M_b = \frac{1}{|R_i|} \sum_{t \in R_i} \sum_{s \in \mathcal{B}_b} A_{ts}.$$

For $t \in R_i$, define the omitted dense contribution and omitted probability mass by

$$D_t = \sum_{b \in \mathcal{C}_i \setminus \mathcal{S}} \sum_{s \in \mathcal{B}_b} A_{ts} v_s, \qquad \delta_t = \sum_{b \in \mathcal{C}_i \setminus \mathcal{S}} \sum_{s \in \mathcal{B}_b} A_{ts}.$$

Let $R_t = O_t^{\mathrm{d}} - D_t$ denote the contribution of the retained dense support. The routing constraints guarantee a nonempty retained support, so $\delta_t < 1$. Since the retained keys carry dense attention mass $1 - \delta_t$, sparse attention renormalizes them as

$$O_t^{\mathrm{s}} = \frac{R_t}{1 - \delta_t}.$$

Proof of Lemma 3.6. From the preceding decomposition,

$$O_t^{\mathrm{d}} - O_t^{\mathrm{s}} = D_t - \frac{\delta_t}{1 - \delta_t} R_t.$$

Since $\|v_s\|_2 \le V_{\max}$ and the retained dense probability mass equals $1 - \delta_t$,

$$\|R_t\|_2 \le (1 - \delta_t) V_{\max}.$$

Therefore,

$$\|O_t^{\mathrm{d}} - O_t^{\mathrm{s}}\|_2 \le \|D_t\|_2 + \delta_t V_{\max}.$$

By the triangle inequality,

$$\|D_t\|_2 \le \sum_{b \in \mathcal{C}_i \setminus \mathcal{S}} \left\| \sum_{s \in \mathcal{B}_b} A_{ts} v_s \right\|_2.$$

Averaging over $t \in R_i$ yields

$$\frac{1}{|R_i|} \sum_{t \in R_i} \|D_t\|_2 \le \sum_{b \in \mathcal{C}_i \setminus \mathcal{S}} I_b,$$

while

$$\frac{1}{|R_i|} \sum_{t \in R_i} \delta_t = \sum_{b \in \mathcal{C}_i \setminus \mathcal{S}} M_b.$$

Combining these inequalities gives

$$\frac{1}{|R_i|} \sum_{t \in R_i} \left\| O_t^{\mathrm{d}} - O_t^{\mathrm{s}} \right\|_2 \le \sum_{b \in \mathcal{C}_i \setminus \mathcal{S}} \left(I_b + V_{\max} M_b \right),$$

which proves Equation (6). □

For every selectable block $b \in \mathcal{C}_i$,

$$I_b \le \frac{1}{|R_i|} \sum_{t \in R_i} \sum_{s \in \mathcal{B}_b} A_{ts} \|v_s\|_2 \le V_{\max} M_b.$$

Theorem A.2 (Calibrated Sparse-Routing Error)**.** *Consider a layer equipped with the calibrated router and assume $\|v_s\|_2 \le V_{\max}$ for every valid key token. Suppose the conditions of Theorem 3.5 hold at every nontrivial constrained ranking stage. Then the predicted and target routers return the same support $\mathcal{S}^\star$. With a common deterministic tie-breaking convention, the target router based on Y_b makes the same decisions as the router based on I_b. For each query region R_i, let $\mathcal{S}_i^\star = \mathcal{S}^\star \cap \mathcal{C}_i$. If every query retains a nonempty support, the output satisfies*

$$\frac{1}{|R_i|}\sum_{t\in R_i}\left\|O_t^{\mathrm{d}} - O_t^{\mathrm{s}}\right\|_2 \le \sum_{b\in\mathcal{C}_i\setminus\mathcal{S}_i^\star}(I_b + V_{\max}M_b) \le 2V_{\max}\sum_{b\in\mathcal{C}_i\setminus\mathcal{S}_i^\star} M_b. \tag{13}$$

Proof of Theorem A.2. Under the conditions of Theorem 3.5, the predicted and target routers make identical decisions at every constrained ranking stage. With the same deterministic tie-breaking, PRank preserves the ordering induced by I_b, so replacing Y_b by I_b leaves every target ranking decision unchanged.

Let $\mathcal{S}^\star$ denote the complete selectable support returned by the target router and $\mathcal{S}_i^\star = \mathcal{S}^\star \cap \mathcal{C}_i$ its retained selectable blocks for query region R_i. Applying Lemma 3.6 with $\mathcal{S} = \mathcal{S}_i^\star$ gives

$$\frac{1}{|R_i|}\sum_{t\in R_i}\left\|O_t^{\mathrm{d}} - O_t^{\mathrm{s}}\right\|_2 \le \sum_{b\in\mathcal{C}_i\setminus\mathcal{S}_i^\star}(I_b + V_{\max}M_b)\,.$$

Finally, $I_b \le V_{\max}M_b$ for every omitted block, so

$$\frac{1}{|R_i|}\sum_{t\in R_i}\left\|O_t^{\mathrm{d}} - O_t^{\mathrm{s}}\right\|_2 \le 2V_{\max}\sum_{b\in\mathcal{C}_i\setminus\mathcal{S}_i^\star} M_b,$$

which proves Equation (13). □

B Detailed Experimental Setup

Evaluation protocols. For camera pose estimation on CO3Dv2 (Reizenstein et al., 2021), we evaluate with 10, 30, and 100 input frames to assess performance under varying sequence lengths, following (Wang et al., 2025b). At an angular threshold τ, a camera pair is considered correct when both its relative rotation and translation-direction errors are below τ. AUC@30 integrates this joint pose accuracy over $\tau \in [0^\circ, 30^\circ]$.

For dense geometry prediction, we follow the evaluation protocol of each backbone (Wang et al., 2025b; 2026b; Lin et al., 2025): on 7 Scenes (Shotton et al., 2013), NRGBD (Azinović et al., 2022), and ETH3D (Schops et al., 2017), we report the depth accuracy $\delta_{1.25}$ and point-cloud accuracy (Acc.) and completeness (Comp.) following (Azinović et al., 2022). For 7Scenes (Shotton et al., 2013), we use the official test split and retain every 20th frame. For NRGBD (Azinović et al., 2022), frames with missing RGB-D observations or invalid camera poses are discarded, after which every 40th frame is retained. For ETH3D (Schops et al., 2017), we follow the 11-scene subset and view-filtering protocol used by DA3 (Lin et al., 2025), retaining 14–76 valid views per scene.

All methods except SAF3R (Deng et al., 2026) are timed directly under the standard CO3Dv2 evaluation protocol, which incorporates native resolution variations. In our evaluation, the official SAF3R implementation incurs substantial additional overhead when processing sequences with varying resolutions, stemming from dynamic indexing, per-head execution, and memory allocation; no comparable overhead is observed for the remaining methods. We therefore report SAF3R speedups for 10, 30, and 100 frames at a fixed resolution of 392×518 after warm-up, following its fixed-shape benchmarking setup.

Implementation details. Visual queries and keys are partitioned into blocks of 128 and 64 tokens, respectively. Routing proxies are computed in FP16 from the normalized, position-encoded queries and keys; lower-precision computation is applied only to the selected token-level attention blocks. The contribution calibrator and layer-sensitivity profile are fitted on a hold-out calibration set disjoint from the evaluation data and remain fixed at test time, so inference incurs no dense attention or online profiling. End-to-end latency is measured on a single RTX 5090 with batch size 1.

Table 7: Mixed-precision camera-pose estimation on CO3Dv2 (Reizenstein et al., 2021).

Model	Sparsity	10 Frames		30 Frames		100 Frames	
		AUC@30↑	Latency (s)	AUC@30↑	Latency (s)	AUC@30↑	Latency (s)
VGGT	0%	90.03	0.529 (1.000×)	90.47	2.290 (1.000×)	90.67	12.21 (1.000×)
	50%	88.45	0.491 (1.077×)	89.03	1.541 (1.486×)	89.42	5.455 (2.238×)
	75%	85.77	0.488 (1.084×)	87.16	1.496 (1.531×)	87.73	5.296 (2.306×)
π^3	0%	90.38	0.491 (1.000×)	90.72	2.203 (1.000×)	91.10	12.10 (1.000×)
	50%	88.96	0.466 (1.054×)	89.79	1.414 (1.558×)	90.41	4.957 (2.441×)
	75%	80.46	0.456 (1.077×)	82.89	1.377 (1.600×)	84.78	4.161 (2.908×)
DA3	0%	87.57	0.383 (1.000×)	88.25	1.287 (1.000×)	88.56	5.439 (1.000×)
	50%	86.38	0.358 (1.070×)	87.15	1.055 (1.220×)	88.20	3.569 (1.524×)
	75%	82.27	0.351 (1.091×)	84.49	1.025 (1.256×)	86.54	3.024 (1.799×)

Table 8: Mixed-precision depth estimation and dense Reconstruction on 7 Scenes (Shotton et al., 2013), NRGBD (Azinović et al., 2022) and ETH3D (Schops et al., 2017)

Model	Sparsity	7Scenes			NRGBD			ETH3D		
		δ. ↑	Acc.↓	Comp.↓	δ. ↑	Acc.↓	Comp.↓	δ. ↑	Acc.↓	Comp.↓
VGGT	50%	95.98	0.0188	0.0257	99.27	0.0158	0.0133	98.47	0.1468	0.1213
	75%	96.64	0.0191	0.0251	99.13	0.0266	0.0168	97.68	0.2154	0.1389
π^3	50%	98.75	0.0138	0.0139	99.22	0.0247	0.0129	99.76	0.0965	0.0507
	75%	98.67	0.0268	0.0167	99.19	0.0580	0.0244	99.65	0.2509	0.1292
DA3	50%	97.75	0.0199	0.0212	99.20	0.0394	0.0803	99.21	0.1630	0.0855
	75%	97.77	0.0239	0.0207	99.15	0.0431	0.0727	98.99	0.1914	0.0921

Sparse execution is implemented with our own low-precision sparse-attention kernel, tailored to our block layout and priority-based bit assignment; all routing-related computation remains in FP16.

Calibration dataset. The contribution calibrator and layer-sensitivity profile are fitted on a subset from the CO3Dv2 (Reizenstein et al., 2021) training set with at least 16 frames. Evaluation on the geometry benchmarks is thus strictly cross-dataset. All fitted parameters remain frozen at test time.

C Mixed-Precision Results

Tables 7 and 8 report full-system results under the even mixed 4/8-bit execution used in Figure 5, where the top half of the retained blocks run in 8-bit and the rest in 4-bit. Across all three models and both tasks, the mixed-precision results closely follow their 8-bit counterparts in Tables 1 and 2, indicating that the calibrated priority assignment concentrates 4-bit execution on low-contribution blocks and leaves accuracy largely intact. The 0% rows involve no sparse execution and are identical to the dense FP16 reference.

D Further Analysis of the Block Contribution Calibrator

We further analyze the block-level features, contextual conditioning, and head-dependent terms used by the block contribution calibrator. The complementarity of Avg and Max pooling is evaluated in the main ablation in Table 3; here, we focus on the design of the calibrator.

Experimental protocol. We extract frozen activations from multi-view geometry Transformers and evaluate intermediate layers, where contribution calibration is applied in the final deployment profile. The calibrator is fitted on a calibration dataset. All variants use identical physical block budgets and row-connectivity constraints. Ridge regularization is fixed to $\lambda = 10^{-3}$. For each ablation, the calibrator is refitted from scratch after removing the corresponding feature and all interaction terms involving that feature.

We report the relative ℓ_2 distortion between sparse and dense attention outputs, the Spearman correlation between predicted priority ρ_b and dense-output utility I_b, and the IoU between the predicted mask and the utility-oracle mask under the same constrained routing procedure. Metrics are computed for each layer and input instance and then macro-averaged over layers and held-out sequences.

Table 9: **Leave-one-feature-out ablation of block-level features.** Each calibrator is refitted after removing the indicated feature and its associated interaction terms. $\Delta\ell_2$ is measured relative to the all-context reference.

Removed feature	Rel. $\ell_2 \downarrow$	$\Delta\ell_2$	Spearman $\uparrow$	Oracle IoU $\uparrow$
None (all-context reference)	0.08564	–	0.8915	0.7744
Avg row rank	0.08612	+0.56%	0.8911	0.7739
Max row rank	0.08602	+0.44%	0.8909	0.7730
Avg within-head rank	0.08593	+0.34%	0.8538	0.7397
Max within-head rank	0.08614	+0.58%	0.8855	0.7683
Rank agreement	0.08576	+0.14%	0.8912	0.7741
Rank product	0.08675	+1.30%	0.8898	0.7736
Valid-area ratio	0.08624	+0.70%	0.8902	0.7728

Table 10: **Group ablation of contextual conditioning, feature–context interactions, and head identity.** Removing a context group also removes its associated interaction terms.

Configuration	Rel. $\ell_2 \downarrow$	$\Delta\ell_2$	Spearman $\uparrow$	Oracle IoU $\uparrow$
All-context reference	**0.08564**	–	**0.8915**	**0.7744**
No dynamic context	0.08894	+3.86%	0.8690	0.7522
No context	0.08904	+3.97%	0.8689	0.7519
No head identity	0.09203	+7.46%	0.8787	0.7609
No feature–context interactions	0.08641	+0.90%	0.8780	0.7642

The leave-one-feature-out and group ablations use the larger all-context model as a common diagnostic reference. We separately ablate the interaction scope to determine the final calibrator used in the main experiments.

Block-level features. As shown in Table 9, the rank product has the largest individual effect on output distortion. This supports explicitly representing candidates ranked highly by both pooled branches. The valid-area ratio is also beneficial because ragged physical blocks may contain different numbers of valid tokens.

The row-wise and within-head ranks provide complementary normalization scopes. Row-wise ranks compare candidate keys for an individual query region, whereas within-head ranks make priorities more comparable across query rows. Notably, removing the Avg within-head rank substantially reduces both rank correlation and oracle-mask overlap.

Context and head calibration. The group ablations in Table 10 show that the context statistics are most useful jointly. Removing dynamic context increases output distortion by 3.86%, while removing all context increases it by 3.97%. Removing feature–context interactions also reduces rank correlation and mask overlap, supporting conditional modulation of the block-level evidence.

Head identity has the largest group-level effect, increasing distortion by 7.46% when removed. Although relative ranks reduce raw score-scale differences, the dense-output utility distributions remain head-dependent. The head-specific terms therefore improve the comparability of priorities before candidates compete under the shared cross-head budget.

Interaction scope. As shown in Table 11, expanding the feature–context interaction from dynamic statistics to all context variables yields only a marginal improvement, while increasing the feature dimension from 69 to 111. We therefore interact block features only with the dynamic proxy statistics, which directly describe the input-dependent distributions of the Avg and Max branches.

Removing static context entirely causes a further degradation. We thus retain the static descriptors as main effects but do not interact them with the block-level features. The final calibrator is consequently defined as

$$\psi_b = \left[f_b;\, c^{\mathrm{static}};\, c_h^{\mathrm{dyn}};\, f_b \otimes c_h^{\mathrm{dyn}};\, \mathrm{onehot}(\mathrm{head}(b)) \right]^\top \in \mathbb{R}^{69}. \tag{14}$$

Table 11: **Scope of feature–context interactions.** The all-context model is used as a diagnostic reference, whereas the dynamic-only interaction model is used in the final method.

Interaction design	Dimensions	Rel. $\ell_2 \downarrow$	Oracle IoU $\uparrow$
All-context interactions (reference)	111	**0.085642**	**0.77438**
Dynamic-only interactions (final)	69	0.085715	0.77336
No static context	63	0.085830	0.77316

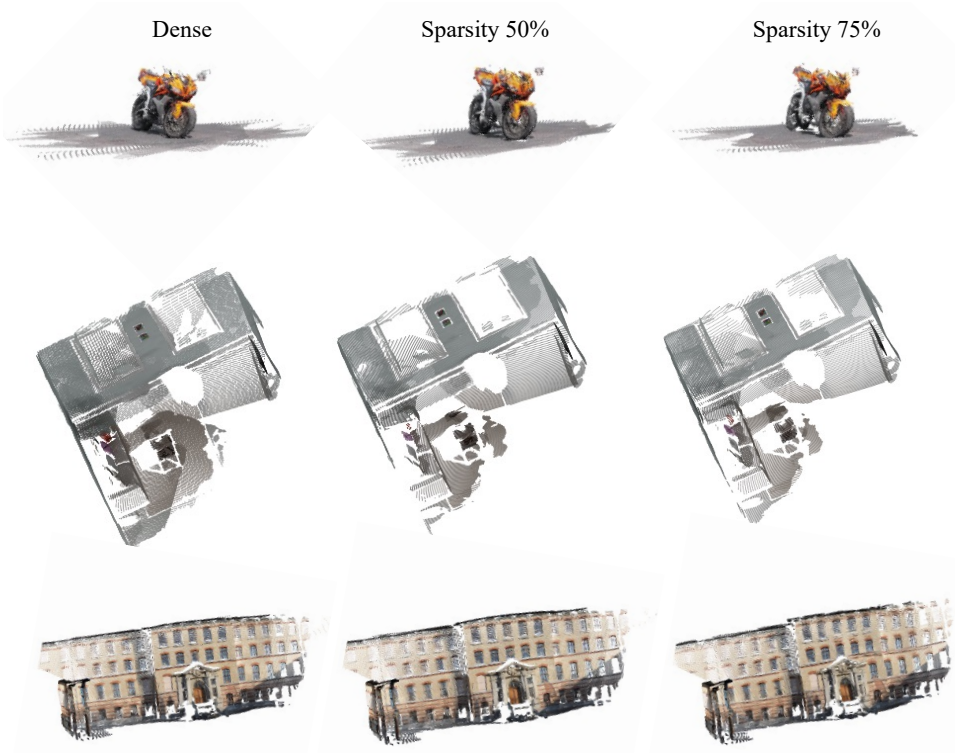
The 69 dimensions comprise seven block features, six static descriptors, five dynamic statistics, 35 dynamic feature–context interactions, and 16 head-identity indicators. Overall, the 69-dimensional calibrator preserves nearly all of the routing fidelity of the larger diagnostic model while avoiding static interaction terms with limited empirical benefit. These ablations measure routing fidelity under a matched sparse budget rather than feature-wise changes in downstream task accuracy; camera-pose and dense-geometry results are reported separately in the main experiments.

E MORE VISUALIZATION RESULTS

Figure 6 presents additional qualitative results on object-centric, indoor, and outdoor scenes. Panel (a) compares confidence-filtered reconstructions at 75% sparsity: SparseVGGT and FastVGGT lose object parts or entire surfaces, whereas MosaicAttention retains substantially more high-confidence geometry and stays closest to the dense reference. Panel (b) shows MosaicAttention at increasing sparsity levels, where both scene layout and fine structures remain stable from 50% to 75%. These observations are consistent with the quantitative results in Table 2.



(a) Confidence-filtered point-cloud reconstruction comparison.



(b) Qualitative results across sparsity levels.

Figure 6: **Additional visualization results.** (a) Confidence-filtered reconstructions at 75% sparsity on object-centric, indoor, and outdoor scenes. (b) MosaicAttention reconstructions as sparsity increases from 50% to 75%. The confidence-filtering protocol follows Figure 4.