QUADVGGT: QUOTIENT ATTENTION FOR ACCELERATING DENSE-TOKEN INFERENCE IN VISUAL GEOMETRY GROUNDED TRANSFORMERS

Anonymous authors
Paper under double-blind review

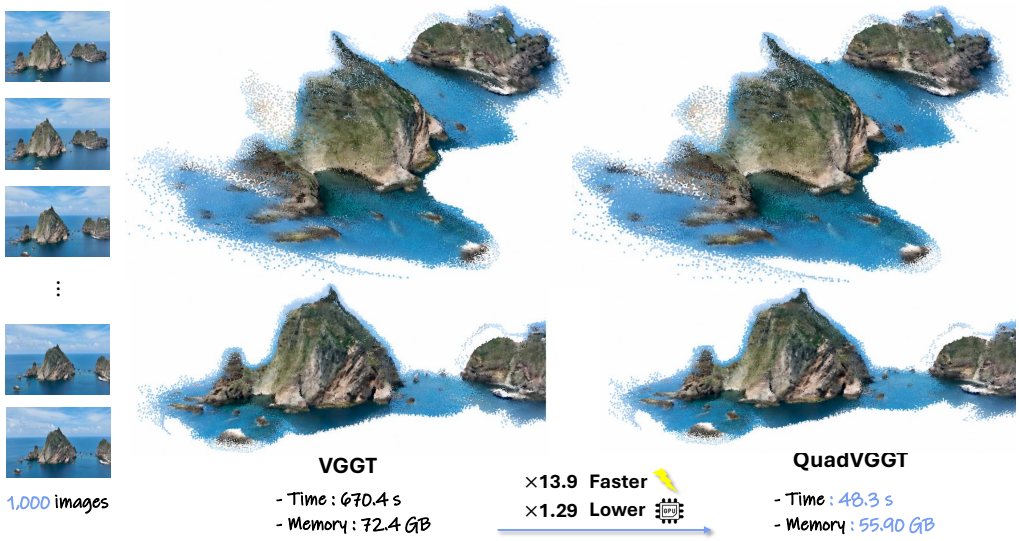
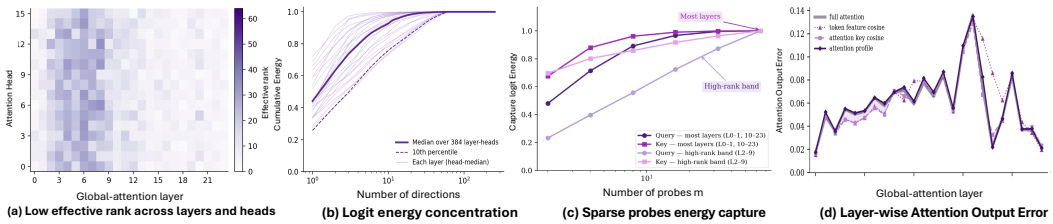


Figure 1: **Qualitative comparison on a single 1,000-image island scene.** QuadVGGT preserves geometry comparable to VGGT while delivering faster inference and lower peak GPU memory.

ABSTRACT

Feed-forward geometric foundation models such as VGGT have significantly advanced multi-view 3D reconstruction by directly predicting camera parameters and dense scene geometry from uncalibrated images. However, their global attention jointly processes all image tokens, causing computational and memory costs to grow rapidly with the number of input views and limiting long-sequence inference. Existing acceleration methods primarily reduce, merge, or compress tokens, potentially altering token-specific interactions within attention. We introduce QuadVGGT, a training-free framework based on Quotient Attention. Our key observation is that representational redundancy and computational redundancy are distinct: tokens that retain distinct representations for dense geometric prediction can nonetheless induce similar attention interactions, allowing their computations to be shared without merging their representations. QuadVGGT uses sparse operator-induced profiles to select original token representatives for independent key–value and query computation sharing while preserving the original dense token stream. To further support long-sequence inference, we combine Quotient Attention with scene-adaptive anchor construction and hierarchical feature offloading. QuadVGGT can be directly applied to pretrained VGGT without additional training or modification of its prediction heads. Extensive experiments across dense 3D reconstruction, depth estimation, and camera pose estimation demonstrate substantial efficiency gains while maintaining comparable geometric accuracy. On 1,000-view reconstruction, QuadVGGT achieves a $14.8\times$ speedup and reduces peak GPU memory from 72.4 to 55.9 GB.

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.



(a) Low effective rank across layers and heads (b) Logit energy concentration (c) Sparse probes energy capture (d) Layer-wise Attention Output Error

Figure 2: **Motivation and validation of sparse operator-induced representative selection.** (a) Effective rank across VGGT's global-attention layers and heads, revealing low-dimensional interactions. (b) Cumulative logit energy across 384 layer–head pairs, showing concentration in a few dominant directions. (c) Logit-energy capture versus probe count, showing that few probes capture most logit energy, except in higher-rank layers. (d) Layer-wise attention-output error comparing full-profile selection (using all query/key tokens as probes), sparse profiles with 16 probes, feature cosine, and key cosine under identical representative budgets and candidate pools. Sparse profiles closely match full-profile selection across layers.

1 Introduction

Feed-forward geometric foundation models (Lin et al., 2025; Wang et al., 2026d;b) such as VGGT (Wang et al., 2025a) have advanced multi-view 3D reconstruction by directly predicting cameras and dense scene geometry from uncalibrated images. However, their reliance on global attention incurs computational costs that grow quadratically with the input sequence length, limiting long-sequence inference. Recent methods address this bottleneck through token merging, sparsification, and compact representations. FastVGGT (Shen et al., 2026) exploits attention redundancy through training-free token reduction, while HTTM (Wang et al., 2026c) performs head-wise temporal merging. Although attention-internal merging can preserve the dense residual stream, it replaces original tokens with merged representations during attention computation. This distinction matters because tokens with different representations may still induce similar attention interactions. For example, patches in neighboring frames may observe overlapping scene regions and exhibit similar attention behavior, while retaining distinct features for frame-specific dense predictions. Rather than merging these representations, we can potentially share their redundant attention computations. Query clustering partially exploits this idea by computing attention for representative queries and reusing their outputs. However, it still attends to the original key–value tokens, leaving redundancy on the key–value side unaddressed. This motivates a different formulation: *Can both sides of the attention operator be independently approximated using original token representatives, selected according to their effects on attention rather than representation similarity?*

The key challenge is to identify attention computations that can be shared with limited output distortion. Token similarity provides a useful signal, but does not fully characterize its effect under the attention operator. In particular, query and key differences affect attention only through the directions exposed by their interactions, and query distributions are additionally invariant to a common shift of the logit row. Directly evaluating these relations from full attention responses, however, would incur the computation we seek to avoid. We observe that VGGT's query–key interactions exhibit low effective dimensionality, with most logit energy concentrated in a few dominant directions (Fig. 2(a,b)). Consequently, sparse probes capture much of the interaction structure (Fig. 2(c)). We further examine whether the resulting sparse operator criteria track the actual attention distortions induced by computation sharing (Fig. 7).

Building on these observations, we introduce **QuadVGGT**, a training-free framework that reduces redundant global-attention computation while preserving distinct dense residual states. Its core component, **Quotient Attention**, independently approximates key–value contributions and query outputs using original token representatives. On the key–value side, sparse probe queries compare the logits induced by candidate keys, while multiplicity correction accounts for group attention mass. On the query side, sparse probe keys measure centered logit discrepancies, and representatives are selected independently for each attention head. Quotient Attention therefore shares repeated attention computations without averaging the dense token representations. We analyze the resulting approximation by separating representative value substitution, key–value mass distortion, and query-output sharing. The analysis characterizes conditions for exact key–value aggregation and

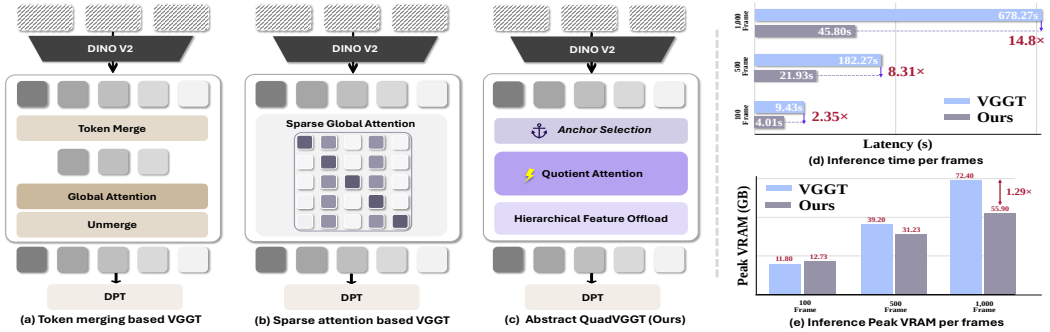


Figure 3: **VGGT acceleration strategies and efficiency.** (a) Token merging, (b) sparse global attention, and (c) ours. (d–e) Inference time and peak GPU memory across frame counts.

exact query sharing, and relates the two distributional distortion terms to logit perturbations under the attention operator. This motivates the sparse probe distances used for representative selection.

To further support long-sequence inference, QuadVGGT incorporates scene-adaptive **Delta-Walking** to limit costly representative searches by selecting anchor frames. To address the memory overhead of intermediate features retained for dense prediction, we additionally introduce **Hierarchical Feature Offloading**, which transfers inactive features to CPU memory (Rhu et al., 2016; Peng et al., 2020). The resulting framework supports long-sequence inference without additional training or modifications to the pretrained prediction heads.

Our contributions are threefold:

- We formulate *dual-sided attention computation sharing*, independently approximating key–value contributions and query outputs using original token representatives. We characterize its error through value substitution, key–value mass distortion, and head-wise query sharing.
- We introduce *operator-induced representative selection* for both sides of the operator. Sparse incoming-logit profiles guide key–value assignments, while centered-logit profiles guide head-wise query assignments. These criteria target attention-relevant discrepancies without constructing merged token representations.
- We develop a training-free long-sequence inference framework that integrates Quotient Attention with scene-adaptive Delta-Walking and Hierarchical Feature Offloading, and evaluate its geometric fidelity and efficiency.

2 Related Work

Feed-forward Multi-view 3D Reconstruction. Feed-forward methods directly predict geometry from images. DUSt3R (Wang et al., 2024) introduces dense point-map regression, while MASt3R (Leroy et al., 2024) incorporates local features for matching. Fast3R (Yang et al., 2025) extends reconstruction to multiple views, and VGGT (Wang et al., 2025a) jointly predicts cameras, depth, point maps, and tracks using frame-wise and global attention. Building on pretrained VGGT, we reduce global-attention computation while preserving dense token states.

Efficient Multi-view Geometry Transformers. Recent methods improve VGGT scalability by exploiting different forms of redundancy (Shu et al., 2026). FastVGGT (Shen et al., 2026) reduces attention computation through training-free token reduction, while HTTM (Wang et al., 2026c) performs head-wise temporal merging to preserve feature uniqueness. FlashVGGT (Wang & Xu, 2026) compresses spatial information into compact descriptors for efficient global interactions. Block-sparse attention (Wang et al., 2026a) instead reduces computation by selectively evaluating query–key blocks. In contrast, **Quotient Attention separates computation sharing from representation merging**. It selects original representatives using sparse operator-induced profiles and independently approximates key–value contributions and query outputs while preserving distinct dense residual states. Representative selection is motivated by attention-sharing distortion rather than feature similarity alone.

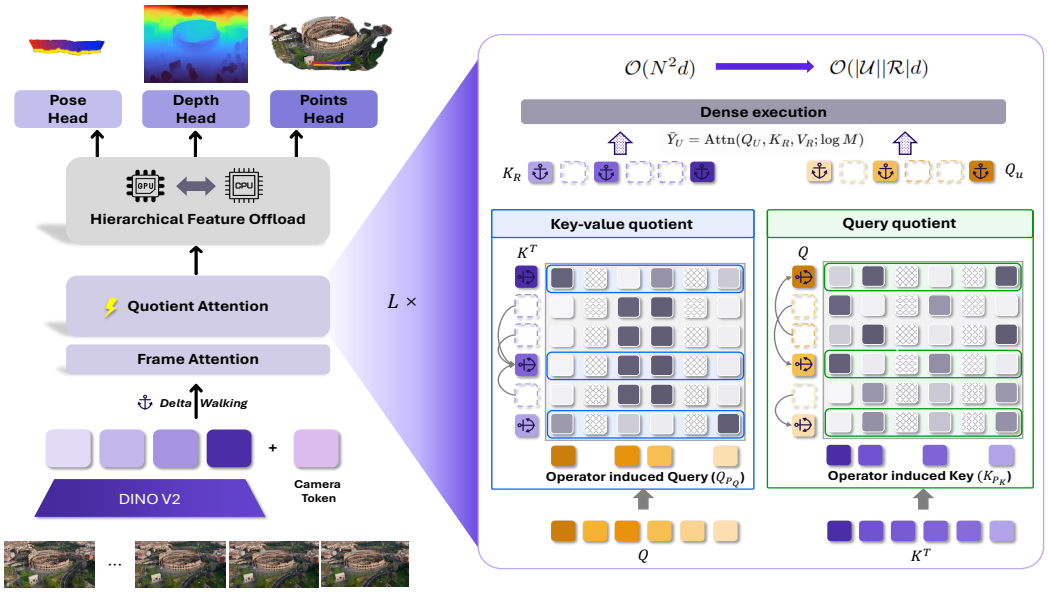


Figure 4: **Overview of QuadVGGT.** Delta-Walking selects anchor frames for representative search. Quotient Attention shares redundant key–value and query computations through operator-induced representatives while preserving dense token states. The key–value quotient applies multiplicity correction, whereas the query quotient shares representative attention outputs among assigned queries. Hierarchical Feature Offloading reduces peak GPU memory by transferring intermediate features to CPU memory and restoring them in frame slabs.

Efficient Attention and Token Reduction. Transformer efficiency has been widely studied through token reduction and attention approximation (Vyas et al., 2020). ToMe (Bolya et al., 2023) merges similar token representations and uses proportional attention to account for token multiplicity. Performer (Choromanski et al., 2022) approximates softmax attention through random-feature kernelization, whereas Nyströmformer (Xiong et al., 2021) employs landmark-based attention approximation. In contrast, Quotient Attention uses sparse operator-induced responses to select original token representatives and independently shares key–value contributions and query outputs. This avoids feature averaging while preserving token-specific residual states, without requiring reconstruction of the full attention matrix.

3 Method

QuadVGGT accelerates long-sequence VGGT inference without reducing the dense token stream or modifying the pretrained prediction heads. As illustrated in Fig. 4, it combines **Delta-Walking** for scene-adaptive anchor construction, **Quotient Attention** for sharing redundant global-attention computations, and **Hierarchical Feature Offloading** for reducing peak GPU memory.

3.1 Delta-Walking Anchor Construction

We construct scene-adaptive anchors from normalized DINOv2 (Oquab et al., 2023) patch tokens $t_{s,p}$, excluding special tokens. For frame s, the descriptor is

$$g_s = \mathrm{norm}\left(\frac{1}{P}\sum_{p=1}^{P}\mathrm{norm}(t_{s,p})\right). \tag{1}$$

We initialize the anchor set with the first two frames. For each subsequent frame s, a new anchor is added when

$$1 - \langle g_s, g_{a_{\mathrm{last}}}\rangle > \Delta^\star, \tag{2}$$

where a_{last} denotes the latest anchor. The selected anchors restrict representative candidates while all non-anchor frames and their dense tokens remain intact. Delta-Walking assumes a meaningful temporal ordering of the input views, which is natural for the long-sequence video and sequential-capture settings targeted in this work; we discuss this design choice further in Appendix A.2.

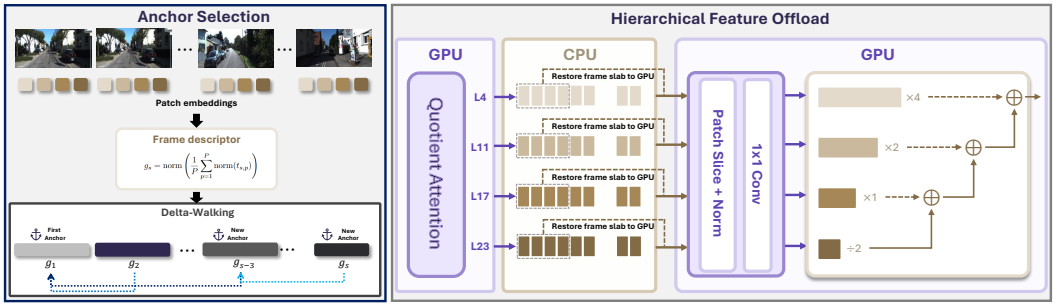


Figure 5: **Delta-Walking and Hierarchical Feature Offloading.** (a) Delta-Walking selects scene-adaptive anchors while retaining all frames. (b) Intermediate features are offloaded to CPU and restored in frame slabs, reducing peak GPU memory while preserving dense predictions.

3.2 Quotient Attention

For head h, let $Q_h, K_h, V_h \in \mathbb{R}^{N\times d}$, $Z_h = Q_h K_h^\top/\sqrt{d}$, and $Y_h = \mathrm{softmax}_{\mathrm{row}}(Z_h)V_h$. Quotient Attention independently approximates key–value contributions and query outputs using original token representatives. Let $\mathcal{T}$, $\mathcal{A}$, and $\mathcal{S}$ denote the patch-token, anchor-patch, and camera/register-token indices, respectively.

Operator-induced profiles. At each layer, we uniformly sample m probe queries P_Q and m probe keys P_K. For KV selection, we use incoming logit profiles

$$\zeta_{j,h} = \frac{Q_{P_Q,h}k_{j,h}}{\sqrt{md}}, \qquad d_h^K(j,a) = \|\zeta_{j,h} - \zeta_{a,h}\|_2^2. \tag{3}$$

For query selection, we remove the row-wise shift to which softmax is invariant:

$$\psi_{i,h} = \frac{C_m K_{P_K,h} q_{i,h}}{\sqrt{md}}, \qquad C_m = I_m - \frac{1}{m}\mathbf{1}\mathbf{1}^\top, \qquad d_h(i,a) = \|\psi_{i,h} - \psi_{a,h}\|_2^2. \tag{4}$$

For each non-anchor patch, $\mathcal{C}(\cdot)$ contains patches from its two temporally nearest anchor frames. We select

$$\pi_K(j) = \arg\min_{a\in\mathcal{C}(j)} \sum_h d_h^K(j,a), \qquad b(i) = \arg\min_{a\in\mathcal{C}(i)} \sum_h d_h(i,a). \tag{5}$$

Key–value quotient. The retained KV indices are $\mathcal{R} = \mathcal{A}\cup\mathcal{S}$, with $\pi_K(a) = a$ for $a \in \mathcal{R}$. Let $\mathcal{S}_a = \{j : \pi_K(j) = a\}$ and $M_a = |\mathcal{S}_a|$. Using the original representative $(k_{a,h}, v_{a,h})$, we compute

$$\bar{p}_{i,h} = \mathrm{softmax}\left(z_{i\mathcal{R},h} + \log M\right), \qquad \bar{y}_{i,h} = \sum_{a\in\mathcal{R}} \bar{p}_{ia,h} v_{a,h}. \tag{6}$$

The $\log M_a$ correction is exact when group members induce the same logits and values as their representative; otherwise it introduces mass and value distortion.

Query quotient and dense execution. Let $\mathcal{U}$ contain all anchor patches and camera/register tokens. For $i \notin \mathcal{U}$, let $b'(i)$ denote the patch at the same image-grid position as i in the anchor frame containing $b(i)$. Each head independently selects

$$\pi_{Q,h}(i) = \arg\min_{a\in\{b(i),\, b'(i)\}} d_h(i,a), \qquad i \notin \mathcal{U}. \tag{7}$$

Queries in $\mathcal{U}$ map to themselves. We evaluate reduced attention only for $\mathcal{U}$:

$$\bar{Y}_{\mathcal{U},h} = \mathrm{Attn}\left(Q_{\mathcal{U},h}, K_{\mathcal{R},h}, V_{\mathcal{R},h}; \log M\right), \qquad \hat{y}_{i,h} = \bar{y}_{\pi_{Q,h}(i),h}. \tag{8}$$

This reduces the attention cost to $\mathcal{O}(|\mathcal{U}||\mathcal{R}|d)$ per head, with an additional $\mathcal{O}(mNd)$ profiling cost plus representative matching and output gathering.

Output distortion. Let $\alpha_{i,h} = \mathrm{softmax}(z_{i\cdot,h})$ and

$$p_{ia,h} = \sum_{j \in \mathcal{S}_a} \alpha_{ij,h}, \qquad \mu_{ia,h} = p_{ia,h}^{-1} \sum_{j \in \mathcal{S}_a} \alpha_{ij,h} v_{j,h}.$$

Writing $D_{V,h} = \max_{a,c \in \mathcal{R}} \|v_{a,h} - v_{c,h}\|_2$ and $\mathrm{TV}(p,q) = \frac{1}{2}\|p - q\|_1$, the approximation error satisfies

$$\|y_{i,h} - \hat{y}_{i,h}\|_2 \le \underbrace{\sum_{a \in \mathcal{R}} p_{ia,h} \|\mu_{ia,h} - v_{a,h}\|_2}_{\text{value substitution}} + \underbrace{D_{V,h}\ \mathrm{TV}(p_{i,h}, \bar{p}_{i,h})}_{\text{KV mass distortion}} + \underbrace{D_{V,h}\ \mathrm{TV}\left(\bar{p}_{i,h}, \bar{p}_{\pi_{Q,h}(i),h}\right)}_{\text{head-wise query sharing}}. \tag{9}$$

The three terms correspond to value substitution, KV mass distortion, and query sharing. Appendix B derives the corresponding exactness conditions and probe-distance interpretations.

3.3 Hierarchical Feature Offloading

Although Quotient Attention reduces transformer computation, long-sequence inference remains memory-intensive because VGGT retains intermediate features from layers L_4, L_{11}, L_{17}, and L_{23} for its hierarchical prediction heads. Keeping these features on the GPU until they are consumed causes memory usage to grow rapidly with the number of input views.

We therefore decouple feature production from feature consumption. As shown in Fig. 5(b), once a required hierarchical feature is produced, we immediately transfer it from GPU memory to pinned CPU memory, releasing the corresponding GPU allocation. The stored feature is restored only when required by the dense prediction head, where it undergoes the original patch slicing, normalization, 1×1 projection, and multi-scale resizing operations.

For long sequences, restoring an entire hierarchical feature at once can still create a large temporary GPU-memory peak. We therefore restore and process the features in small frame slabs: only a subset of frames is transferred back to the GPU and forwarded through the prediction head at a time. This bounds the temporary memory required during feature reconstruction while preserving the original feature values and prediction-head computation.

4 Experiments

We evaluate QuadVGGT across three geometric prediction tasks: camera pose estimation, monocular depth estimation, and long-sequence dense 3D reconstruction. We assess geometric accuracy and inference efficiency across varying input sequence lengths and compare against representative feed-forward reconstruction models and efficient VGGT variants. We further conduct ablation studies to analyze the contributions of individual components.

4.1 Experimental Setup

Baselines and implementation. We compare QuadVGGT against the dense backbone VGGT (Wang et al., 2025a) and efficient reconstruction methods that reduce the cost of its global attention: FastVGGT (Shen et al., 2026), which merges tokens inside the attention operator; FlashVGGT (Wang & Xu, 2026), which attends to spatially compressed descriptors; SAF3R (Deng et al., 2026), which sparsifies attention per head; LiteVGGT (Shu et al., 2026), which employs geometry-aware cached token merging; and SparseVGGT (Wang et al., 2026a), which predicts a block-sparse attention mask. HTTM (Wang et al., 2026c) is discussed as a closely related approach, but is not included in the quantitative comparison because no public implementation was available at the time of our experiments. Every evaluated baseline is run from its authors' public code and released checkpoint inside our harness, so that all methods see the same frames, preprocessing, and metric code. SparseVGGT is evaluated at an effective block sparsity of 0.75, the operating point reported in its paper. QuadVGGT uses the pretrained VGGT weights without additional training and leaves the patch embedding, frame attention, and all prediction heads unchanged; only the 24 global attention layers are replaced. Three constants are fixed for every experiment: $m = 16$ probe rows, a walk radius $\Delta^\star = 0.002$ in the frontend descriptor space, and a slab size of 8 frames for hierarchical feature offloading, matching the chunk size of the dense prediction head. All efficiency measurements are taken on a single NVIDIA A100 80GB PCIe.

Table 1: Camera pose estimation on RealEstate10K and CO3Dv2. Best and second-best values are highlighted.

Method	RealEstate10K (Zhou et al., 2018)			CO3Dv2 (Reizenstein et al., 2021)		
	RRA@30↑	RTA@30↑	AUC@30↑	RRA@30↑	RTA@30↑	AUC@30↑
VGGT (Wang et al., 2025a)	99.97	95.68	86.73	98.84	97.11	87.98
FastVGGT (Shen et al., 2026)	99.92	95.66	84.57	97.27	96.52	86.88
LiteVGGT (Shu et al., 2026)	99.92	94.82	84.93	96.97	96.01	86.53
FlashVGGT (Wang & Xu, 2026)	99.95	95.61	85.13	98.49	96.75	87.42
SparseVGGT (Wang et al., 2026a)	99.83	92.62	84.47	96.38	93.71	85.32
SAF3R (Deng et al., 2026)	99.93	95.23	85.42	96.42	93.88	86.59
QuadVGGT (Ours)	99.96	95.65	86.59	98.79	96.91	87.42

Table 2: Monocular depth estimation on Sintel, Bonn, and NYU-v2. Best and second-best values are highlighted.

Method	Sintel (Butler et al., 2012)		Bonn (Palazzolo et al., 2019)		NYU-v2 (Silberman et al., 2012)	
	Abs Rel↓	$\tau < 1.25$ ↑	Abs Rel↓	$\tau < 1.25$ ↑	Abs Rel↓	$\tau < 1.25$ ↑
VGGT (Wang et al., 2025a)	0.246	0.721	0.053	0.969	0.057	0.951
FastVGGT (Shen et al., 2026)	0.295	0.694	0.051	0.971	0.055	0.949
LiteVGGT (Shu et al., 2026)	0.337	0.582	0.056	0.952	0.058	0.943
FlashVGGT (Wang & Xu, 2026)	0.273	0.669	0.053	0.966	0.058	0.956
SparseVGGT (Wang et al., 2026a)	0.257	0.701	0.051	0.974	0.057	0.954
SAF3R (Deng et al., 2026)	0.249	0.712	0.054	0.961	0.058	0.952
QuadVGGT (Ours)	0.247	0.720	0.051	0.968	0.057	0.951

Table 3: Large-scale dense 3D reconstruction. Results for sequences of 100, 500, and 1,000 images, averaged over N-RGBD (Azinović et al., 2022), 7-Scenes (Shotton et al., 2013), and ScanNet (Dai et al., 2017). Best and second-best values are highlighted. Rankings are computed separately for each frame count. OOM results are excluded.

Frames	Method	Depth		Point				Camera				Resource	
		Abs Rel↓	$\tau < 1.25$ ↑	Acc↓	Comp↓	CD↓	NC↑	APE↓	ARE↓	RPE-Trans↓	RPE-Rot↓	Time (s)↓	Mem. (GB)↓
100	VGGT (Wang et al., 2025a)	0.0226	0.9659	0.0078	0.0094	0.0084	0.7179	0.0130	0.0230	0.0026	0.0036	9.43	11.80
	FastVGGT (Shen et al., 2026)	0.0249	0.9651	0.0098	0.0103	0.0099	0.6815	0.0147	0.0282	0.0033	0.0057	6.59	12.83
	LiteVGGT (Shu et al., 2026)	0.0293	0.9627	0.0113	0.0099	0.0096	0.6648	0.0155	0.0278	0.0035	0.0047	4.40	12.70
	FlashVGGT (Wang & Xu, 2026)	0.0240	0.9713	0.0120	0.0127	0.0123	0.6673	0.0137	0.0250	0.0027	0.0037	4.20	12.67
	SparseVGGT (Wang et al., 2026a)	0.0257	0.9670	0.0100	0.0080	0.0090	0.6350	0.0163	0.0303	0.0040	0.0057	5.37	12.80
	SAF3R (Deng et al., 2026)	0.0227	0.9663	0.0080	0.0083	0.0083	0.6993	0.0133	0.0223	0.0027	0.0037	7.97	12.80
	QuadVGGT (Ours)	0.0240	0.9663	0.0077	0.0093	0.0087	0.7080	0.0140	0.0240	0.0033	0.0040	4.01	11.69
500	VGGT (Wang et al., 2025a)	0.0283	0.9677	0.0087	0.0083	0.0087	0.7100	0.0360	0.0540	0.0053	0.0087	182.27	39.20
	FastVGGT (Shen et al., 2026)	0.0295	0.9711	0.0102	0.0081	0.0084	0.6651	0.0368	0.0596	0.0049	0.0069	42.81	40.21
	LiteVGGT (Shu et al., 2026)	0.0347	0.9697	0.0110	0.0077	0.0093	0.6517	0.0407	0.0590	0.0070	0.0103	36.87	26.70
	FlashVGGT (Wang & Xu, 2026)	0.0283	0.9733	0.0113	0.0083	0.0100	0.6687	0.0340	0.0610	0.0047	0.0063	26.17	39.13
	SparseVGGT (Wang et al., 2026a)	0.0360	0.9670	0.0130	0.0073	0.0103	0.6340	0.0563	0.0963	0.0087	0.0183	61.93	42.70
	SAF3R (Deng et al., 2026)	0.0260	0.9680	0.0083	0.0073	0.0083	0.6963	0.0313	0.0500	0.0047	0.0067	62.63	41.40
	QuadVGGT (Ours)	0.0277	0.9673	0.0080	0.0087	0.0087	0.7167	0.0320	0.0520	0.0053	0.0060	21.93	31.23
1,000	VGGT (Wang et al., 2025a)	0.0313	0.9670	0.0119	0.0088	0.0104	0.6979	0.0548	0.0985	0.0083	0.0143	678.27	72.40
	FastVGGT (Shen et al., 2026)	0.0383	0.9712	0.0134	0.0079	0.0101	0.6924	0.0459	0.0621	0.0058	0.0139	94.83	72.60
	LiteVGGT (Shu et al., 2026)	0.0402	0.9690	0.0175	0.0089	0.0132	0.6504	0.0711	0.1313	0.0104	0.0184	108.77	51.10
	FlashVGGT (Wang & Xu, 2026)	0.0301	0.9745	0.0115	0.0079	0.0097	0.6701	0.0403	0.0639	0.0053	0.0071	72.23	72.40
	SparseVGGT (Wang et al., 2026a)	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM
	SAF3R (Deng et al., 2026)	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM
	QuadVGGT (Ours)	0.0274	0.9670	0.0087	0.0083	0.0085	0.7229	0.0362	0.0577	0.0058	0.0075	45.80	55.90

4.2 Monocular and Sparse-View Reconstruction

Camera pose estimation. Following the evaluation protocol of FlashVGGT, we evaluate camera pose estimation on RealEstate10K (Zhou et al., 2018) and CO3Dv2 (Reizenstein et al., 2021) using short input sequences. We report relative rotation accuracy (RRA), relative translation accuracy (RTA), and the area under the accuracy-threshold curve (AUC). This evaluation examines whether Quotient Attention preserves the cross-view geometric information required for accurate camera estimation. Table 1 presents the quantitative comparison with existing methods.

Monocular depth estimation. Following FlashVGGT, we evaluate per-frame depth prediction on Sintel (Butler et al., 2012), Bonn (Palazzolo et al., 2019), and NYU-v2 (Silberman et al., 2012). Rather than feeding a single image, each sequence is given to the model as a whole, so that the attention reduction is active, and the predicted depth of every frame is evaluated independently with per-frame scale alignment. For NYU-v2, which provides no temporal sequences, the test images are fed in groups. We report Abs Rel and threshold accuracy at $\tau < 1.25$. This experiment assesses whether reducing attention computation preserves the dense geometric information required for pixel-level depth prediction. Table 2 presents the comparison across all three benchmarks.

4.3 Long-Sequence Dense 3D Reconstruction

We evaluate long-sequence dense 3D reconstruction using 100, 500, and 1,000 views from N-RGBD (Azinović et al., 2022), 7-Scenes (Shotton et al., 2013), and ScanNet (Dai et al., 2017). Reconstruction quality is assessed using depth (Abs Rel, $\tau < 1.25$), point-cloud (accuracy, completeness, Chamfer distance, and normal consistency), and camera-pose metrics (APE, ARE, RPE-Trans, and RPE-Rot). We additionally report inference time and peak GPU memory to assess efficiency.

Quantitative and qualitative results. Table 3 summarizes the quantitative results across different input sequence lengths. At 1,000 views, QuadVGGT achieves a $14.8\times$ speedup over VGGT while

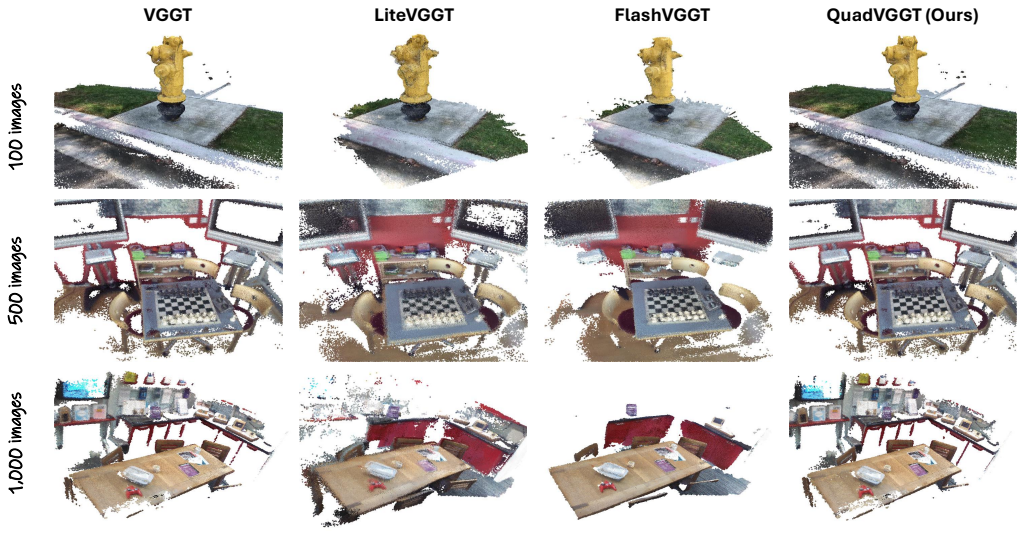


Figure 6: **Qualitative comparison of long-sequence 3D reconstruction.** We compare VGGT, LiteVGGT, FlashVGGT, and QuadVGGT using 100, 500, and 1,000 input images. QuadVGGT preserves fine-grained geometric details and structural completeness close to the original VGGT, while LiteVGGT and FlashVGGT exhibit more noticeable geometric degradation and missing structures.

reducing peak GPU memory from 72.4 to 55.9 GB. The memory reduction comes from Hierarchical Feature Offloading (Table 5), which is orthogonal to the attention approximation and could be combined with other methods. QuadVGGT maintains comparable geometric accuracy across datasets (Appendix A.6) despite sharing identical weights and heads. We do not attribute this to reduced attention dilution: the $\log M_a$ correction preserves aggregated attention mass, so fewer representatives do not by themselves concentrate attention. A more plausible hypothesis is that the quotient approximation acts as a mild regularizer at sequence lengths beyond the training regime; the effect is absent at 100 views, and we leave its direct verification to future work. Fig. 6 shows that QuadVGGT preserves fine-grained geometry and structural completeness close to VGGT, whereas LiteVGGT and FlashVGGT degrade more noticeably on longer sequences.

4.4 Analysis and Ablation Studies

We further investigate the effectiveness of operator-induced representative selection, the contributions of individual components, and the generality of QuadVGGT across geometric models.

Representative selection. We compare token-feature cosine, projected-key cosine, and our operator-induced profile using identical anchor candidates and quotient settings. Only the selection criterion is changed. Table 4 shows that our profile achieves the lowest reconstruction errors across all reported metrics, reducing Abs Rel from 0.048 and 0.051 to 0.044 while also achieving the lowest inference time without increasing GPU memory. These results support operator-induced similarity as an effective representative-selection criterion. Appendix A.5 further shows that restricting attention to anchors without profile-based assignment degrades pose accuracy several-fold.

Table 4: **Representative selection.** Effect of different similarity criteria on 7-Scenes (500 frames).

Criterion	Abs Rel↓	Acc↓	Comp↓	APE↓	Time↓	Mem↓
Feature cos.	0.048	0.0142	0.0131	0.0384	21.28	31.23
Key cos.	0.051	0.0148	0.0126	0.0391	21.43	31.23
Profile (Ours)	0.044	0.0113	0.0119	0.0372	20.32	31.23

Operator-level analysis of quotient assignments. We further examine whether the operator-induced distances used for representative selection reflect the actual distortions introduced by Quotient Attention. For KV quotienting, the incoming-logit criterion consistently yields lower normalized mass distortion than feature- and key-cosine alternatives across global-attention layers (Fig. 7(a)). Moreover, candidates ranked as closer under the proposed KV distance exhibit smaller measured pairwise attention-weight discrepancies, with stronger alignment than representation-based criteria (Fig. 7(b)). For query quotienting, the centered-logit distance shows the strongest layer-wise correlation with the actual KL divergence induced by output sharing (Fig. 7(c)), directly supporting it as a surrogate for query-sharing distortion. Finally, these assignments are also geo-

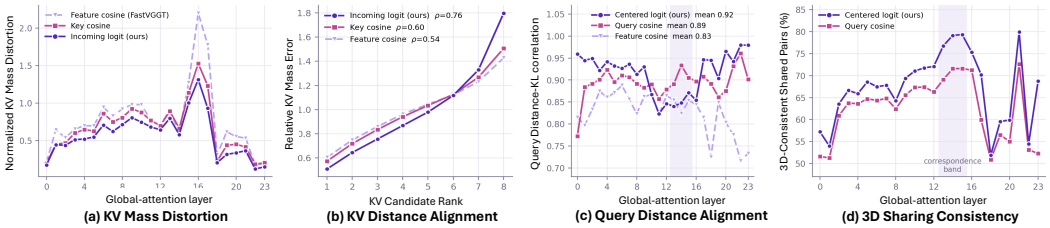


Figure 7: **Operator-level analysis of quotient assignments.** (a) Layer-wise normalized KV mass distortion under different selection criteria. (b) Incoming-logit distance aligns with measured pairwise attention-weight discrepancies across candidate ranks. (c) Centered-logit distance shows the strongest correlation with actual query-sharing KL divergence. (d) Centered-logit matching also yields more 3D-consistent shared pairs than query cosine. These results validate the proposed operator-induced distances as surrogates for attention distortion.

Table 5: **Component ablation on N-RGBD 1,000 frames.** Effect of individual components on reconstruction quality and efficiency.

Configuration	Abs Rel↓	Acc↓	Comp↓	APE↓	Time↓	Mem.↓
VGGT (Wang et al., 2025a)	0.0185	0.0087	0.0038	0.0544	664.0	72.4
+ HFO	0.0185	0.0087	0.0038	0.0544	648.6	55.9
+ KV Quotient	0.0102	0.0043	0.0037	0.0132	55.5	72.7
+ Query Quotient	0.0098	0.0042	0.0036	0.0138	232.4	72.7
+ All (KV + Query + HFO)	0.0096	0.0038	0.0034	0.0127	42.2	55.9

Table 6: **Comparison across geometric foundation models on ScanNet 500 frames.**

Method	Abs Rel↓	Acc↓	APE↓	Time↓	Mem.↓
DA3 (Lin et al., 2025)	0.0369	0.0091	0.0336	92.3	39.7
+ Ours	0.0372	0.0093	0.0339	27.3	32.6
π^3 (Wang et al., 2026d)	0.0303	0.0078	0.0314	134.57	25.5
+ Ours	0.0306	0.0076	0.0323	27.13	23.3
VGGT-Ω (Wang et al., 2026b)	0.0261	0.0066	0.0137	82.62	34.2
+ Ours	0.0258	0.0069	0.0140	26.07	23.5

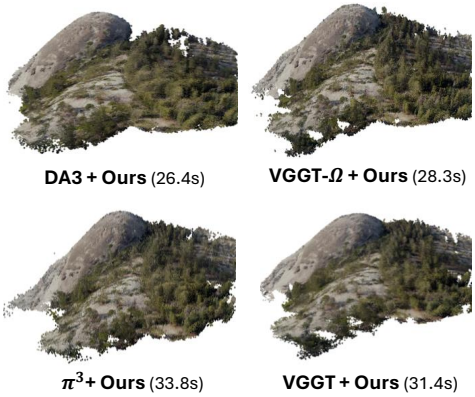


Figure 8: **Qualitative comparison across geometric models.** Reconstruction results on our custom 500-frame Yosemite sequence.

metrically meaningful: centered-logit matching more frequently pairs queries with representatives observing the same 3D point than query cosine (Fig. 7(d)). Together, these results show that the proposed probe distances capture the attention distortions relevant to quotient construction while preserving cross-view geometric consistency.

Component ablation. Table 5 isolates each component on top of VGGT and then combines them. Hierarchical Feature Offloading alone reduces peak GPU memory from 72.4 to 55.9 GB with identical reconstruction metrics. Key–value quotienting alone reduces runtime from 664.0 to 55.5 s and query quotienting alone to 232.4 s, both with similar accuracy and negligible memory overhead from profiling. Combining all three yields 42.2 s and 55.9 GB, confirming that Quotient Attention governs runtime while offloading governs memory. Delta-Walking is retained in all quotient configurations.

Generality across geometric foundation models. To assess the generality of Quotient Attention, we apply our method to DA3, π^3, and VGGT-Ω on 500-frame sequences from ScanNet. As shown in Table 6, our method reduces inference time from 92.3 to 27.3 s, 134.57 to 27.13 s, and 82.62 to 26.07 s, respectively, while maintaining comparable reconstruction accuracy with minor, metric-dependent variations. These results demonstrate that the proposed computation-sharing strategy is applicable beyond the original VGGT architecture. Figure 8 provides qualitative comparisons on our custom 500-frame Yosemite sequence, with inference times measured separately on this scene.

5 Conclusion

We presented QuadVGGT, a training-free framework that accelerates geometric foundation models by sharing redundant attention computations while preserving distinct dense token states. Quotient Attention uses sparse operator-induced profiles to select representatives and independently approximates key–value contributions and query outputs. Combined with Delta-Walking and Hierarchical Feature Offloading, it enables efficient long-sequence inference. Experiments demonstrate substantial reductions in runtime and GPU memory with competitive geometric accuracy, while attention-error analyses provide empirical support for operator-aware representative selection.

AI USE STATEMENT

We used generative AI tools to assist with manuscript writing and language refinement, and to obtain feedback on research methodology, mathematical formulations, and experimental interpretation. All AI-assisted suggestions were critically reviewed by the authors. The authors take full responsibility for the methodology, mathematical claims, experimental results, and final content of this paper.

ETHICS STATEMENT

This work focuses on improving the computational efficiency of existing geometric foundation models. While our method does not introduce new data collection, 3D reconstruction may raise privacy concerns when applied to sensitive imagery. We encourage responsible use of the proposed method and acknowledge the ethical considerations associated with the underlying models and datasets.

REPRODUCIBILITY STATEMENT

We provide detailed descriptions of the proposed method, implementation settings, and evaluation protocols in the main paper. Additional implementation details, ablation studies, and experimental analyses are provided in the appendix to facilitate reproducibility.

REFERENCES

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

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

D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black. A naturalistic open source movie for optical flow evaluation. In *ECCV*, 2012.

Yohann Cabon, Lucas Stoffl, Leonid Antsfeld, Gabriela Csurka, Boris Chidlovskii, Jerome Revaud, and Vincent Leroy. Must3r: Multi-view network for stereo 3d reconstruction, 2025. URL https://arxiv.org/abs/2503.01661.

Zhuoguang Chen, Minghui Qin, Tianyuan Yuan, Zhe Liu, and Hang Zhao. Long3r: Long sequence streaming 3d reconstruction. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 5273–5284. IEEE, 2025.

Krzysztof Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Davis, Afroz Mohiuddin, Lukasz Kaiser, David Belanger, Lucy Colwell, and Adrian Weller. Rethinking attention with performers, 2022. URL https://arxiv.org/abs/2009.14794.

Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *CVPR*, 2017.

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.

Kai Deng, Zexin Ti, Jiawei Xu, Jian Yang, and Jin Xie. Vggt-long: Chunk it, loop it, align it – pushing vggt's limits on kilometer-scale long rgb sequences, 2025. URL https://arxiv.org/abs/2507.16443.

Yushi Lan, Yihang Luo, Fangzhou Hong, Shangchen Zhou, Honghua Chen, Zhaoyang Lyu, Shuai Yang, Bo Dai, Chen Change Loy, and Xingang Pan. Stream3r: Scalable sequential 3d reconstruction with causal transformer, 2025. URL https://arxiv.org/abs/2508.10893.

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

Haotong Lin, Sili Chen, Jun Hao 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.

Yuzheng Liu, Siyan Dong, Shuzhe Wang, Yingda Yin, Yanchao Yang, Qingnan Fan, and Baoquan Chen. Slam3r: Real-time dense scene reconstruction from monocular rgb videos, 2025. URL https://arxiv.org/abs/2412.09401.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

E. Palazzolo, J. Behley, P. Lottes, P. Giguère, and C. Stachniss. ReFusion: 3D Reconstruction in Dynamic Environments for RGB-D Cameras Exploiting Residuals. In *IROS*, 2019. URL https://www.ipb.uni-bonn.de/pdfs/palazzolo2019iros.pdf.

Linfei Pan, Daniel Barath, Marc Pollefeys, and Johannes Lutz Schönberger. Global Structure-from-Motion Revisited. In *European Conference on Computer Vision (ECCV)*, 2024.

Xuan Peng, Xuanhua Shi, Hulin Dai, Hai Jin, Weiliang Ma, Qian Xiong, Fan Yang, and Xuehai Qian. Capuchin: Tensor-based gpu memory management for deep learning. In *Proceedings of the Twenty-Fifth International Conference on Architectural Support for Programming Languages and Operating Systems*, ASPLOS '20, pp. 891–905, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450371025. doi: 10.1145/3373376.3378505. URL https://doi.org/10.1145/3373376.3378505.

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.

Minsoo Rhu, Natalia Gimelshein, Jason Clemons, Arslan Zulfiqar, and Stephen W Keckler. vdnn: Virtualized deep neural networks for scalable, memory-efficient neural network design. In *2016 49th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*, pp. 1–13. IEEE, 2016.

Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

Thomas Schöps, Johannes L. Schönberger, 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 *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2538–2547, 2017. doi: 10.1109/CVPR.2017.272.

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 *CVPR*, 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.

Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Apoorv Vyas, Angelos Katharopoulos, and François Fleuret. Fast transformers with clustered attention. *Advances in Neural Information Processing Systems*, 33:21665–21674, 2020.

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 (CVPR)*, pp. 14546–14555, June 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, 2025a.

Jianyuan Wang, Minghao Chen, Shangzhan Zhang, Nikita Karaev, Johannes Schönberger, Patrick Labatut, Piotr Bojanowski, David Novotny, Andrea Vedaldi, and Christian Rupprecht. VGGT-Ω. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026b.

Qianqian Wang, Yifei Zhang, Aleksander Holynski, Alexei A Efros, and Angjoo Kanazawa. Continuous 3d perception model with persistent state. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10510–10522. IEEE, 2025b.

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.

Sinong Wang, Belinda Z. Li, Madian Khabsa, Han Fang, and Hao Ma. Linformer: Self-attention with linear complexity, 2020.

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, 2026c.

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 *International Conference on Learning Representations*, volume 2026, pp. 10481–10497, 2026d.

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.

Yuqi Wu, Wenzhao Zheng, Jie Zhou, and Jiwen Lu. Point3r: Streaming 3d reconstruction with explicit spatial pointer memory. *Advances in Neural Information Processing Systems*, 38:69675–69699, 2026.

Yunyang Xiong, Zhanpeng Zeng, Rudrasis Chakraborty, Mingxing Tan, Glenn Fung, Yin Li, and Vikas Singh. Nyströmformer: A nyström-based algorithm for approximating self-attention. 2021.

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, 2025.

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

Dong Zhuo, Wenzhao Zheng, Jiahe Guo, Yuqi Wu, Jie Zhou, and Jiwen Lu. Streaming visual geometry transformer. In *International Conference on Learning Representations*, volume 2026, pp. 88055–88072, 2026.

A Additional Analysis

Ablation studies were conducted in separate experimental runs; therefore, the default configuration may exhibit minor numerical variations across tables.

A.1 Hyperparameter Sensitivity

Sensitivity to the number of probes. Motivated by prior work on low-rank attention approximation (Wang et al., 2020), we examine the sensitivity of sparse operator profiling to the number of probes. We vary m from 4 to 64, using the same m for probe queries P_Q and probe keys P_K, with all other settings fixed ($\Delta^\star = 0.002$). Table 7 reports results on 7-Scenes with 1,000 frames and ETH3D (Schöps et al., 2017). Every 7-Scenes metric stays within 2% of the default $m = 16$ over this 16× range, and ETH3D AUC@30 stays within ± 0.003 of dense VGGT. These results suggest that sparse probes adequately capture the dominant interaction structure (Fig. 2(c)). We therefore regard $m = 16$ as a stable operating point rather than a sensitive constant.

Table 7: **Sensitivity to the number of probes m.** 7-Scenes with 1,000 frames and ETH3D (416 images, 11 scenes); $\Delta^\star = 0.002$. The default is shown in bold.

	7-Scenes, 1,000 frames										ETH3D	
m	Abs Rel↓	$\tau < 1.25$ ↑	Acc↓	Comp↓	CD↓	NC↑	APE↓	ARE↓	RPE-T↓	RPE-R↓	AUC@30↑	AUC@5↑
Dense VGGT	0.0433	0.9233	0.0119	0.0107	0.0113	0.6300	0.0450	0.0680	0.0064	0.0097	0.8224	0.4259
4	0.0432	0.9233	0.0108	0.0107	0.0107	0.6395	0.0418	0.0667	0.0055	0.0072	0.8225	0.4215
8	0.0431	0.9233	0.0108	0.0107	0.0107	0.6402	0.0419	0.0663	0.0054	0.0071	0.8240	0.4250
16	0.0431	0.9234	0.0108	0.0107	0.0107	0.6404	0.0420	0.0664	0.0054	0.0071	0.8224	0.4220
32	0.0431	0.9232	0.0108	0.0108	0.0108	0.6406	0.0417	0.0664	0.0054	0.0071	0.8203	0.4233
64	0.0432	0.9233	0.0108	0.0108	0.0108	0.6406	0.0417	0.0663	0.0053	0.0071	0.8199	0.4230

Delta-Walking anchor analysis. The walk radius $\Delta^\star$ sets the anchor density and hence the representative budget $|\mathcal{R}|/N$ (Table 8). The number of anchors adapts to the scene rather than to the sequence length: at the default $\Delta^\star = 0.002$, the 1,000-frame 7-Scenes sequences use 32–80 anchors (3–8% of the frames), whereas ETH3D, a collection of wide-baseline photographs with little redundancy, keeps 67% of its images as anchors. Halving $\Delta^\star$ roughly doubles the budget without a measurable benefit: on 7-Scenes every metric changes by at most 2%, and ETH3D AUC@30 decreases slightly. Coarser walks trade camera accuracy for depth and point-cloud metrics. At $\Delta^\star = 0.008$ ($|\mathcal{R}|/N = 2.0\%$), RPE-Trans increases by 12% and NC decreases by 0.010, while Abs Rel and completeness improve by 3% and 9%; the completeness gain coincides with the drop in normal consistency. At $\Delta^\star = 0.016$ ($|\mathcal{R}|/N = 1.0\%$), pose estimation degrades sharply (RPE-Trans $+80\%$, APE $+42\%$; ETH3D AUC@30 0.638). We use $\Delta^\star = 0.002$, which lies at the knee of this trade-off on both datasets.

Table 8: **Delta-Walking radius $\Delta^\star$.** $|\mathcal{R}|/N$: retained KV tokens on 7-Scenes (1,000 frames); Anchors: fraction of ETH3D images selected as anchors. $m = 16$. The default is shown in bold.

		7-Scenes, 1,000 frames										ETH3D		
$\Delta^\star$	$\lvert\mathcal{R}\rvert/N$	Abs Rel↓	$\tau < 1.25$ ↑	Acc↓	Comp↓	CD↓	NC↑	APE↓	ARE↓	RPE-T↓	RPE-R↓	Anchors	AUC@30↑	AUC@5↑
Dense VGGT	100%	0.0433	0.9233	0.0119	0.0107	0.0113	0.6300	0.0450	0.0680	0.0064	0.0097	100%	0.8224	0.4259
0.001	14.4%	0.0429	0.9235	0.0107	0.0105	0.0106	0.6411	0.0423	0.0660	0.0054	0.0072	82%	0.8168	0.4199
0.002	6.9%	0.0431	0.9234	0.0108	0.0107	0.0107	0.6404	0.0420	0.0664	0.0054	0.0071	67%	0.8224	0.4220
0.004	3.7%	0.0442	0.9239	0.0107	0.0104	0.0105	0.6376	0.0430	0.0675	0.0055	0.0072	54%	0.8176	0.4051
0.008	2.0%	0.0419	0.9224	0.0106	0.0097	0.0102	0.6308	0.0428	0.0671	0.0061	0.0077	38%	0.8022	0.3722
0.016	1.0%	0.0475	0.9198	0.0103	0.0080	0.0091	0.6171	0.0594	0.0813	0.0097	0.0095	22%	0.6375	0.2496

A.2 Input Order and Order-Robust Anchor Construction

Delta-Walking is designed for sequentially captured inputs, which constitute the long-sequence regime targeted in this work: video, handheld and robotic scans, and trajectory-based acquisition. Sequential 3D perception and long-sequence reconstruction have also been investigated in prior work (Wang et al., 2025b; Deng et al., 2025; Zhuo et al., 2026; Wang & Agapito, 2025; Lan et al., 2025; Chen et al., 2025; Wu et al., 2026; Cabon et al., 2025; Liu et al., 2025). All long-sequence benchmarks we evaluate (7-Scenes, N-RGBD, and ScanNet) follow this setting. In such inputs, consecutive frames are strongly correlated, so comparing each frame only with the most recent anchor

suffices to track scene coverage, and the two temporally nearest anchors provide natural representative candidates. This temporal prior departs from the permutation-equivariant aggregation of VGGT and introduces order dependence into the approximation. Unlike the sequential setting targeted here, conventional SfM pipelines also address unordered image collections (Schönberger & Frahm, 2016; Pan et al., 2024). To quantify this dependence, we randomly shuffle the 500-frame 7-Scenes sequences (Table 9). Because consecutive frames are no longer similar, the number of anchors grows from 29–44 to 440–467 per scene, and the speedup over shuffled dense VGGT drops to $1.2\times$ (152.1 s versus 179.2 s). Accuracy, however, matches shuffled dense VGGT: when the ordering assumption is violated, Delta-Walking becomes conservative rather than inaccurate. The same behavior appears on inputs without temporal redundancy, such as the wide-baseline photographs of ETH3D, where most frames are retained as anchors (Table 8).

This dependence stems from the anchor rule rather than from Quotient Attention itself. We therefore consider an order-robust variant, *Delta-cover*, which compares each frame against all anchors created so far rather than only the most recent one. Frames are processed in input order, and the first two frames are anchors. Using the same descriptor g_s and threshold $\Delta^\star$ as Delta-Walking, each subsequent frame f is handled as follows:

1. If $1 - \langle g_f, g_{a_{\mathrm{last}}} \rangle \le \Delta^\star$, f becomes a member of a_{last}, as in Delta-Walking.
2. Otherwise, let a^* be the anchor closest to f. If $1 - \langle g_f, g_{a^*} \rangle > \Delta^\star$, f becomes a new anchor.
3. Otherwise, f revisits a previously covered region, and we verify the match at the patch level: f becomes a member of a^* if its patch tokens correspond to those of a^* at least as well as to those of a_{last}, and a new anchor otherwise.

For each member frame, representative candidates are drawn from its assigned anchor and the two anchors nearest in camera-token space; the assignment rule within the candidates is unchanged. Every member frame thus retains a candidate anchor within $\Delta^\star$, as in Delta-Walking, and the number of anchors is governed by the number of $\Delta^\star$-balls needed to cover the visited viewpoints rather than by the input order. The rule is greedy and therefore not strictly permutation invariant, since which frames become anchors can change with the order. Its overhead is one descriptor comparison per existing anchor and frame, which is negligible.

On shuffled inputs, Delta-cover uses 19–30 anchors per scene, runs in 20.1 s ($8.9\times$ faster than shuffled dense VGGT), and stays within approximately 3.5% of ordered Delta-Walking across all order-independent metrics. On ordered input, however, Delta-cover is less accurate than Delta-Walking: RPE-Trans and RPE-Rot increase by 31% and 18%, APE and Abs Rel by 10% and 9%, and NC decreases by 0.016. Completeness improves, again together with lower normal consistency. We hypothesize that viewpoint-based candidates occasionally select visually similar but geometrically distinct anchors, which temporal adjacency avoids in sequential input; we have not isolated this effect. We therefore keep Delta-Walking as the default for the sequential-capture setting targeted in this work, and regard Delta-cover as an alternative only when the input order is unknown or unreliable. Designing a single anchor rule that matches Delta-Walking on ordered input while remaining robust to shuffling is left to future work.

Table 9: **Input-order sensitivity** on 7-Scenes (500 frames, averaged over seven scenes). "Shuffled" randomly permutes the input frames. Anchors are reported per scene. RPE is not reported for shuffled inputs, where adjacent-frame pairs no longer follow the original trajectory.

Method	Order	Anchors	Time (s)↓	Abs Rel↓	Comp↓	NC↑	APE↓	ARE↓	RPE-T↓	RPE-R↓
Dense VGGT	original	–	184.0	0.0435	0.0112	0.6318	0.0387	0.0629	0.0054	0.0076
Dense VGGT	shuffled	–	179.2	0.0435	0.0112	0.6316	0.0387	0.0629	–	–
Delta-Walking	original	29–44	20.3	0.0444	0.0119	0.6331	0.0372	0.0629	0.0052	0.0071
Delta-Walking	shuffled	440–467	152.1	0.0433	0.0110	0.6307	0.0386	0.0628	–	–
Delta-cover	original	25–36	20.9	0.0481	0.0099	0.6168	0.0410	0.0661	0.0068	0.0084
Delta-cover	shuffled	19–30	20.1	0.0429	0.0117	0.6312	0.0366	0.0641	–	–

A.3 Operator-Level Distortion Analysis

We provide additional details for the operator-level analysis in Fig. 7. Across all comparisons, the anchor assignments, candidate pools, representative budget, and probe count are fixed; only the criterion used to rank or select representatives is changed. This isolates whether the proposed operator-induced distances reflect the attention distortions introduced by quotient construction.

Layer-wise KV mass distortion. Figure 7(a) measures the KV mass-distortion term appearing in Theorem B.3 after representative selection. For layer ℓ, we report

$$\mathcal{E}_{\mathrm{KV}}^{(\ell)} = \mathbb{E}_{i,h}\left[\frac{D_{V,h}^{(\ell)}\ \mathrm{TV}\left(p_{i,h}^{(\ell)}, \bar{p}_{i,h}^{(\ell)}\right)}{\|y_{i,h}^{(\ell)}\|_2 + \epsilon_0}\right], \tag{10}$$

where ϵ_0 is a small numerical constant. The normalization removes layer-dependent output scale, allowing the mass distortion to be compared across global-attention layers. All selection criteria use the same anchor candidates and quotient configuration.

KV distance–mass alignment. Figure 7(b) examines whether the KV distance used for representative selection orders candidates according to their actual attention-mass distortion. For a key j and candidate representative a, recall

$$\delta_{ij,a}^{K} = z_{ij} - z_{ia}. \tag{11}$$

Since

$$\frac{\alpha_{ij}}{\alpha_{ia}} = e^{\delta_{ij,a}^{K}}, \tag{12}$$

we measure the relative pairwise mass error as

$$e^{K}(j,a) = \mathbb{E}_{i,h}\left[\left|e^{\delta_{ij,a}^{K}} - 1\right|\right]. \tag{13}$$

For each token, candidates are sorted according to the corresponding selection criterion, and Eq. (13) is averaged at each candidate rank. We additionally report Spearman rank correlation between the candidate distance and the measured mass error. A criterion that better reflects KV distortion should therefore assign smaller distances to candidates with smaller $e^{K}(j,a)$.

Query distance–KL alignment. Figure 7(c) directly evaluates the relation established in Proposition B.6. For each admissible query–candidate pair (i,a) and head h, we compute the actual sharing divergence

$$\kappa_h(i,a) = \mathrm{KL}\left(\bar{p}_{a,h} \| \bar{p}_{i,h}\right). \tag{14}$$

At each global-attention layer ℓ, we measure the Spearman correlation between the candidate distance and Eq. (14):

$$\rho^{(\ell)} = \frac{1}{H}\sum_{h=1}^{H} \rho_{\mathrm{Sp}}\left(\{d_h(i,a)\}_{(i,a)}, \{\kappa_h(i,a)\}_{(i,a)}\right). \tag{15}$$

Higher correlation indicates that the selection distance more faithfully orders candidates according to the actual attention-distribution distortion incurred by query sharing.

3D consistency of query sharing. Figure 7(d) examines whether low-distortion query sharing also corresponds to meaningful cross-view geometry. For every shared query–representative pair (i,a), we back-project the corresponding image locations using the reference geometry and determine whether they correspond to the same 3D surface point. We report

$$\mathcal{C}_{\mathrm{3D}}^{(\ell)} = 100\,\frac{\#\{\text{3D-consistent shared pairs at layer } \ell\}}{\#\{\text{shared pairs at layer } \ell\}}. \tag{16}$$

The same geometric correspondence criterion is used for all selection methods. This analysis is not used by QuadVGGT at inference time; it only evaluates whether the operator-induced assignments recover cross-view relationships that are geometrically meaningful.

Table 10: **Representative preservation versus averaging.** (a) Reconstruction quality and runtime on 500-frame sequences from 7-Scenes and N-RGBD. (b) Camera-pose accuracy on ETH3D. Dense VGGT is included as a reference. Best values among the three accelerated variants are shown in bold.

(a) Reconstruction quality and runtime.

Group treatment	Abs Rel↓	$\tau < 1.25$ ↑	Acc↓	Comp↓	CD↓	NC↑	APE↓	ARE↓	RPE-T↓	RPE-R↓	Time (s)↓
7-Scenes, 500 frames											
Dense VGGT	0.0435	0.9223	0.0110	0.0112	0.0111	0.6318	0.0387	0.0629	0.0054	0.0076	184.0
Original representatives	0.0444	0.9220	0.0113	0.0119	0.0116	**0.6331**	**0.0372**	**0.0629**	**0.0052**	**0.0071**	**20.3**
Value averaging	**0.0427**	0.9257	**0.0104**	**0.0095**	**0.0100**	0.6223	0.0439	0.0670	0.0068	0.0090	20.7
Key–value averaging	0.0615	**0.9264**	0.0205	**0.0095**	0.0150	0.5885	0.0951	0.1529	0.0150	0.0260	21.0
N-RGBD, 500 frames											
Dense VGGT	0.0116	0.9971	0.0034	0.0027	0.0031	0.8043	0.0140	0.0139	0.0025	0.0024	170.8
Original representatives	**0.0088**	0.9974	**0.0031**	**0.0027**	**0.0029**	**0.8168**	**0.0115**	**0.0168**	**0.0027**	**0.0025**	**17.9**
Value averaging	0.0189	**0.9976**	0.0080	0.0050	0.0065	0.7571	0.0429	0.0731	0.0057	0.0075	18.3
Key–value averaging	0.0442	0.9966	0.0233	0.0081	0.0157	0.6651	0.0858	0.1291	0.0155	0.0290	19.1

(b) ETH3D camera-pose accuracy.

Group treatment	AUC@30↑	AUC@5↑
Dense VGGT	0.8224	0.4259
Original representatives	0.8242	0.4268
Value averaging	0.8044	0.3791
Key–value averaging	**0.8283**	**0.4308**

Summary. The four analyses test complementary consequences of the proposed selection rules. Figure 7(a) measures the resulting KV mass distortion, while Fig. 7(b) tests whether the KV distance itself orders candidates by their measured mass error. Figure 7(c) performs the analogous test for query sharing using the KL divergence derived in Proposition B.6, and Fig. 7(d) verifies that the resulting query assignments remain consistent with cross-view 3D structure. Together, they empirically connect the sparse probe distances to the distributional distortion terms analyzed in Theorem B.3.

A.4 Representative Preservation versus Averaging

We examine how averaging within assigned KV groups affects geometric predictions. Table 10 compares three treatments: preserving the original representative key and value, averaging group values while retaining the original representative key, and averaging both group keys and values. All accelerated variants use the proposed operator-induced profiles for representative selection. We evaluate reconstruction quality and runtime on 500-frame sequences from 7-Scenes and N-RGBD, together with camera-pose accuracy on ETH3D.

Effects on reconstruction. On 7-Scenes, value averaging improves depth and point-distance metrics, but reduces normal consistency and increases APE, RPE-T, and RPE-R by 18.0%, 30.8%, and 26.8%, respectively, relative to original representatives. Joint KV averaging further degrades pose accuracy and normal consistency. The effect is more pronounced on N-RGBD, where original representatives achieve the best results in nine of the ten geometric metrics among the accelerated variants. Value averaging increases APE to $3.73\times$ the value obtained with original representatives. Joint KV averaging increases APE and RPE-R to $7.46\times$ and $11.60\times$, respectively, while NC decreases from 0.8168 to 0.6651. These differences occur at comparable runtimes: 20.3–21.0 seconds on 7-Scenes and 17.9–19.1 seconds on N-RGBD.

Dataset-dependent trade-offs. The benefit of preserving original representatives is not uniform across all metrics and datasets. In addition to the improvements in selected 7-Scenes metrics from averaging, joint KV averaging achieves the highest pose AUC on ETH3D, exceeding original rep-

Table 11: **Comparison with simple anchor baselines** on 7-Scenes (500 frames, averaged over seven scenes). All variants use the $\log M_a$ correction and identical candidate pools unless noted. Assignment is listed as query rule / KV rule.

	Anchors	Assignment (Q / KV)	Abs Rel↓	Comp↓	NC↑	APE↓	ARE↓	RPE-T↓	RPE-R↓	Time (s)↓
	Dense VGGT		0.0435	0.0112	0.6318	0.0387	0.0629	0.0054	0.0076	184.0
(a)	Delta-Walking	grid position / grid position	0.0784	0.0047	0.5614	0.0793	0.1046	0.0344	0.0522	17.6
(b)	Delta-Walking	random / random	0.1756	0.0085	0.5241	0.4369	0.6368	0.2334	0.3992	17.7
(c)	Uniform	profile / profile	0.0438	0.0117	0.6347	0.0381	0.0631	0.0053	0.0071	20.1
(d)	Delta-Walking	profile / profile (Ours)	0.0444	0.0119	0.6331	0.0372	0.0629	0.0052	0.0071	20.3
	Uniform	feat. cos. / feat. cos.	0.0430	0.0106	0.6245	0.0382	0.0630	0.0066	0.0083	17.1
	Uniform	grid position / feat. cos.	0.0440	0.0109	0.6248	0.0374	0.0631	0.0064	0.0079	17.9
	Delta-Walking	feat. cos. / feat. cos.	0.0484	0.0131	0.6278	0.0384	0.0625	0.0059	0.0076	21.2
	Delta-Walking	profile / profile, w/o $\log M_a$	0.0403	0.0065	0.5892	0.0424	0.0669	0.0059	0.0077	19.8

Table 12: **Simple anchor baselines on ETH3D** (11 scenes). Uniform anchors use the same anchor count as Delta-Walking.

	Anchors	Assignment (Q / KV)	AUC@30↑
	Dense VGGT		0.8224
(a)	Delta-Walking	grid position / grid position	0.6142
(b)	Delta-Walking	random / random	0.3946
(c)	Uniform	profile / profile	0.8183
(d)	Delta-Walking	profile / profile (Ours)	0.8214
	Uniform	feat. cos. / feat. cos.	0.7433

resentatives by 0.0041 at AUC@30 and 0.0040 at AUC@5. Nevertheless, original representatives provide the best pose accuracy and normal consistency among the accelerated variants on both reconstruction benchmarks, while maintaining competitive ETH3D pose accuracy. These results support preserving original representatives as the default, without implying that averaging is universally detrimental.

A.5 Comparison with Simple Anchor Baselines

Quotient Attention evaluates global attention only for anchor patches and camera/register tokens, and non-anchor tokens inherit representative outputs. We therefore examine whether the gains stem from the operator-induced assignment or merely from restricting global attention to anchor frames. Using Delta-Walking anchors, the $\log M_a$ correction, and identical candidate pools, we compare (a) same-position assignment without profiling, where each non-anchor token uses the patch at the same grid position in a candidate anchor frame as both its query and KV representative; (b) random assignment within the candidate pool for both queries and keys; (c) uniformly spaced anchors with the same anchor count as Delta-Walking and our assignment; and (d) the full method. We also report variants that assign representatives by token-feature cosine, and the full method without the $\log M_a$ correction (Tables 11 and 12).

Restricting global attention to anchor frames is not sufficient by itself. Same-position assignment (a) increases RPE-Trans and RPE-Rot by $6.6\times$ and $7.4\times$, doubles APE, lowers NC from 0.633 to 0.561, and reduces ETH3D AUC@30 from 0.821 to 0.614, while random assignment (b) degrades every pose metric by more than an order of magnitude. Replacing the profiles with feature cosine is far less harmful but still consistently weakens cross-view consistency: RPE-Trans increases by 13–27%, RPE-Rot by 7–17%, and NC decreases by 0.005–0.009; on ETH3D, changing only the assignment rule under identical uniform anchors reduces AUC@30 from 0.818 to 0.743. Anchor placement, in contrast, contributes little: uniform anchors (c) change every 7-Scenes metric by at most 2.4% and ETH3D AUC@30 by 0.3 points, so the main role of Delta-Walking is to set a scene-adaptive anchor budget.

Per-pixel depth and completeness behave differently. Feature-cosine variants with uniform anchors remain competitive in Abs Rel, and most simplified variants improve completeness, most strongly

Table 13: Dataset-specific dense 3D reconstruction using 100 input frames. Results are grouped by dataset rather than averaged across datasets. Lower is better except for $\delta_{1.25}$ and NC.

Dataset	Method	Depth		Point				Camera				Resource	
		Abs Rel↓	$\delta_{1.25}$ ↑	Acc↓	Comp↓	CD↓	NC↑	APE↓	ARE↓	RPE-T↓	RPE-R↓	Time (s)↓	Mem. (GB)↓
7-Scenes	VGGT	0.0320	0.9250	0.0110	0.0130	0.0120	0.6290	0.0170	0.0310	0.0030	0.0050	9.80	10.20
	FastVGGT	0.0350	0.9240	0.0130	0.0140	0.0130	0.5840	0.0190	0.0370	0.0040	0.0080	6.20	12.80
	LiteVGGT	0.0340	0.9310	0.0130	0.0140	0.0140	0.6150	0.0190	0.0310	0.0040	0.0060	4.40	12.70
	FlashVGGT	0.0330	0.9290	0.0150	0.0170	0.0160	0.6190	0.0150	0.0260	0.0030	0.0050	4.20	12.60
	SparseVGGT	0.0330	0.9280	0.0120	0.0110	0.0120	0.5870	0.0200	0.0310	0.0040	0.0060	5.40	12.80
	SAF3R	0.0310	0.9260	0.0110	0.0110	0.0110	0.6210	0.0170	0.0300	0.0030	0.0050	8.00	12.80
	QuadVGGT (Ours)	0.0330	0.9270	0.0110	0.0130	0.0120	0.6220	0.0190	0.0320	0.0030	0.0050	4.00	10.17
N-RGBD	VGGT	0.0110	0.9920	0.0050	0.0060	0.0050	0.8410	0.0030	0.0050	0.0010	0.0010	9.30	12.60
	FastVGGT	0.0140	0.9910	0.0070	0.0070	0.0070	0.8120	0.0050	0.0120	0.0020	0.0030	6.50	12.83
	LiteVGGT	0.0160	0.9764	0.0090	0.0050	0.0050	0.7257	0.0065	0.0184	0.0030	0.0031	5.00	12.60
	FlashVGGT	0.0170	0.9920	0.0120	0.0120	0.0120	0.7290	0.0070	0.0130	0.0010	0.0010	4.10	12.60
	SparseVGGT	0.0130	0.9920	0.0080	0.0050	0.0060	0.6900	0.0070	0.0180	0.0030	0.0040	5.40	12.80
	SAF3R	0.0110	0.9920	0.0050	0.0060	0.0060	0.8130	0.0040	0.0040	0.0010	0.0010	7.80	12.80
	QuadVGGT (Ours)	0.0110	0.9920	0.0040	0.0050	0.0050	0.8270	0.0040	0.0060	0.0020	0.0020	4.00	12.10
ScanNet-50	VGGT	0.0248	0.9807	0.0074	0.0092	0.0082	0.6837	0.0190	0.0330	0.0038	0.0048	9.19	12.60
	FastVGGT	0.0257	0.9803	0.0094	0.0099	0.0097	0.6485	0.0201	0.0356	0.0039	0.0061	7.07	12.86
	LiteVGGT	0.0379	0.9807	0.0120	0.0107	0.0098	0.6537	0.0210	0.0340	0.0035	0.0050	3.80	12.80
	FlashVGGT	0.0220	0.9930	0.0090	0.0090	0.0090	0.6540	0.0190	0.0360	0.0040	0.0050	4.30	12.81
	SparseVGGT	0.0310	0.9810	0.0100	0.0080	0.0090	0.6280	0.0220	0.0420	0.0050	0.0070	5.31	12.80
	SAF3R	0.0260	0.9810	0.0080	0.0080	0.0080	0.6640	0.0190	0.0330	0.0040	0.0050	8.11	12.80
	QuadVGGT (Ours)	0.0280	0.9799	0.0080	0.0100	0.0090	0.6750	0.0190	0.0340	0.0050	0.0050	4.03	12.80

Table 14: Dataset-specific dense 3D reconstruction using 500 input frames. Results are grouped by dataset rather than averaged across datasets. Lower is better except for $\delta_{1.25}$ and NC.

Dataset	Method	Depth		Point				Camera				Resource	
		Abs Rel↓	$\delta_{1.25}$ ↑	Acc↓	Comp↓	CD↓	NC↑	APE↓	ARE↓	RPE-T↓	RPE-R↓	Time (s)↓	Mem. (GB)↓
7-Scenes	VGGT	0.0430	0.9220	0.0110	0.0110	0.0110	0.6320	0.0390	0.0630	0.0050	0.0080	184.00	39.20
	FastVGGT	0.0440	0.9230	0.0130	0.0100	0.0100	0.6010	0.0410	0.0680	0.0050	0.0090	43.10	40.21
	LiteVGGT	0.0430	0.9290	0.0140	0.0100	0.0120	0.5990	0.0420	0.0600	0.0070	0.0100	37.50	26.70
	FlashVGGT	0.0340	0.9310	0.0100	0.0080	0.0090	0.6130	0.0350	0.0560	0.0040	0.0070	26.80	39.20
	SparseVGGT	0.0480	0.9260	0.0130	0.0090	0.0110	0.5880	0.0480	0.0740	0.0070	0.0090	62.50	42.70
	SAF3R	0.0420	0.9230	0.0110	0.0100	0.0110	0.6230	0.0360	0.0610	0.0050	0.0070	63.50	41.40
	QuadVGGT (Ours)	0.0440	0.9220	0.0110	0.0120	0.0120	0.6340	0.0370	0.0630	0.0050	0.0070	22.10	31.20
N-RGBD	VGGT	0.0120	0.9970	0.0030	0.0030	0.0030	0.8040	0.0140	0.0140	0.0020	0.0020	180.10	39.20
	FastVGGT	0.0140	0.9960	0.0050	0.0050	0.0050	0.7620	0.0160	0.0210	0.0030	0.0040	42.53	40.21
	LiteVGGT	0.0230	0.9960	0.0060	0.0030	0.0040	0.6970	0.0260	0.0270	0.0050	0.0080	36.30	26.70
	FlashVGGT	0.0190	0.9970	0.0090	0.0040	0.0070	0.7270	0.0170	0.0370	0.0030	0.0020	25.50	39.20
	SparseVGGT	0.0240	0.9970	0.0100	0.0040	0.0070	0.6820	0.0430	0.0610	0.0060	0.0090	60.30	42.70
	SAF3R	0.0090	0.9970	0.0040	0.0030	0.0040	0.7830	0.0110	0.0140	0.0020	0.0020	63.50	41.40
	QuadVGGT (Ours)	0.0090	0.9970	0.0030	0.0030	0.0030	0.8170	0.0110	0.0170	0.0030	0.0020	20.10	31.00
ScanNet-50	VGGT	0.0299	0.9841	0.0121	0.0109	0.0121	0.6940	0.0550	0.0850	0.0089	0.0161	182.71	39.20
	FastVGGT	0.0305	0.9943	0.0126	0.0093	0.0102	0.6323	0.0534	0.0898	0.0067	0.0077	42.80	40.21
	LiteVGGT	0.0381	0.9841	0.0130	0.0101	0.0119	0.6591	0.0541	0.0900	0.0090	0.0129	36.81	26.70
	FlashVGGT	0.0319	0.9919	0.0149	0.0129	0.0140	0.6661	0.0500	0.0900	0.0071	0.0099	26.21	38.99
	SparseVGGT	0.0360	0.9780	0.0160	0.0090	0.0129	0.6320	0.0779	0.1539	0.0131	0.0369	63.00	42.70
	SAF3R	0.0270	0.9840	0.0100	0.0090	0.0099	0.6829	0.0469	0.0750	0.0071	0.0111	60.89	41.40
	QuadVGGT (Ours)	0.0301	0.9829	0.0100	0.0111	0.0111	0.6991	0.0480	0.0760	0.0079	0.0090	23.59	31.49

same-position assignment (0.0047 versus 0.0119), often at the expense of normal consistency. The same co-occurrence appears for coarser walks (Table 8), group averaging (Table 10), and removing the $\log M_a$ correction, which gives the best Abs Rel among all variants but reduces NC to 0.589 and increases APE by 14%. We report this co-occurrence to make the trade-off explicit, without claiming a causal mechanism. Runtime differences are small: the simplified variants take 17.1–21.2 s versus 20.3 s for the full method, so the accuracy gaps do not reflect a runtime trade-off.

A.6 Dataset-Specific Reconstruction Results

Tables 13–15 report the dataset-specific results underlying the aggregate long-sequence comparison in Table 3. At 1,000 frames, QuadVGGT improves normal consistency and absolute camera-pose

Table 15: Dataset-specific dense 3D reconstruction using 1,000 input frames. Results are grouped by dataset rather than averaged across datasets. OOM indicates that the method could not complete inference under the evaluated GPU-memory budget. Lower is better except for $\delta_{1.25}$ and NC.

Dataset	Method	Depth		Point				Camera				Resource	
		Abs Rel↓	$\delta_{1.25}$ ↑	Acc↓	Comp↓	CD↓	NC↑	APE↓	ARE↓	RPE-T↓	RPE-R↓	Time (s)↓	Mem. (GB)↓
7-Scenes	VGGT	0.0433	0.9233	0.0119	0.0107	0.0113	0.6300	0.0450	0.0680	0.0064	0.0097	670.40	72.40
	FastVGGT	0.0508	0.9288	0.0152	0.0098	0.0111	0.6120	0.0416	0.0520	0.0048	0.0096	94.49	72.60
	LiteVGGT	0.0467	0.9303	0.0164	0.0096	0.0130	0.5937	0.0536	0.0988	0.0088	0.0125	108.20	51.10
	FlashVGGT	0.0331	0.9357	0.0088	0.0057	0.0073	0.6148	0.0409	0.0579	0.0045	0.0065	72.60	72.40
	SparseVGGT	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM
	SAF3R	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM
	QuadVGGT (Ours)	0.0428	0.9235	0.0108	0.0109	0.0109	0.6414	0.0418	0.0671	0.0050	0.0068	48.20	55.90
N-RGBD	VGGT	0.0185	0.9947	0.0087	0.0038	0.0063	0.7794	0.0544	0.1217	0.0077	0.0118	664.00	72.40
	FastVGGT	0.0258	0.9937	0.0099	0.0028	0.0058	0.7708	0.0380	0.0286	0.0029	0.0108	90.00	72.60
	LiteVGGT	0.0345	0.9945	0.0161	0.0041	0.0101	0.6983	0.0770	0.1574	0.0104	0.0234	107.90	51.10
	FlashVGGT	0.0223	0.9950	0.0103	0.0053	0.0078	0.7247	0.0222	0.0348	0.0033	0.0031	71.00	72.40
	SparseVGGT	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM
	SAF3R	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM
	QuadVGGT (Ours)	0.0096	0.9950	0.0038	0.0034	0.0036	0.8264	0.0127	0.0166	0.0033	0.0030	42.20	55.90
ScanNet-50	VGGT	0.0321	0.9830	0.0151	0.0119	0.0136	0.6843	0.0650	0.1058	0.0108	0.0214	700.41	72.40
	FastVGGT	0.0383	0.9911	0.0151	0.0111	0.0134	0.6944	0.0581	0.1057	0.0097	0.0213	100.00	72.60
	LiteVGGT	0.0394	0.9822	0.0200	0.0130	0.0165	0.6592	0.0827	0.1377	0.0120	0.0193	110.21	51.10
	FlashVGGT	0.0349	0.9928	0.0154	0.0127	0.0140	0.6708	0.0578	0.0990	0.0081	0.0117	73.09	72.40
	SparseVGGT	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM
	SAF3R	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM	OOM
	QuadVGGT (Ours)	0.0298	0.9825	0.0115	0.0106	0.0110	0.7009	0.0541	0.0894	0.0091	0.0127	47.00	55.90

error over VGGT on all three evaluated datasets. Improvements are metric-dependent: for example, completeness on 7-Scenes and depth threshold accuracy on ScanNet-50 are slightly lower than VGGT. On shorter sequences, the relative accuracy varies by dataset and metric. These results support the conclusion that the proposed approximation maintains competitive geometric fidelity while reducing long-sequence inference cost, without implying universal improvement in every metric or individual scene. The results are aggregated within each dataset; they do not measure scene-wise variance.

B Analysis of Quotient Attention

B.1 Setup

We analyze a single global-attention (Vaswani et al., 2017) layer with fixed inputs Q, K, V and fixed representative assignments. All logits are finite, and attention dropout is disabled. We omit the head index unless needed.

The KV assignment partitions the token indices into nonempty groups $\{\mathcal{S}_a\}_{a\in\mathcal{R}}$, with multiplicities $M_a = |\mathcal{S}_a|$. For query i, define

$$\alpha_i = \mathrm{softmax}(z_{i\cdot}), \qquad p_{ia} = \sum_{j\in\mathcal{S}_a} \alpha_{ij}, \qquad \mu_{ia} = \frac{1}{p_{ia}} \sum_{j\in\mathcal{S}_a} \alpha_{ij} v_j. \tag{17}$$

The original output is therefore $y_i = \sum_{a\in\mathcal{R}} p_{ia}\mu_{ia}$. The KV approximation is

$$\bar{z}_{ia} = z_{ia} + \log M_a, \qquad \bar{p}_i = \mathrm{softmax}(\bar{z}_i), \qquad \bar{y}_i = \sum_{a\in\mathcal{R}} \bar{p}_{ia} v_a. \tag{18}$$

For analysis, $\bar{p}_i$ and $\bar{y}_i$ are defined for every query, although the implementation evaluates them only for $i \in \mathcal{U}$. The final approximation is $\hat{y}_i = \bar{y}_{\pi_Q(i)}$.

These results compare the original and approximate operators at the same layer input. They do not directly establish an end-to-end error bound across multiple layers.

B.2 Exact Group Aggregation and KV Approximation

The exact logit associated with group $\mathcal{S}_a$ is

$$s_{ia} = \log \sum_{j \in \mathcal{S}_a} \exp(z_{ij}). \tag{19}$$

Since the groups partition all tokens,

$$p_i = \mathrm{softmax}(s_i), \qquad y_i = \sum_{a \in \mathcal{R}} p_{ia} \mu_{ia}. \tag{20}$$

Thus, grouping itself is an exact re-expression of attention. The approximation arises when group logits and conditional values are replaced by representative quantities.

Specifically,

$$s_{ia} = \bar{z}_{ia} + \varepsilon_{ia}, \qquad \varepsilon_{ia} = \log \left(\frac{1}{M_a} \sum_{j \in \mathcal{S}_a} \exp(z_{ij} - z_{ia}) \right). \tag{21}$$

Consequently,

$$p_i = \mathrm{softmax}(\bar{z}_i + \varepsilon_i), \qquad \bar{p}_i = \mathrm{softmax}(\bar{z}_i). \tag{22}$$

For later use, define the key-substitution logit perturbation

$$\delta^K_{ij,a} = z_{ij} - z_{ia} = \frac{q_i^\top (k_j - k_a)}{\sqrt{d}}. \tag{23}$$

Then Eq. (21) is equivalently

$$\varepsilon_{ia} = \log \mathbb{E}_{j \sim \mathrm{Unif}(\mathcal{S}_a)} \left[e^{\delta^K_{ij,a}} \right]. \tag{24}$$

Proposition B.1 (Exact KV aggregation)**.** *For a fixed query i, the group masses satisfy $p_i = \bar{p}_i$ if and only if ε_{ia} is constant over $a \in \mathcal{R}$. In particular, if $z_{ij} = z_{ia}$ for every $j \in \mathcal{S}_a$ and every $a \in \mathcal{R}$, then the group masses are preserved exactly. If additionally $\mu_{ia} = v_a$ for every group, then $y_i = \bar{y}_i$.*

Proof. For finite vectors u and v, $\mathrm{softmax}(u) = \mathrm{softmax}(v)$ if and only if $u - v$ is a constant vector. Indeed, equality of the normalized exponentials implies

$$u_a - v_a = \log \sum_c e^{u_c} - \log \sum_c e^{v_c},$$

which is independent of a; the converse follows from softmax shift invariance. Applying this fact to Eq. (22) proves the first statement. Identical logits within each group give $\varepsilon_{ia} = 0$ in Eq. (21). Finally, substituting $p_i = \bar{p}_i$ and $\mu_{ia} = v_a$ into Eq. (20) gives $y_i = \bar{y}_i$. □

The condition $v_j = v_a$ for every $j \in \mathcal{S}_a$ is sufficient for $\mu_{ia} = v_a$. These are exactness conditions, rather than assumptions imposed on the groups selected in practice.

B.3 Proof of the Output-Distortion Bound

We first establish a bound for expectations over the representative values.

Lemma B.2 (Value diameter and total variation)**.** *Let $D_V = \max_{a,c \in \mathcal{R}} \|v_a - v_c\|_2$. For any probability vectors r, t on $\mathcal{R}$,*

$$\left\| \sum_{a \in \mathcal{R}} (r_a - t_a) v_a \right\|_2 \le D_V \, \mathrm{TV}(r, t). \tag{25}$$

Proof. Write $\beta_a = r_a - t_a$ and $s = \mathrm{TV}(r, t)$. Because $\sum_a \beta_a = 0$, the positive and negative parts of β each have total mass s. If $s = 0$, the claim is immediate. Otherwise, define

$$v^+ = \frac{1}{s} \sum_{\beta_a > 0} \beta_a v_a, \qquad v^- = \frac{1}{s} \sum_{\beta_a < 0} (-\beta_a) v_a.$$

Both are convex combinations of representative values, and $\sum_a \beta_a v_a = s(v^+ - v^-)$. Writing their difference as a convex combination of pairwise differences gives $\|v^+ - v^-\|_2 \le D_V$. The result follows. □

Theorem B.3 (Quotient Attention distortion). *Let $b = \pi_Q(i)$. Then*

$$\begin{aligned}\|y_i - \hat{y}_i\|_2 \le & \sum_{a\in\mathcal{R}} p_{ia}\|\mu_{ia} - v_a\|_2 \\ & + D_V \,\mathrm{TV}(p_i, \bar{p}_i) \\ & + D_V \,\mathrm{TV}(\bar{p}_i, \bar{p}_b).\end{aligned} \tag{26}$$

Proof. Using $\hat{y}_i = \bar{y}_b$, we obtain the exact decomposition

$$\begin{aligned}y_i - \hat{y}_i = & \sum_{a\in\mathcal{R}} p_{ia}(\mu_{ia} - v_a) \\ & + \sum_{a\in\mathcal{R}} (p_{ia} - \bar{p}_{ia})v_a \\ & + \sum_{a\in\mathcal{R}} (\bar{p}_{ia} - \bar{p}_{ba})v_a.\end{aligned} \tag{27}$$

Apply the triangle inequality to the first sum and Lemma B.2 to the remaining sums. Restoring the head index and setting $b = \pi_{Q,h}(i)$ gives Eq. (9). □

The first two terms bound KV approximation error. The last term bounds query sharing under the fixed KV quotient. For directly evaluated queries, $b = i$ and the last term vanishes, while KV approximation error may remain.

Corollary B.4 (Bounded within-group variation). *For a fixed query i, suppose*

$$|z_{ij} - z_{ia}| \le \eta, \qquad \|v_j - v_a\|_2 \le \nu$$

for every $a \in \mathcal{R}$ and $j \in \mathcal{S}_a$. Then

$$\|y_i - \bar{y}_i\|_2 \le \nu + D_V \min\{1, e^{2\eta} - 1\}. \tag{28}$$

Proof. Equation (21) gives $|\varepsilon_{ia}| \le \eta$. Hence,

$$\frac{p_{ia}}{\bar{p}_{ia}} = \frac{e^{\varepsilon_{ia}}}{\sum_c \bar{p}_{ic} e^{\varepsilon_{ic}}} \in [e^{-2\eta}, e^{2\eta}],$$

which implies the conservative bound

$$\mathrm{TV}(p_i, \bar{p}_i) \le \min\{1, e^{2\eta} - 1\}.$$

Since μ_{ia} is a convex combination of values in $\mathcal{S}_a$, $\|\mu_{ia} - v_a\|_2 \le \nu$. Apply Theorem B.3 with $b = i$. □

The KV probe distance in Eq. (3) measures the squared logit deviations appearing in the condition of Corollary B.4 on a sparse set of probe queries, providing an operator-level surrogate for representative selection.

B.4 Query Equivalence and Logit Variance

For a fixed KV quotient, define query equivalence in each head by equality of the reduced attention distributions.

Proposition B.5 (Exact query-distribution sharing). *Suppressing the head index,*

$$\bar{p}_i = \bar{p}_b \quad \Longleftrightarrow \quad z_{i\mathcal{R}} - z_{b\mathcal{R}} \in \mathrm{span}(\mathbf{1}). \tag{29}$$

This condition implies $\bar{y}_i = \bar{y}_b$.

Proof. The reduced logits are $z_{i\mathcal{R}} + \log M$ and $z_{b\mathcal{R}} + \log M$. The common multiplicity correction cancels in their difference. The softmax equivalence established in the proof of Proposition B.1 therefore gives Eq. (29). Equal distributions yield equal weighted sums of the representative values. □

The converse implication from equal outputs to equal distributions need not hold: distinct distributions can produce the same output for a particular value matrix.

For approximate sharing, let

$$\ell_b = z_{b\mathcal{R}} + \log M, \qquad \delta = z_{i\mathcal{R}} - z_{b\mathcal{R}}, \qquad p_t = \mathrm{softmax}(\ell_b + t\delta).$$

Thus, $p_0 = \bar{p}_b$ and $p_1 = \bar{p}_i$.

Proposition B.6 (Variance representation of query divergence)**.** *The query-distribution divergence satisfies*

$$\begin{aligned} \mathrm{KL}(\bar{p}_b \| \bar{p}_i) &= \log \mathbb{E}_{\bar{p}_b}[e^{\delta}] - \mathbb{E}_{\bar{p}_b}[\delta] \\ &= \int_0^1 (1-t)\,\mathrm{Var}_{p_t}(\delta)\,dt. \end{aligned} \tag{30}$$

Proof. The normalized distributions satisfy

$$\bar{p}_{ia} = \frac{\bar{p}_{ba} e^{\delta_a}}{\mathbb{E}_{\bar{p}_b}[e^{\delta}]}.$$

Substitution into the definition of KL divergence yields the first equality. For the second, define

$$f(t) = \log \sum_a e^{\ell_{ba} + t\delta_a}.$$

Then

$$f'(t) = \mathbb{E}_{p_t}[\delta], \qquad f''(t) = \mathrm{Var}_{p_t}(\delta).$$

The divergence equals $f(1) - f(0) - f'(0)$. Taylor's theorem with integral remainder yields the result. □

Let

$$\delta_c = \delta - \mathbb{E}_{\bar{p}_b}[\delta]\mathbf{1}.$$

As $\|\delta_c\|_\infty \to 0$,

$$\mathrm{KL}(\bar{p}_b \| \bar{p}_i) = \frac{1}{2}\,\mathrm{Var}_{\bar{p}_b}(\delta) + O(\|\delta_c\|_\infty^3). \tag{31}$$

Moreover, Lemma B.2 and Pinsker's inequality give

$$\|\bar{y}_i - \bar{y}_b\|_2 \le D_V\,\mathrm{TV}(\bar{p}_i, \bar{p}_b) \le D_V \sqrt{\frac{1}{2}\,\mathrm{KL}(\bar{p}_b \| \bar{p}_i)}. \tag{32}$$

These relations motivate comparing centered logit discrepancies when selecting queries for output sharing.

B.5 Interpretation of Sparse Operator Profiles

We now relate the sparse profile distances used for representative selection to the perturbations appearing in the preceding analysis.

KV-side operator metric. For $\Delta k = k_j - k_a$, the KV distance in Eq. (3) satisfies

$$d^K(j,a) = \frac{1}{md}\left\|Q_{P_Q}\Delta k\right\|_2^2. \tag{33}$$

Define the probe-query second-moment matrix as

$$\widehat{S}_Q = \frac{1}{m}\sum_{p \in P_Q} q_p q_p^\top. \tag{34}$$

Expanding Eq. (33) gives

$$d^K(j,a) = \frac{1}{d}\Delta k^\top \widehat{S}_Q \Delta k. \tag{35}$$

Equivalently, using

$$\delta^K_{pj,a} = \frac{q_p^\top (k_j - k_a)}{\sqrt{d}},$$

we obtain

$$d^K(j,a) = \frac{1}{m}\sum_{p\in P_Q} \left(\delta^K_{pj,a}\right)^2. \tag{36}$$

Thus, the KV criterion measures key differences along directions observed by the sampled probe queries.

For any query i, the same perturbation satisfies

$$\delta^K_{ij,a} = z_{ij} - z_{ia} = \log \frac{\alpha_{ij}}{\alpha_{ia}}. \tag{37}$$

Proposition B.7 (Local expansion of the KV group-logit residual)**.** *For a fixed query i and group $\mathcal{S}_a$, let*

$$\bar{\delta}^K_{ia} = \mathbb{E}_{j\sim \mathrm{Unif}(\mathcal{S}_a)}\left[\delta^K_{ij,a}\right]. \tag{38}$$

Then

$$\varepsilon_{ia} = \bar{\delta}^K_{ia} + \log \mathbb{E}_{j\sim \mathrm{Unif}(\mathcal{S}_a)}\left[e^{\delta^K_{ij,a} - \bar{\delta}^K_{ia}}\right]. \tag{39}$$

As

$$\max_{j\in\mathcal{S}_a} \left|\delta^K_{ij,a} - \bar{\delta}^K_{ia}\right| \to 0,$$

we have

$$\varepsilon_{ia} = \bar{\delta}^K_{ia} + \frac{1}{2}\,\mathrm{Var}_{j\in\mathcal{S}_a}\left(\delta^K_{ij,a}\right) + O\Big(\big\|\delta^K_{i,\mathcal{S}_a} - \bar{\delta}^K_{ia}\mathbf{1}\big\|^3_\infty\Big). \tag{40}$$

Proof. Write

$$\delta^K_{ij,a} = \bar{\delta}^K_{ia} + \widetilde{\delta}^K_{ij,a},$$

where $\mathbb{E}_j[\widetilde{\delta}^K_{ij,a}] = 0$. Equation (24) becomes

$$\varepsilon_{ia} = \bar{\delta}^K_{ia} + \log \mathbb{E}_j\left[e^{\widetilde{\delta}^K_{ij,a}}\right].$$

The Taylor expansion of the log moment-generating function around zero gives

$$\log \mathbb{E}_j\left[e^{\widetilde{\delta}^K_{ij,a}}\right] = \frac{1}{2}\mathbb{E}_j\left[(\widetilde{\delta}^K_{ij,a})^2\right] + O\Big(\|\widetilde{\delta}^K_{i,\mathcal{S}_a}\|^3_\infty\Big),$$

which proves the result. □

The sparse KV distance and Eq. (40) characterize the same key-substitution logit perturbation from complementary views: $d^K(j,a)$ evaluates a key–representative pair over sampled queries, whereas ε_{ia} aggregates perturbations within a group for a fixed query. We therefore use d^K as a sparse representative-selection surrogate.

Proposition B.8 (Variance representation of KV mass divergence)**.** *For a fixed query i, define*

$$r_t = \mathrm{softmax}\left(\bar{z}_i + t\varepsilon_i\right), \qquad t \in [0,1], \tag{41}$$

so that $r_0 = \bar{p}_i$ and $r_1 = p_i$. Then

$$\begin{aligned}\mathrm{KL}(\bar{p}_i \| p_i) &= \log \mathbb{E}_{\bar{p}_i}\left[e^{\varepsilon_i}\right] - \mathbb{E}_{\bar{p}_i}\left[\varepsilon_i\right] \\ &= \int_0^1 (1-t)\,\mathrm{Var}_{r_t}(\varepsilon_i)\,dt.\end{aligned} \tag{42}$$

Proof. From Eq. (22),

$$p_{ia} = \frac{\bar{p}_{ia} e^{\varepsilon_{ia}}}{\mathbb{E}_{\bar{p}_i}[e^{\varepsilon_i}]}.$$

Substitution into the definition of $\mathrm{KL}(\bar{p}_i \| p_i)$ gives the first equality. For the second, define

$$g(t) = \log \sum_a e^{\bar{z}_{ia} + t\varepsilon_{ia}}.$$

Then

$$g'(t) = \mathbb{E}_{r_t}[\varepsilon_i], \qquad g''(t) = \mathrm{Var}_{r_t}(\varepsilon_i).$$

The divergence equals $g(1) - g(0) - g'(0)$. Taylor's theorem with integral remainder gives Eq. (42). □

Let

$$\varepsilon_i^c = \varepsilon_i - \mathbb{E}_{\bar{p}_i}[\varepsilon_i]\mathbf{1}.$$

As $\|\varepsilon_i^c\|_\infty \to 0$,

$$\mathrm{KL}(\bar{p}_i \| p_i) = \frac{1}{2} \mathrm{Var}_{\bar{p}_i}(\varepsilon_i) + O(\|\varepsilon_i^c\|_\infty^3). \tag{43}$$

Pinsker's inequality therefore gives

$$D_V \, \mathrm{TV}(p_i, \bar{p}_i) \le D_V \sqrt{\frac{1}{2} \mathrm{KL}(\bar{p}_i \| p_i)}. \tag{44}$$

Together with Theorem B.3, this relates the KV group-logit perturbation to the mass-distortion term in the output bound.

Query-side operator metric. For $\Delta q = q_i - q_a$, the query distance in Eq. (4) satisfies

$$d(i,a) = \frac{1}{md} \left\| C_m K_{P_K} \Delta q \right\|_2^2. \tag{45}$$

Thus, $d(i,a) = 0$ if and only if the logit difference is constant over the probe keys. This is the restriction of the equivalence condition in Eq. (29) to the probe set.

Define the probe-key mean and covariance as

$$\bar{k}_P = \frac{1}{m} \sum_{p \in P_K} k_p, \qquad \widehat{\Sigma}_K = \frac{1}{m} \sum_{p \in P_K} (k_p - \bar{k}_P)(k_p - \bar{k}_P)^\top.$$

Expanding Eq. (45) gives

$$d(i,a) = \frac{1}{d} \Delta q^\top \widehat{\Sigma}_K \Delta q. \tag{46}$$

The distance therefore weights query differences according to the directions observed by the probe keys.

Under the additional condition $\|q_i\|_2 = \|q_a\|_2 = r$, query cosine instead corresponds to

$$\|\Delta q\|_2^2 = 2r^2\big(1 - \cos(q_i, q_a)\big).$$

The two criteria can rank candidates differently when the key covariance is anisotropic.

The probe distance is a uniformly weighted variance over sampled keys, whereas Eq. (31) involves an attention-weighted variance over retained KV representatives. These quantities are not identical. Likewise, equality on the probes need not imply equality on all retained keys. We therefore use the probe distance as a theoretically motivated selection surrogate, rather than a certificate of output error. Low effective dimensionality motivates sparse probing; the adequacy of a particular probe count is assessed empirically.

Together, the KV and query criteria characterize logit perturbations through sparse interactions with the opposite side of the attention operator. The resulting probe distances serve as operator-aware surrogates for the two distributional distortion terms in Theorem B.3.

B.6 Head-wise Selection and Output Projection

For each shared query, the common candidate $b(i)$ belongs to every head's candidate set. Consequently, the selection rule in Eq. (7) guarantees

$$d_h(i, \pi_{Q,h}(i)) \le d_h(i, b(i)), \qquad \sum_h d_h(i, \pi_{Q,h}(i)) \le \sum_h d_h(i, b(i)). \tag{47}$$

Thus, head-wise selection cannot increase the summed selection distance relative to using the common candidate. This statement concerns the surrogate distance and does not guarantee a corresponding decrease in actual output error.

When RoPE (Su et al., 2024) depends only on within-image position, patches at the same grid position receive the same rotation. The additional candidate $b'(i)$ therefore removes the difference in positional rotations. If the pre-RoPE queries also coincide, the resulting queries are identical and can share their attention output exactly. In general, content differences remain, so the candidate is evaluated using the centered logit distance.

Finally, write the original output projection in head blocks as $W_o^{(h)}$. Then

$$o_i - \hat{o}_i = \sum_h W_o^{(h)}(y_{i,h} - \hat{y}_{i,h}), \qquad \|o_i - \hat{o}_i\|_2 \le \sum_h \|W_o^{(h)}\|_2 \|y_{i,h} - \hat{y}_{i,h}\|_2. \tag{48}$$

Applying Theorem B.3 to each head bounds the projected attention error. Adding the same input residual preserves this difference, while retaining a distinct residual state for each token.

Relation to clustered attention and token merging. Fast Transformers with Clustered Attention (Vyas et al., 2020) provides a closely related approach to sharing attention computations. Its basic construction groups queries, evaluates attention using cluster centroids, and broadcasts the resulting outputs to the queries within each cluster. The grouping can differ across attention heads, while residual connections preserve token-specific representations. Quotient Attention shares these principles; query-output sharing, head-dependent assignments, and distinct residual states are therefore not, by themselves, its distinguishing features.

Our formulation differs in how representatives and assignments are constructed. Quotient Attention uses original token queries and key–value pairs as representatives, with separate assignments for KV approximation and query-output sharing. On the KV side, incoming-logit profiles guide assignment to retained representatives, whose contributions are adjusted by group multiplicity. On the query side, centered-logit profiles guide head-wise selection among original anchor queries. Whereas the basic clustered-attention construction groups queries by proximity in query space, these criteria measure discrepancies through sampled interactions with the opposite side of the attention operator. In particular, query-profile centering accounts for the invariance of softmax to a common logit shift. The analysis above characterizes the resulting value-substitution, KV-mass, and query-sharing distortions, motivating these selection criteria without treating sparse-profile agreement as a guarantee of exact output preservation.

The proposed design choices are further supported by complementary empirical analyses. Table 10 isolates the effect of representative preservation from feature averaging, revealing substantial improvements in several geometric metrics, particularly camera-pose accuracy, while also showing dataset-dependent trade-offs. Table 11 evaluates profile-based assignments against simpler alternatives, and Fig. 7 directly examines their alignment with measured attention distortion and 3D consistency.

These experiments validate the individual design choices underlying Quotient Attention. Together with the preceding analysis, they establish the motivation for independently quotienting both sides of the attention operator using operator-induced representatives.