RegimeVGGT: Layer-Wise Spatially Preserving Redundancy Removal for Visual Geometry Grounded Transformer

Jinhao You[1,*] **Shuo Lyu**[1,*,†] **Zhuohang Lyu**[1,*] **Tanxuan Li**[1,*]
Zibo Zhao[1,*] **Jiaxiang Hu**[2] **Kai Tang**[3] **Yichen Guo**[3]
[1]University of Pennsylvania
[2]University of California, Irvine
[3]Nanyang Technological University

Abstract

Visual Geometry Grounded Transformer (VGGT) recovers dense 3D scene structure from multi-view images in one forward pass, but quadratic cross-frame attention limits its scalability. Existing training-free accelerators reduce computation uniformly along one axis, missing layer heterogeneity. Our spectral, probing, and causal analyses reveal three regimes: shallow layers lack cross-view structure, middle layers drive cross-view alignment, and deep layers are redundant for dense geometry yet their cross-frame attention remains essential for pose. RegimeVGGT applies layer-wise U-shaped compression along two axes: *Saliency-Guided Banded Merging* protects geometry- and edge-salient tokens, while *Selectively Protected K/V Downsampling* preserves cross-frame spatial coverage and the pose-critical path through a phase-shifted spatial grid, a reference-frame anchor, and uncompressed camera/register tokens. Training-free, RegimeVGGT achieves a $6.7\times$ speedup over VGGT* at matched reconstruction quality. Project page: `https://tracyou07.github.io/regimevggt/`

1 Introduction

Recovering the 3D structure of a scene from multi-view images is fundamental to autonomous driving [9, 5], robotics [29, 19], and augmented reality [11], all of which demand efficient, end-to-end inference pipelines. Traditional multi-stage pipelines such as COLMAP [24, 23] are computationally expensive and brittle in challenging conditions [30, 1]. Recent feed-forward models, pioneered by DUSt3R [34] and extended by subsequent works [16, 33, 39], regress dense 3D geometry directly from uncalibrated images, but typically operate on image pairs with costly global alignment or focus on a single output modality.

VGGT [32] addresses both limitations with a unified transformer that accepts an arbitrary number of images and jointly predicts parameters for 3D reconstruction in a single forward pass. At its core is a 24-layer aggregator with alternating frame-wise and global cross-frame attention, where each patch token is geometrically grounded to a specific 3D surface point through dedicated prediction heads. This design incurs $O(S^2P^2)$ cost that grows quadratically with both the number of frames S and patches per frame P. As a result, the vanilla VGGT runs out of memory at roughly 300 frames on an NVIDIA H800 GPU, and even the memory-optimized variant VGGT* remains prohibitively slow on long sequences.

[*]Equal contribution.
[†]Corresponding author.

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

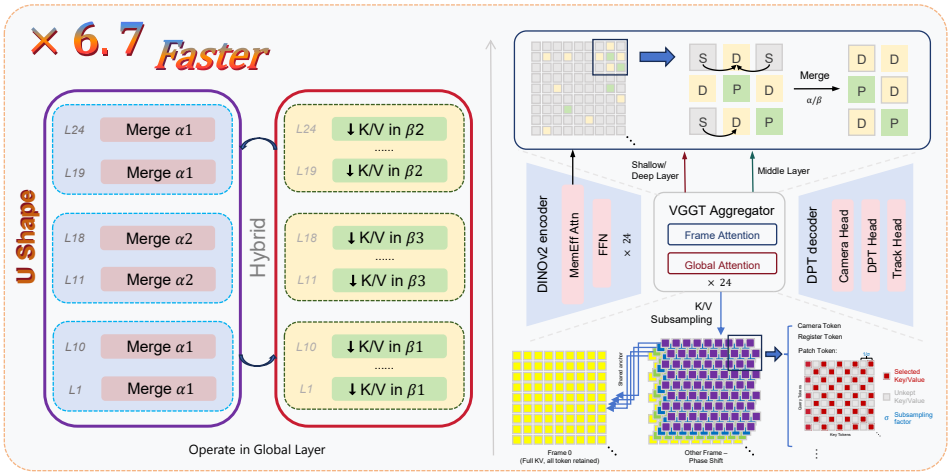


Figure 1: **Overview of RegimeVGGT.** VGGT's 24 global-attention layers expose a three-band rank structure: shallow (L1–L10) and deep (L19–L24) bands have effective rank ≤ 20, while the middle band (L11–L18) peaks at ≈ 90 where cross-view alignment is concentrated. The same band partition configures two orthogonal training-free operators: *Saliency-Guided Banded Merging* compresses the token-count axis with a U-shaped per-band merge ratio and a band-aligned index cache, and *Selectively Protected K/V Downsampling* compresses the K/V-set axis with a phase-shifted per-frame grid plus a full-density anchor at the reference frame. The two operators compose into a single global block in RegimeVGGT.

Concurrent training-free accelerators reduce VGGT's bottleneck along different axes: FastVGGT [25] merges tokens uniformly across all 24 layers, while S-VGGT [17] partitions frames into independent subscenes. AVGGT [28] goes further by establishing that VGGT's layers are not equally redundant: it converts early global attention to per-frame mode and uniformly subsamples K/V on a spatial grid in the remaining layers. Yet two finer questions remain open: $\mathcal{Q}_1$*: How does VGGT's aggregator partition along depth into functionally distinct regimes?* $\mathcal{Q}_2$*: Within each such regime, how should its functional role and internal structure inform compression design?*

We answer both through a spectral, probing, and causal analysis that uncovers three bands holding across scenes. The middle band (L11–L18) drives most of the point-cloud reconstruction, the shallow band (L1–L10) lacks cross-view structure, and the deep band (L19–L24), though geometrically inert, does not admit aggressive compression: doing so collapses long-sequence camera-pose estimation. Each band exposes two complementary compressible axes (a *token-count axis* and a *K/V-density axis*) with different headroom: the middle band has high effective rank but concentrates information in a small set of geometry- and edge-salient tokens at 3D boundaries and along 2D image structure, while its dense attention limits K/V sparsification; the shallow band has effective rank $\lesssim 20$ and the deep band is near-rank-1, both admitting aggressive compression along the rank dimension. Beyond these per-band patterns, pose stability imposes a global constraint cutting across all bands: pose lives in the cross-frame attention between camera/register[8] tokens and the surrounding patches, requiring patches to remain in this path throughout.

These findings translate directly into RegimeVGGT, a training-free framework with two operators composed within every global block, both following the same three-band partition. *Saliency-Guided Banded Merging* protects geometry- and edge-salient tokens on the token-count axis via a U-shape merge ratio that mirrors the rank profile (aggressive on the low-rank shallow and deep bands, conservative on the rank-heavy middle), guided by a frozen DINOv2 class-attention saliency Ψ that approximates this protection target at zero aggregator-state cost; matching is amortised by band-aligned caching across consecutive layers. *Selectively Protected K/V Downsampling* compresses the K/V-density axis under the same U-shape pattern while honouring the cross-band pose path: queries always participate, but each non-anchor frame's patch K/V is restricted to a phase-shifted sub-grid whose offsets make the cross-frame union approach full coverage, while a full-density frame-0 anchor and uncompressed cam/reg tokens preserve the cross-frame attention between cam/reg and patches at every layer. RegimeVGGT reaches $6.7\times$ speedup over VGGT* on ScanNet-1000.

2 Related Work

Feed-forward 3D reconstruction. Classical multi-view reconstruction relies on feature matching, triangulation, and bundle adjustment [23, 24], later enhanced by learned cost volumes [40] but still limited to calibrated setups. Recent feed-forward approaches relax these constraints: DUSt3R [34] predicts dense pointmaps from uncalibrated pairs, Fast3R [39] and CUT3R [33] scale to hundreds of views, VGGT [32] jointly estimates geometry and camera poses, and π^3 [36] improves permutation robustness. However, their global cross-frame attention incurs $O(S^2P^2)$ cost and introduces 3D-specific redundancy that generic transformer acceleration cannot address.

Token reduction in vision transformers. Token reduction has been extensively explored for accelerating vision transformers. Pruning-based methods [22] drop less informative tokens, reorganization approaches such as EViT [18] retain salient ones, and ToMe [4] merges redundant tokens via training-free similarity. Later works extend token merging to generative models, pruning–merging hybrids, vision–language systems, and long-form video [3, 13, 38, 15]. Other studies highlight structured token roles: Haurum et al. [10] reveal systematic sensitivity variation, while multimodal fusion [37] exploits cross-modal redundancy. Such findings indicate that token-aware reduction is crucial in structured domains; in 3D reconstruction, this heterogeneity has explicit geometric origins, with boundary and smooth-region tokens differing markedly in importance.

Accelerating VGGT. Several concurrent works accelerate VGGT along different axes. At the token level, FastVGGT [25] applies ToMe/ToMeSD-style bipartite merging [4, 3] uniformly across layers, LiteVGGT [27] introduces geometry-aware cached merging requiring fine-tuning and FP8, and HTTM [35] tailors merging to VGGT's global attention via head-wise temporal merging, spatio-temporal reordering, and outlier filtering. At the attention-pattern level, Faster VGGT [31] replaces dense attention with SpargeAttention-style block-sparse kernels [41] by exploiting geometry-concentrated probability mass. At the frame level, S-VGGT [17] partitions inputs into independently processed subscenes, while StreamVGGT [42] distills VGGT into a causal streaming aggregator. At the layer level, AVGGT [28] converts early layers to frame attention but still relies on uniform grid-based K/V downsampling. In contrast, our method simultaneously exploits heterogeneity along two compression axes.

3 Method

3.1 Analysis: The Case for Dual-Axis Compression

Layer-wise U-shape redundancy. We diagnose VGGT's 24 global-attention layers along three independent dimensions, all converging to a narrow middle band. *Information availability* (ridge probe; Appendix A.1.1) is recoverable mainly across L11–L18; *information utilization* (key-permutation intervention; Appendix A.1.2) peaks at the camera head there and vanishes outside; *spectral structure* of the per-head attention probability matrices (Appendix A.1.3) is inverted-U-shaped, peaking in the middle and dropping sharply on both flanks. Cross-view geometric alignment is therefore concentrated in L11–L18; the flanks are redundant for *distinct* reasons: shallow because cross-view content is not yet operative, deep because attention has collapsed into low-dimensional refinement. This U-shape redundancy invites two orthogonal compression axes, the **token axis** and the **K/V axis**.

Heterogeneous geometry tokens and middle-band concentration. Within the middle band not all patch tokens are equally important. We rank patches by the squared residual of regressing ground-truth depth on low-level 2D cues; the top subset, the *geometry tokens*, concentrates around depth discontinuities, object contours, and high-frequency 3D boundaries not explainable by 2D cues (formal definition in Appendix A.2.1). Figure 2 verifies they (i) align spatially with depth edges rather than appearance texture, (ii) disrupt the depth head under ablation substantially more than both random ablation and a Sobel-ranked image-edge baseline on 3D-rich scenes, with the gap narrowing on geometrically flat scenes where 3D boundaries are scarce, and (iii) accumulate across L11–L18, the same band where the probe and CI localize alignment. Thus the middle band must preserve this small subset; how it is approximated and protected at inference is detailed in *Saliency-Guided Banded Merging* (Section 3.2).

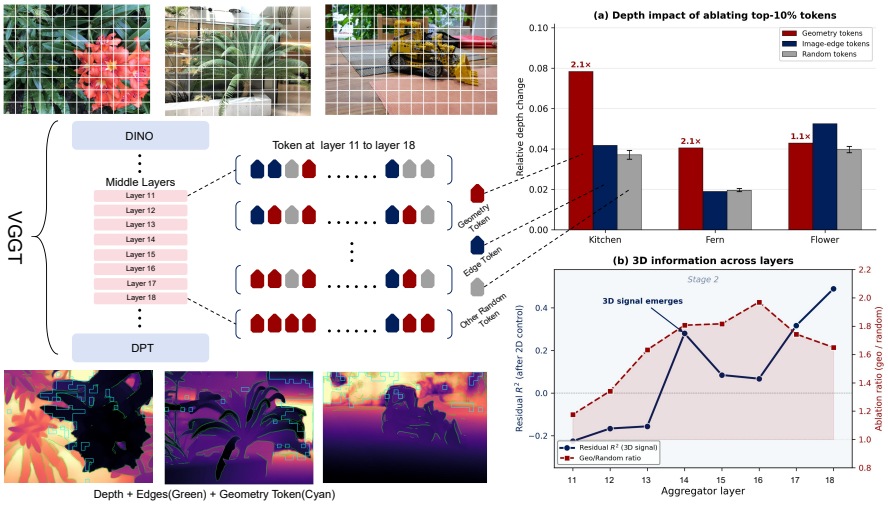


Figure 2: **Heterogeneous geometry tokens and middle-band concentration.** (a) Ablating the top-10% patch tokens by squared depth residual (after regressing out 2D cues) disrupts the depth head 2.1× more than random ablation and more than a Sobel-ranked image-edge baseline on 3D-rich scenes; the gap narrows on flat scenes (Flower), confirming the score captures 3D boundaries rather than 2D texture. Bottom panels overlay geometry-token locations (cyan) on depth edges (green). (b) Residual R^2 accumulates across L11–L18, the band where probe and CI localize cross-view alignment. The middle band must preserve both its rank and this small heterogeneous subset.

Asymmetric token roles and spectral K/V redundancy. Two findings on the K/V axis expose its redundancy. First, token classes play *asymmetric* roles: the camera head and the pose probe both read only from the camera token, so camera and register tokens carry the final pose signal across frames—empirically, skipping deep-band global attention preserves dense reconstruction but collapses pose (Appendix A.3), confirming they need a full cross-frame access path while patch tokens admit aggressive K/V compression. Second, the rank spectrum above quantifies each layer's K/V redundancy: shallow and deep layers concentrate attention in a single subspace, even the rank-heavy middle band uses only a fraction of its capacity, and the redundancy follows a U-shape. The design and the empirical path that arrived at it are detailed in *Selectively Protected K/V Downsampling* (Section 3.3) and Appendix A.3, respectively.

3.2 Saliency-Guided Banded Merging

We compress the token axis: at every global block we run a bipartite merge–unmerge operator, with the merge schedule and cache schedule both governed by the three bands of Section 3.1, and the protected token subset anchored by a frozen DINOv2 `[CLS]` importance score Ψ. Throughout, $L = T + 5$ denotes the per-frame token count (T patches plus camera and register tokens), and $L'_b \approx (1 - \rho_b)\, T + 5$ is the post-merge count in band b.

Per-band merge ratio. At each global block we apply the bipartite merge–unmerge of ToMe [4]: every non-reference frame's patch tokens are split into $\mathcal{A}_s$ and $\mathcal{B}_s$, the top-$\rho\,|\mathcal{A}_s|$ cosine-most-similar pairs are averaged, and the merged values are written back after global attention so frame attention and DPT [21] taps see full resolution. The top-α tokens under Ψ (defined below) and all reference-frame tokens are excluded from $\mathcal{A}_s$ throughout. The per-band ratios follow the rank prior:

$$\rho^{(\ell)} = \begin{cases} \rho_s = 0.99 & \ell \in [1, 10] \\ \rho_m = 0.50 & \ell \in [11, 18] \\ \rho_d = 0.99 & \ell \in [19, 24] \end{cases} \tag{1}$$

The U-shape, aggressive on the low-rank shallow and deep bands while conservative on the rank peak, tracks the rank analysis (Appendix A.1.3): shallow and deep layers are dominated by a single singular component, while the middle band retains substantial spread across the spectrum.

The protection set Ψ uses frozen DINOv2 saliency. The middle-band emergence of 3D-boundary information (Appendix A.2.1) requires per-layer protection of a small token subset; we hold out the top-α patches ($\alpha = 0.1$) under an importance score Ψ. We take Ψ to be the head-averaged `[CLS]` class-attention of the frozen DINOv2 encoder's final block,

$$\Psi_p \;=\; \frac{1}{H}\sum_{h=1}^{H} \mathrm{softmax}\left(\frac{q_{\mathrm{CLS}}^{(h)}(\mathbf{k}_p^{(h)})^\top}{\sqrt{d_h}}\right). \tag{2}$$

DINOv2's self-distillation and masked patch prediction make `[CLS]` attention *edge-aware but not edge-only* (Figure 8) [20]: it lands on salient objects rather than tracing every image edge. Because these object boundaries coincide with both depth discontinuities and high-frequency 2D edges, the high-Ψ patches spatially overlap with the geometry- and edge-token protection target of Appendix A.2.1, and no ground-truth depth is needed at inference.

Three-band cache and size-aware attention. We add two operator-level optimizations. *First*, a fixed-Δ index cache leaks across rank regimes; we instead align cache windows to band boundaries, computing matches at $\ell \in \{1, 11, 19\}$ and reusing them within each band, except for the rank-peak sub-range L11–L14 which is recomputed every layer. *Second*, after merging each surviving key represents $g_k \geq 1$ source tokens, and the softmax must be weighted by the group-size vector $\mathbf{g} = (g_1, \ldots, g_N)^\top$:

$$\mathrm{Attn}^\star(\mathbf{Q}, \mathbf{K}, \mathbf{V}; \mathbf{g}) \;=\; \mathrm{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d}} \;+\; \log \mathbf{g}^\top\right)\mathbf{V}. \tag{3}$$

Rather than implementing the bias as an attention mask [4], we absorb it into $\mathbf{Q}, \mathbf{K}$ themselves: append a column of 1s to $\mathbf{Q}$ and a column of $\sqrt{d}\log \mathbf{g}$ to $\mathbf{K}$, so the appended dimension contributes exactly $\log g_k$ to each logit under the standard $1/\sqrt{d}$ scale. The head dimension is zero-padded to the next FlashAttention-supported size [7] ($64 \to 80$); original output channels are recovered by slicing.

3.3 Selectively Protected K/V Downsampling

We compress the K/V axis: at every global block, query-side updates are preserved while each non-anchor frame's K/V is restricted to a phase-shifted spatial sub-grid at a per-band rate σ_b, with cam/reg tokens, the frame-0 anchor, and the geometry- and edge-salient patches selected by Ψ (Section 3.2) held at full density. The three sub-mechanisms (rate, sub-grid, and protection) are described next.

Per-band K/V rate. The keeper density follows the same three-band rank profile as the merge ratio, but inverted: densest K/V in the rank-peak middle band, sparse on the low-rank flanks. Concretely:

$$\sigma^{(\ell)} = \begin{cases} \sigma_s = 1.5 & \ell \in [1, 10] \quad \text{(kept fraction} \approx 44\%\text{)} \\ \sigma_m = 1.3 & \ell \in [11, 18] \quad \text{(kept fraction} \approx 56\%\text{)} \\ \sigma_d = 1.7 & \ell \in [19, 24] \quad \text{(kept fraction} \approx 36\%\text{)} \end{cases} \tag{4}$$

Each band's kept fraction is implemented on a discrete tile (t, k) with $(k/t)^2 \approx 1/\sigma_b^2$ (tile sizes in Appendix A.3.3).

Phase-shifted per-frame sub-grid. A uniform sub-grid—the same in every frame—would suppress the same spatial coordinates and leave systematic cross-frame blind spots. We instead *phase-shift* the sub-grid by frame index: row and column offsets at frame f are $(f \operatorname{div} t) \bmod t$ and $f \bmod t$, where t is the tile size set by σ_b. Different frames thus retain disjoint sub-grids, and the union of S frames' kept positions approaches full spatial coverage as S grows, at no extra per-frame cost.

Selective protection: cam/reg, anchor frame, and geometry- and edge-salient patches. Three classes are kept at full K/V density. **Camera and register tokens** carry the asymmetric pose signal (Section 3.1) and require a full cross-frame access path. **The frame-0 anchor** is held at full density at every layer: phase-shifting alone gives only union coverage across S frames, but each layer still needs a stable, fully resolved world reference for the middle-band alignment to regress against; an ablation in Appendix A.3 confirms single-anchor at frame 0 outperforms no-anchor, two-anchor, and rotating-anchor variants. **Geometry- and edge-salient patches** share the same DINOv2 `[CLS]` score Ψ as token merging (Section 3.2); the top-α patches under Ψ approximate the geometry+edge protection target (Appendix A.2.1) at zero additional cost. Combined with the merging axis, the per-block cost reduces to $O(S^2(L_b')^2/\sigma_b^2) + O(SL^2)$, up to protected-token terms (Appendix B).

4 Experiments

4.1 Experimental Setup

Datasets. We evaluate RegimeVGGT on dense 3D reconstruction and camera pose estimation. For dense reconstruction, we use 7 Scenes [26], NRGBD [2], and ScanNet-50, a 50-scene subset of ScanNet [6]. For camera pose estimation, we use the Tanks & Temples training scenes [14], DTU [12], and ScanNet-50 at varying sequence lengths.

Metrics. For 3D reconstruction, we report Accuracy (Acc↓), Completeness (Comp↓), and Normal Consistency (NC↑), each as mean and median across scenes; for ScanNet-50 we follow FastVGGT [25] and report Chamfer Distance (CD↓). For camera pose, we report AUC at $5^\circ/15^\circ/30^\circ$ thresholds of relative pose error (max of rotation and translation angular error). ScanNet-50 long-sequence evaluation additionally uses Absolute Trajectory Error (ATE↓), Absolute Rotation Error (ARE↓), and Relative Pose Error in rotation and translation (RPE-rot↓ and RPE-trans↓). Wall-clock time is reported throughout.

Baselines and Implementation Details. We compare RegimeVGGT with VGGT [32], the memory-optimized VGGT baseline VGGT*, two concurrent training-free acceleration methods, FastVGGT [25] and S-VGGT [17], and two feed-forward 3D reconstruction baselines, Fast3R [39] and CUT3R [33]. AVGGT [28] is excluded from the experimental tables because neither its code nor model configurations are publicly available, precluding evaluation under a unified protocol. A systematic design-level and theoretical comparison is provided in Appendix C. All experiments are conducted on a single NVIDIA H800 GPU with 80 GB memory at input resolution 518×392.

4.2 3D Reconstruction

Short indoor RGB-D sequences. Table 1 reports point-cloud reconstruction on 7 Scenes and NRGBD with keyframes sampled every 3 or 10 frames. This setting evaluates whether RegimeVGGT can reduce redundant global-attention computation while preserving dense local surface geometry on short indoor RGB-D sequences. On 7 Scenes, RegimeVGGT achieves the fastest inference under both stride settings, reducing runtime to 13.9s at stride 3 and 3.0s at stride 10, while matching the best Accuracy and keeping Completeness and Normal Consistency close to the strongest VGGT-based baselines. On NRGBD, RegimeVGGT also gives the fastest runtime, with 21.8s at stride 3 and 4.2s at stride 10, while maintaining competitive reconstruction quality across all three metrics. These results indicate that RegimeVGGT removes redundant global-attention computation without sacrificing the dense geometric information needed for accurate surface reconstruction.

Table 1: Quantitative results of point cloud reconstruction on the 7 Scenes and NRGBD datasets. Keyframes are sampled every 3 or 10 frames. *OOM* denotes out-of-memory. Best results among VGGT-based methods are in **bold**.

Method	Stride 3							Stride 10						
	Acc ↓		Comp ↓		NC ↑		Time ↓	Acc ↓		Comp ↓		NC ↑		Time ↓
	Mean	Med.	Mean	Med.	Mean	Med.		Mean	Med.	Mean	Med.	Mean	Med.	
7 Scenes														
VGGT *CVPR'25*	OOM							OOM						
Fast3R *CVPR'25*	0.045	0.027	0.047	0.010	0.605	0.657	33.0s	0.040	0.021	0.056	0.013	0.628	0.687	4.1s
CUT3R *CVPR'25*	0.179	0.121	0.097	0.043	0.577	0.611	10.9s	0.041	0.021	0.029	0.010	0.640	0.707	3.2s
VGGT* *CVPR'25*	0.018	0.008	0.028	0.012	0.600	0.658	56.0s	0.019	0.008	0.029	0.013	0.609	0.675	6.5s
FastVGGT *ICLR'26*	0.017	0.008	0.027	0.012	**0.603**	0.662	18.1s	0.017	0.008	0.028	0.012	0.611	0.677	3.5s
S-VGGT *ICME'26*	0.021	0.010	**0.021**	**0.009**	0.602	**0.663**	17.1s	0.025	0.012	**0.025**	**0.011**	**0.619**	**0.688**	4.0s
RegimeVGGT	**0.017**	**0.008**	0.027	0.012	0.602	0.661	**13.9s**	**0.017**	**0.008**	0.028	0.012	0.611	0.676	**3.0s**
NRGBD														
VGGT *CVPR'25*	OOM							OOM						
Fast3R *CVPR'25*	0.074	0.036	0.024	0.011	0.588	0.732	52.0s	0.061	0.028	0.031	0.013	0.599	0.762	5.6s
CUT3R *CVPR'25*	0.346	0.243	0.184	0.090	0.509	0.673	13.8s	0.132	0.064	0.056	0.011	0.599	0.775	4.3s
VGGT* *CVPR'25*	0.028	0.018	0.018	0.010	0.657	0.779	107.3s	**0.016**	**0.010**	**0.017**	**0.009**	0.668	0.790	10.4s
FastVGGT *ICLR'26*	0.031	0.020	0.019	0.010	**0.662**	**0.788**	29.1s	0.017	0.011	0.018	0.010	0.670	0.792	4.9s
S-VGGT *ICME'26*	0.027	0.019	**0.016**	0.009	0.636	0.725	28.0s	0.040	0.029	0.019	0.011	0.659	0.761	4.7s
RegimeVGGT	**0.024**	**0.014**	0.018	**0.009**	0.659	0.784	**21.8s**	0.018	0.011	0.018	0.010	**0.670**	**0.794**	**4.2s**

Scaling to long sequences. Table 2 evaluates reconstruction on ScanNet-50 with 100–1000 input images, directly stressing the quadratic cost of global attention. Under this ScanNet-50 protocol, the original VGGT runs out of memory at all sequence lengths, and S-VGGT also fails at 1000 frames. At 1000 frames, RegimeVGGT reduces runtime to 71.5s, compared with 477.3s for VGGT* and 113.5s for FastVGGT, while keeping CD nearly unchanged. It is also the fastest VGGT-based method at 500, 300, and 100 frames, and achieves the best VGGT-based CD at 500 frames. These results show that RegimeVGGT preserves reconstruction quality while making long-sequence VGGT inference substantially more scalable.

Table 2: Quantitative results of point cloud reconstruction on the ScanNet-50 dataset with input sequences of 1000, 500, 300, and 100 images. *OOM* denotes out-of-memory. Best results among VGGT-based methods are in **bold**.

	1000 frames		500 frames		300 frames		100 frames	
Method	CD ↓	Time ↓	CD ↓	Time ↓	CD ↓	Time ↓	CD ↓	Time ↓
VGGT *CVPR'25*	OOM		OOM		OOM		OOM	
Fast3R *CVPR'25*	0.684	299.0s	0.701	73.0s	0.711	26.2s	0.723	3.6s
CUT3R *CVPR'25*	0.786	26.1s	0.774	14.1s	0.775	8.3s	0.767	2.7s
VGGT* *CVPR'25*	**0.463**	477.3s	0.467	130.2s	0.454	48.7s	0.438	6.8s
FastVGGT *ICLR'26*	0.464	113.5s	0.453	41.4s	0.447	17.8s	**0.425**	3.7s
S-VGGT *ICME'26*	OOM		0.439	26.1s	**0.444**	14.3s	0.427	3.2s
RegimeVGGT	0.465	**71.5s**	**0.452**	**25.6s**	0.449	**12.2s**	0.430	**3.0s**

4.3 Camera Pose Estimation

We next evaluate whether RegimeVGGT preserves the cross-frame geometry needed by the camera head. Camera pose estimation directly tests whether compression keeps the global correspondence structure that supports camera recovery. As a qualitative example, Figure 3 shows the 1000-frame trajectory on ScanNet-50 `scene0648_01`; RegimeVGGT maintains the global loop structure, consistent with the quantitative long-sequence pose results in Table 5.

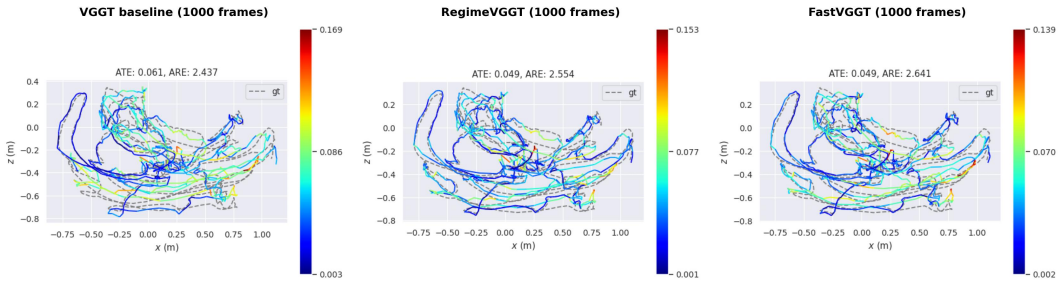


Figure 3: **Predicted camera trajectories on ScanNet-50** `scene0648_01` **under 1000-frame inference.** VGGT baseline (left), RegimeVGGT (center), and FastVGGT (right) compared against ground truth (gray); predictions colored by per-frame ATE.

Large-scale outdoor pose. Table 3 reports camera pose estimation on the six Tanks & Temples training scenes using per-scene AUC@30, mean AUC@30, and total wall-clock time. RegimeVGGT matches FastVGGT in mean AUC@30 (90.9) while reducing total runtime from 242.2s to 111s. It also improves over S-VGGT in both mean AUC@30 and runtime. Compared with the full-token VGGT* baseline, RegimeVGGT preserves a close average AUC@30 while requiring only about one fifth of the runtime. These results show that the compressed global-attention computation still maintains scene-level camera consistency on large outdoor sequences with broad viewpoint changes.

Object-level pose. Table 4 reports camera pose estimation on DTU using AUC at 30°, 15°, and 5°. The VGGT-family methods are tightly clustered on this benchmark, suggesting that object-level pose estimation is largely saturated under this protocol. RegimeVGGT matches the best AUC@30 with 94.5, matches VGGT* at AUC@15 with 89.0, and remains close to the best AUC@5 result with

Table 3: Camera pose estimation on Tanks & Temples over the six training scenes. We report per-scene AUC@30, mean AUC@30 across scenes, and total wall-clock inference time. Best results among VGGT-based methods are in **bold**.

Method	Barn	Caterp.	Courth.	Ignat.	Meeting.	Truck	Avg.↑	Time↓
VGGT* *CVPR'25*	94.4	92.5	83.8	92.5	92.8	93.7	**91.6**	552.8s
FastVGGT *ICLR'26*	93.0	91.3	82.5	93.5	91.6	93.4	90.9	242.2s
S-VGGT *ICME'26*	92.3	91.5	82.0	91.0	90.0	90.0	89.5	148.2s
RegimeVGGT	93.1	91.5	81.2	93.0	92.6	93.9	90.9	**111s**

Table 4: Camera pose estimation on DTU using 22 standard test scans with 48 frames per object. Best results are in **bold**.

Method	AUC@30↑	AUC@15↑	AUC@5↑
VGGT* *CVPR'25*	94.5	89.0	67.6
FastVGGT *ICLR'26*	94.5	**89.1**	**67.7**
S-VGGT *ICME'26*	94.3	88.6	66.5
RegimeVGGT	**94.5**	89.0	67.4

67.4. These results indicate that RegimeVGGT preserves the camera-relevant cross-view alignment on object-level multi-view sequences, even though it changes the global-attention computation at inference time.

Long-sequence pose on ScanNet-50. Table 5 evaluates camera pose estimation on ScanNet-50 with 100, 300, 500, and 1000 input frames. This setting tests whether RegimeVGGT can preserve camera consistency as the number of views grows. At 1000 frames, RegimeVGGT achieves the best results across all four pose metrics, while S-VGGT runs out of memory. At 500 frames, RegimeVGGT also obtains the best ARE, RPE-rot, and RPE-trans among the reported methods. At 300 and 100 frames, its errors remain close to VGGT* and FastVGGT, showing that the proposed compression preserves camera-relevant alignment across both short and long input sequences. The strongest gains appear in the longest setting, where global-attention redundancy is most pronounced.

Table 5: Camera pose estimation on ScanNet-50 at varying sequence lengths. We report ATE↓, ARE↓, RPE-rot↓, RPE-trans↓, and Time↓. *OOM* denotes out-of-memory. Best results among VGGT-based methods are in **bold**.

	1000 frames					500 frames					300 frames					100 frames				
Method	ATE↓	ARE↓	R-rot↓	R-tr↓	Time↓	ATE↓	ARE↓	R-rot↓	R-tr↓	Time↓	ATE↓	ARE↓	R-rot↓	R-tr↓	Time↓	ATE↓	ARE↓	R-rot↓	R-tr↓	Time↓
VGGT* *CVPR'25*	0.117	5.127	0.962	0.034	477.3s	0.105	4.645	0.693	0.032	130.2s	0.096	4.206	0.786	0.040	48.7s	**0.092**	3.949	**1.246**	**0.056**	6.8s
FastVGGT *ICLR'26*	0.111	4.166	0.660	0.029	113.5s	**0.093**	**4.016**	**0.642**	**0.031**	41.4s	**0.093**	4.040	**0.776**	**0.036**	17.8s	0.093	**3.839**	1.260	0.060	3.7s
S-VGGT *ICME'26*			OOM			0.113	4.113	0.925	0.091	26.1s	0.137	4.287	1.307	0.122	14.3s	0.223	5.838	3.869	0.258	3.2s
RegimeVGGT	**0.092**	**3.97**	**0.60**	**0.025**	**71.5s**	0.096	4.06	0.68	0.033	**25.6s**	0.095	**4.03**	0.78	0.038	**12.2s**	0.098	3.98	1.30	0.064	**3.0s**

4.4 Ablation Studies

Table 6 ablates the two components of RegimeVGGT on Tanks & Temples (Courthouse separated from the five shorter scenes). Within Saliency-Guided Banded Merging, the band-aligned cache is purely efficiency, removing it leaves AUC@30 unchanged but cuts speedup from $2.78\times$ to $1.89\times$, while removing the size-bias term lowers AUC@30, especially on Courthouse. Within Selectively Protected K/V Downsampling, the reference-frame anchor and phase-shifted grid raise accuracy at unchanged runtime (removing them drops mean AUC@30 from 0.9185 to 0.9126 and 0.9132). Composing the two reaches $5.01\times$ mean and $5.79\times$ Courthouse speedup—our default RegimeVGGT.

5 Memory Efficiency

Table 7 reports peak GPU memory across the full evaluation suite, complementing the headline VRAM comparison in the main paper. VGGT exceeds 80 GB (OOM) on every tested sequence length; VGGT* avoids OOM via activation checkpointing but still pays the full $O(S^2L^2)$ attention cost. FastVGGT, S-VGGT cut runtime but raise peak memory through merge indices and auxiliary

Table 6: Component ablations on Tanks & Temples, split by sequence length: *Long* = Courthouse alone (1106 frames), *Short* = the other five training scenes (251–410 frames each), and *Mean* = all six scenes. Speedup is wall-clock relative to VGGT* on the corresponding split. Higher AUC@30 and higher speedup are better.

		Long (>1k frames)		**Short**		**Mean**	
Config	**Variant**	AUC@30↑	Sp.↑	AUC@30↑	Sp.↑	AUC@30↑	Sp.↑
VGGT* (baseline)		0.8381	1.00×	0.9319	1.00×	0.9162	1.00×
Saliency-Guided Banded Merging	without cache	**0.8409**	1.83×	0.9314	2.02×	0.9163	1.89×
	without size bias	0.8225	2.90×	0.9311	2.57×	0.9064	2.79×
	standard	0.8358	2.90×	0.9320	2.57×	0.9159	2.78×
Selectively Protected K/V Downsampling	without anchor	0.8134	2.27×	0.9324	2.01×	0.9126	2.18×
	without phase shift	0.8254	2.28×	0.9307	2.01×	0.9132	2.18×
	standard	0.8329	2.28×	**0.9356**	2.01×	**0.9185**	2.18×
RegimeVGGT	standard	0.8125	**5.79**×	0.9281	**3.93**×	0.9088	**5.01**×

buffers. RegimeVGGT's K/V downsampling derives its keeper index from the frame index alone and adds no per-layer state.

Table 7: Peak GPU memory across the full evaluation suite. 7 Scenes, NRGBD, and ScanNet-50; mean and max across all scenes per dataset (GB).

	Stride 3				**Stride 10**				**ScanNet-50**							
	7 Scenes		NRGBD		7 Scenes		NRGBD		1000		500		300		100	
Method	Mean	Max	Mean	Max	Mean	Max	Mean	Max	Mean	Max	Mean	Max	Mean	Max	Mean	Max
VGGT	OOM				OOM				OOM							
VGGT*	24.4	26.3	30.0	38.6	8.9	9.5	10.8	13.2	42.5	45.3	23.7	23.7	15.1	15.1	6.5	6.5
FastVGGT	30.1	32.5	37.2	47.9	10.6	11.3	12.9	16.0	60.3	63.7	32.9	32.9	20.6	20.6	8.3	8.3
S-VGGT	33.2	35.5	41.1	52.1	13.3	16.1	17.3	20.2	OOM		37.0	37.9	28.4	29.1	20.5	21.0
RegimeVGGT	27.1	29.8	**33.1**	**41.9**	9.3	10.3	**11.5**	**14.2**	**49.0**	**51.8**	**26.9**	**27.0**	**17.1**	**17.1**	**7.2**	**7.2**

6 Limitations

RegimeVGGT uses fixed per-band compression rates (ρ_b, σ_b) chosen from scene-pooled diagnostics rather than adapting them per input. Since scene complexity varies, input-adaptive rate selection could further improve the speed–quality trade-off without changing the overall design. Second, our protected token set uses frozen DINOv2 `[CLS]` saliency, which is the strongest training-free proxy among the tested importance scores. If fine-tuning or additional supervision is allowed, a learned geometry-aware selector could identify geometry-critical tokens more directly and may further improve preservation under aggressive compression.

7 Conclusion

RegimeVGGT rests on a single observation: VGGT's global attention is not uniformly redundant but *heterogeneously* so. Spectral, probing, and causal diagnostics reveal three functionally distinct regimes: shallow layers lack cross-view structure, middle layers drive cross-view alignment, and deep layers are redundant for dense geometry yet their cross-frame attention remains essential for pose. More broadly, this pose path cuts across all three bands: pose lives in the cross-frame attention between camera/register tokens and the surrounding patches, so patches must remain in this path throughout, even where the depth-wise diagnostics would otherwise mark them as compressible. This dual structure translates directly into layer-wise U-shaped compression along two axes within every global block: *Saliency-Guided Banded Merging* protects geometry- and edge-salient tokens on the token-count axis, while *Selectively Protected K/V Downsampling* preserves cross-frame spatial coverage and the pose-critical path through a phase-shifted spatial grid, a reference-frame anchor, and uncompressed camera/register tokens. Composed within every global block, the two axes deliver a $6.7\times$ end-to-end speedup over VGGT* on ScanNet-1000 without compromising dense reconstruction quality.

Acknowledgments and Disclosure of Funding

The authors received no third-party funding or support for this work in the 36 months prior to submission. The authors also declare no financial relationships outside the submitted work during this period that could be perceived as influencing it.

References

[1] Sameer Agarwal, Yasutaka Furukawa, Noah Snavely, Ian Simon, Brian Curless, Steven M Seitz, and Richard Szeliski. Building rome in a day. *Communications of the ACM*, 54(10):105–112, 2011.

[2] Dejan Azinović, Ricardo Martin-Brualla, Dan B Goldman, Matthias Nießner, and Justus Thies. Neural rgb-d surface reconstruction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6290–6301, 2022.

[3] Daniel Bolya and Judy Hoffman. Token merging for fast stable diffusion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4599–4603, 2023.

[4] 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 (ICLR)*, 2023.

[5] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11621–11631, 2020.

[6] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5828–5839, 2017.

[7] Tri Dao. Flashattention-2: Faster attention with better parallelism and work partitioning. *arXiv preprint arXiv:2307.08691*, 2023.

[8] Timothée Darcet, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. Vision transformers need registers. *arXiv preprint arXiv:2309.16588*, 2023.

[9] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The international journal of robotics research*, 32(11):1231–1237, 2013.

[10] Joakim Bruslund Haurum, Sergio Escalera, Graham W Taylor, and Thomas B Moeslund. Which tokens to use? investigating token reduction in vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 773–783, 2023.

[11] Shahram Izadi, David Kim, Otmar Hilliges, David Molyneaux, Richard Newcombe, Pushmeet Kohli, Jamie Shotton, Steve Hodges, Dustin Freeman, Andrew Davison, et al. Kinectfusion: real-time 3d reconstruction and interaction using a moving depth camera. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, pages 559–568, 2011.

[12] Rasmus Jensen, Anders Dahl, George Vogiatzis, Engin Tola, and Henrik Aanæs. Large scale multi-view stereopsis evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 406–413, 2014.

[13] Minchul Kim, Shangqian Gao, Yen-Chang Hsu, Yilin Shen, and Hongxia Jin. Token fusion: Bridging the gap between token pruning and token merging. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1383–1392, 2024.

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

[15] Seon-Ho Lee, Jue Wang, Zhikang Zhang, David Fan, and Xinyu Li. Video token merging for long-form video understanding. *arXiv preprint arXiv:2410.23782*, 2024.

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

[17] Xinze Li, Pengxu Chen, Yiyuan Wang, Weifeng Su, and Wentao Cheng. S-vggt: Structure-aware subscene decomposition for scalable 3d foundation models. *arXiv preprint arXiv:2603.17625*, 2026.

[18] Youwei Liang, Chongjian Ge, Zhan Tong, Yibing Song, Jue Wang, and Pengtao Xie. Not all patches are what you need: Expediting vision transformers via token reorganizations. *arXiv preprint arXiv:2202.07800*, 2022.

[19] Raul Mur-Artal and Juan D Tardós. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE transactions on robotics*, 33(5):1255–1262, 2017.

[20] 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.

[21] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 12179–12188, October 2021.

[22] Yongming Rao, Wenliang Zhao, Benlin Liu, Jiwen Lu, Jie Zhou, and Cho-Jui Hsieh. Dynamicvit: Efficient vision transformers with dynamic token sparsification. *Advances in neural information processing systems*, 34:13937–13949, 2021.

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

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

[25] You Shen, Zhipeng Zhang, Yansong Qu, Xiawu Zheng, Jiayi Ji, Shengchuan Zhang, and Liujuan Cao. Fastvggt: Training-free acceleration of visual geometry transformer. *arXiv preprint arXiv:2509.02560*, 2025.

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

[27] 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. *arXiv preprint arXiv:2512.04939*, 2025.

[28] Xianbing Sun, Zhikai Zhu, Zhengyu Lou, Bo Yang, Jinyang Tang, Liqing Zhang, He Wang, and Jianfu Zhang. Avggt: Rethinking global attention for accelerating vggt. *arXiv preprint arXiv:2512.02541*, 2025.

[29] Sebastian Thrun. Probabilistic robotics. *Communications of the ACM*, 45(3):52–57, 2002.

[30] Federico Tombari, Alessandro Franchi, and Luigi Di Stefano. Bold features to detect texture-less objects. In *Proceedings of the IEEE international conference on computer vision*, pages 1265–1272, 2013.

[31] Chung-Shien Brian Wang, Christian Schmidt, Jens Piekenbrinck, and Bastian Leibe. Faster vggt with block-sparse global attention. *arXiv preprint arXiv:2509.07120*, 2025.

[32] Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 5294–5306, 2025.

[33] Qianqian Wang, Yifei Zhang, Aleksander Holynski, Alexei A Efros, and Angjoo Kanazawa. Continuous 3d perception model with persistent state. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 10510–10522, 2025.

[34] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 20697–20709, 2024.

[35] Weitian Wang, Lukas Meiner, Rai Shubham, Cecilia De La Parra, and Akash Kumar. Httm: Head-wise temporal token merging for faster vggt. *arXiv preprint arXiv:2511.21317*, 2025.

[36] Yifan Wang, Jianjun Zhou, Haoyi Zhu, Wenzheng Chang, Yang Zhou, Zizun Li, Junyi Chen, Jiangmiao Pang, Chunhua Shen, and Tong He. π^3: Scalable permutation-equivariant visual geometry learning. *arXiv e-prints*, pages arXiv–2507, 2025.

[37] Yikai Wang, Xinghao Chen, Lele Cao, Wenbing Huang, Fuchun Sun, and Yunhe Wang. Multimodal token fusion for vision transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12186–12195, 2022.

[38] Zhenhailong Wang, Senthil Purushwalkam, Caiming Xiong, Silvio Savarese, Heng Ji, and Ran Xu. Dymu: Dynamic merging and virtual unmerging for efficient vlms. *arXiv preprint arXiv:2504.17040*, 2025.

[39] 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 *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 21924–21935, 2025.

[40] Yao Yao, Zixin Luo, Shiwei Li, Tian Fang, and Long Quan. MVSNet: Depth inference for unstructured multi-view stereo. In *European Conference on Computer Vision (ECCV)*, pages 767–783, 2018.

[41] Jintao Zhang, Chendong Xiang, Haofeng Huang, Jia Wei, Haocheng Xi, Jun Zhu, and Jianfei Chen. Spargeattention: Accurate and training-free sparse attention accelerating any model inference. *arXiv preprint arXiv:2502.18137*, 2025.

[42] Dong Zhuo, Wenzhao Zheng, Jiahe Guo, Yuqi Wu, Jie Zhou, and Jiwen Lu. Streaming 4d visual geometry transformer. *arXiv preprint arXiv:2507.11539*, 2025.

A Extended Analysis Evidence

This appendix collects the empirical evidence underlying RegimeVGGT's two design axes—token merging and K/V downsampling—together with the experiments that motivated each design choice. Both axes rest on a layer-wise partition of VGGT's 24 aggregator layers into three functionally distinct bands (shallow, middle, deep), so we first establish that partition with three independent diagnostics—probing, causal intervention, and rank spectrum analysis (A.1). We then present per-axis evidence. For token merging: a geometry-token classification that defines the protection target under merging, followed by a four-way importance ablation that selects DINOv2 saliency as the deliberate scoring rule (A.2). For K/V downsampling: the empirical path from a deep-only skip variant through asymmetric queries to the final selectively protected design (universal U-shape rate, phase-shifted per-frame grid, frame-0 anchor) (A.3).

A.1 Three-Band Layer Partition of VGGT's Aggregator

Section 3.1 of the main paper rests on a partition of VGGT's 24 aggregator layers into three functionally distinct bands—shallow (L1–L10), middle (L11–L18), and deep (L19–L24). Figure 4 already exposes this partition visually: the cross-frame attention heatmaps and matrix submatrices change qualitatively across layers—diffuse and band-structured in the shallow layers (L1, 3, 5), localized with a clear near-diagonal structure in the middle layers (L13, 15), collapsed or weakly structured in the deep layers (L21, 23). To verify that this visual impression reflects an intrinsic three-band structure rather than a single layer's idiosyncrasy, we triangulate the partition with three independent diagnostics. *Probing* (A.1.1) measures how much geometric information is linearly recoverable from the representations at each layer; *causal intervention* (A.1.2) measures whether that information is actually utilized by the downstream prediction heads; *rank spectrum analysis* (A.1.3) inspects the structure of the attention map itself, free of any target.

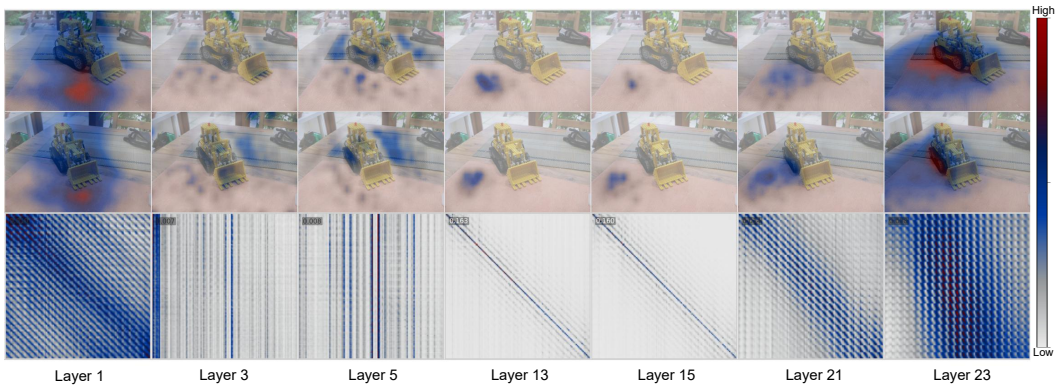


Figure 4: **Three-band layer structure in VGGT's cross-frame attention.** Top: for a fixed query patch in Frame 0, attention to all patch tokens in Frame 1 and Frame 2, visualized as spatial heatmaps. Bottom: cross-frame attention submatrix from all patch tokens in Frame 0 to all patch tokens in Frame 1.

A.1.1 Layer-wise Probe of Geometric Information

Closed-form ridge probe. At each global-attention layer ℓ, we ask how much geometric information the layer's representations carry beyond low-level 2D image cues. The probe script fits closed-form ridge regressors with cross-validated α from frozen VGGT features to four targets: depth, surface normal, camera-frame pointmap, and camera pose. The three dense targets are regressed from patch tokens, while pose is regressed from the per-frame camera token; Figure 5 reports only the three dense patch-token probes. For the dense targets, we also fit the same ridge probe from a 6-dim control feature $\mathbf{Z} = (\text{row}, \text{col}, R, G, B, \text{Sobel})$ computed per patch, and report the control-adjusted coefficient of determination

$$R^2_{\text{resid}}(\ell) \;=\; R^2_{\text{full}}(\ell) - R^2_{\text{ctrl}}, \tag{5}$$

where R^2_{full} uses the VGGT feature and R^2_{ctrl} uses only the control feature. The signed per-layer gain $\Delta(\ell)$ plotted in Figure 5 is a post-processing of this residual trajectory and is normalized by $\max_j |\Delta_j|$ for visual comparability across targets.

Data and robustness. The main probe uses 7 Scenes [5] and NRGBD [19] with stride-10 keyframe sampling and an $80/20$ train/test split on pooled tokens. The dense targets are derived from the saved ground truth: depth is the dataset depth map, surface normal is computed from that depth map, and pointmap is obtained by unprojecting depth with the camera intrinsics. Statistical reliability is checked with a permutation test (n=200) for the dense-target R^2_{full}. The script also saves leave-one-scene-out per-scene R^2 values using the CV-selected α, which we use downstream to assess cross-scene variability. We additionally run a shallow G2F-removal boundary scan over several cutoffs K and feed each re-dumped feature set through the same probe.

Results. Figure 5 reports the signed, max-normalized per-layer gain $\Delta^{(\ell)}/\max_j |\Delta^{(j)}|$ for the three dense per-patch probes: pointmap, depth, and surface normal. The solid curves denote the default configuration used in our final model, in which G2F is removed. The dashed curves denote boundary-scan variants that retain G2F through layers L1–K for several tested cutoffs K. Across the tested cutoffs, retaining shallow G2F yields weaker aggregate dense-geometry probe signals than the default no-G2F configuration, indicating that early G2F suppresses rather than preserves useful dense-geometry accumulation. We therefore remove G2F in the final model. This transition is consistent with Figure 4, where attention changes from diffuse, weakly structured shallow patterns to localized near-diagonal cross-frame structure in the middle layers.

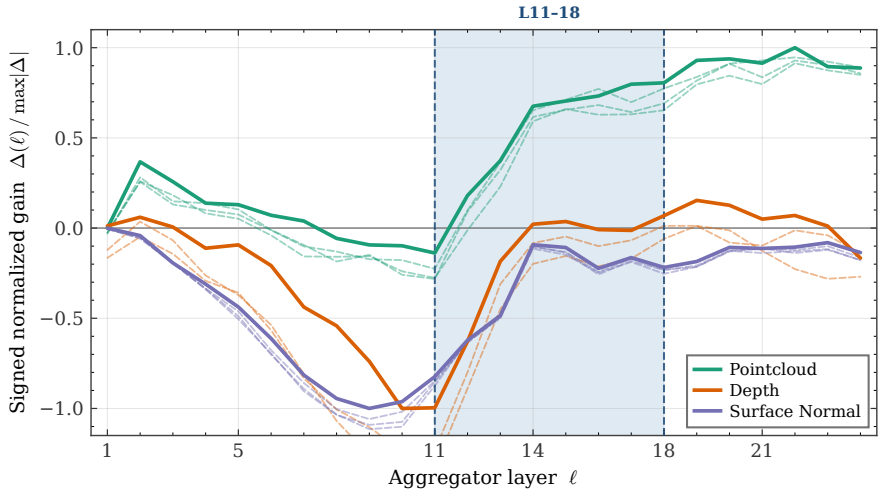


Figure 5: **Geometric probe gain emerges in the middle band L11–L18.** Signed normalized gain $\Delta(\ell)/\max_j |\Delta_j|$ across VGGT's 24 aggregator layers for three dense probes (pointmap, depth, surface normal). Solid curves: default configuration with G2F removed. Dashed curves: boundary-scan variants retaining G2F through L1–K. Shaded region: the *critical* band L11–L18.

A.1.2 Causal and Structural Diagnostics of Cross-View Alignment

From availability to utilization. The probe in A.1.1 establishes that geometric information is linearly *available* in the representations of L11–L18, but does not confirm that the prediction heads actually *use* it. We close this gap with two complementary tests on every global-attention layer: a *causal* test that asks whether perturbing the layer's patch-key correspondence changes the downstream output, and a *structural* test that asks whether the attention pattern itself looks like geometric correspondence matching. Either test in isolation is only suggestive—a layer's attention can look sharp without driving the output, or perturbations may matter for unrelated reasons—but their *convergence* pins a layer as a genuine cross-view alignment site.

Experiment. We compute four per-layer readouts on the same scenes and model configuration, using 7 Scenes [5] capped at 16 frames per sequence to bound the $O(S^2)$ aggregator memory. *(i)* For the causal influence $\mathrm{CI}(\ell)$, we install a forward hook on each layer's **QKV** projection that overwrites the keys $\mathbf{K}$ of all patch tokens with a single random permutation of patch positions across the entire frame stack, leaving $\mathbf{Q}$ and $\mathbf{V}$ unchanged; we then forward the model and record

$$\mathrm{CI}(\ell) \;=\; \frac{\|\mathrm{head_out}_{\mathrm{perm}} - \mathrm{head_out}_{\mathrm{clean}}\|_2}{\|\mathrm{head_out}_{\mathrm{clean}}\|_2} \tag{6}$$

for the depth and camera heads, averaged over $n_{\mathrm{perm}}{=}5$ permutation seeds. *(ii)* For the structural diagnostics, we approximate attention scores from qkv-projected $(\mathbf{Q}, \mathbf{K})$ on $n_{\mathrm{query}}{=}64$ sampled patch queries per frame (flash-attention does not expose softmax weights) and report the normalized entropy $H = -\sum_k a_k \log a_k / \log NL \in [0, 1]$, plotted as $1{-}H$ so that high values mean sharp attention, and the *diagonal ratio*: the fraction of cross-frame attention mass landing within ± 2 patches of each query's spatial position.

Analysis. Figure 6 reports the four metrics, each per-row normalized, across the 24 global-attention layers. The four rows collapse onto the same partition. The shallow band (L1–L10) is uniformly dim: attention is unfocused, unaligned with the cross-frame correspondence diagonal, and key permutation produces no measurable change in either head. The deep band (L19–L24) is also dim across the causal rows and shows little cross-frame diagonal structure, suggesting that these layers are not the main geometric correspondence-building stage. The critical band L11–L18 (red box) is bright across all four rows: attention is sharply concentrated and lies along the cross-frame correspondence diagonal, while key permutation shifts the outputs measurably, with the camera CI peaking at around L15 and the depth CI at around L14. The two structural and two causal signals *co-localize* to L11–L18, supplying the convergence requirement set up above: this is where cross-view geometric attention actually does its work. We interpret the gap between camera and depth CI as partly reflecting head architecture: the camera head concentrates a low-dimensional pose prediction into a single bottleneck token, while depth is per-patch with redundancy that buffers single-layer disruptions.

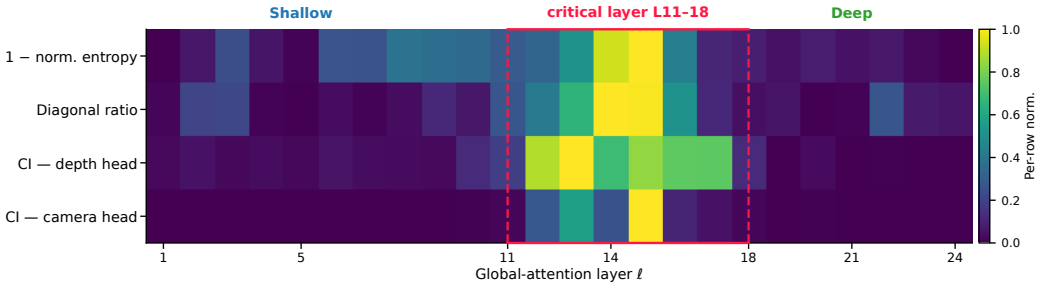


Figure 6: **Four diagnostics co-localize cross-view alignment to L11–L18.** Rows: $1{-}$norm. entropy (sharpness), diagonal ratio (cross-frame correspondence), causal influence on depth head, causal influence on camera head; each row per-row normalized. The critical band L11–L18 (red box) is bright on all four rows; the camera CI peaks near L15.

A.1.3 Rank Spectrum of Global Attention

Target-free structural diagnostic. The probe and causal-intervention results above localize the same middle band from two end-to-end perspectives, but both rely on ground-truth targets or prediction-head outputs. We therefore ask whether the same layer-wise partition is already visible in the global-attention computation itself, and analyze the rank spectrum of each layer's attention probability matrix.

Sampled attention-rank: metric and setup. To probe the layer-wise partition without relying on labels or prediction heads, we examine the rank spectrum of each layer's global-attention probability matrix. The rank reflects the dimensionality of the cross-token interactions a layer actually exercises: a low-rank matrix indicates that all queries attend through a small shared subspace—routing, copying,

or smoothing—whereas a high-rank matrix indicates diverse, query-specific aggregation, the regime in which non-trivial cross-view computation can occur. Since the full token-by-token attention matrix is too large to decompose directly on long sequences, for each layer ℓ and head h we sample m query rows and compute their attention weights over all $N = SP$ tokens, producing an $m \times N$ submatrix. We then compute its singular values and measure the smallest rank required to explain 95% of the spectral energy:

$$r_{95}^{(\ell,h)} = \min\left\{ r : \frac{\sum_{i=1}^{r} \sigma_i^2}{\sum_i \sigma_i^2} \geq 0.95 \right\}.$$

We report the normalized rank ratio $r_{95}^{(\ell,h)}/m$, averaged over heads and scenes. We evaluate the metric on scenes from DTU, ScanNet, and Tanks & Temples, covering object-centric, indoor, and large-scale outdoor settings. Unless otherwise stated, we use $m = 256$ sampled query rows and at most 32 frames per scene, and aggregate $r_{95}^{(\ell,h)}/m$ over all heads at each layer.

Consistent layer boundaries. This target-free spectral diagnostic provides an independent structural confirmation of the three-band partition. The middle band is not only where geometric information is available and causally used; it is also where the global-attention probability map has the highest spectral complexity. This indicates that L11–L18 contains the richest cross-view interaction structure, consistent with its role as the main alignment regime. By contrast, the shallow and deep bands have lower spectral complexity, although for different reasons: the shallow band has not yet developed strong cross-view matching structure, while the deep band has largely collapsed into a redundant refinement regime. Thus, the same L11 and L18 boundaries emerge from representation probing, causal use, and the intrinsic spectrum of global attention itself. Because this spectrum is measured on the attention probability map, the low-rank flanks indicate redundancy in how queries read from the global K/V set, while the rank-heavy middle band requires denser K/V support to preserve its richer cross-view correspondence structure.

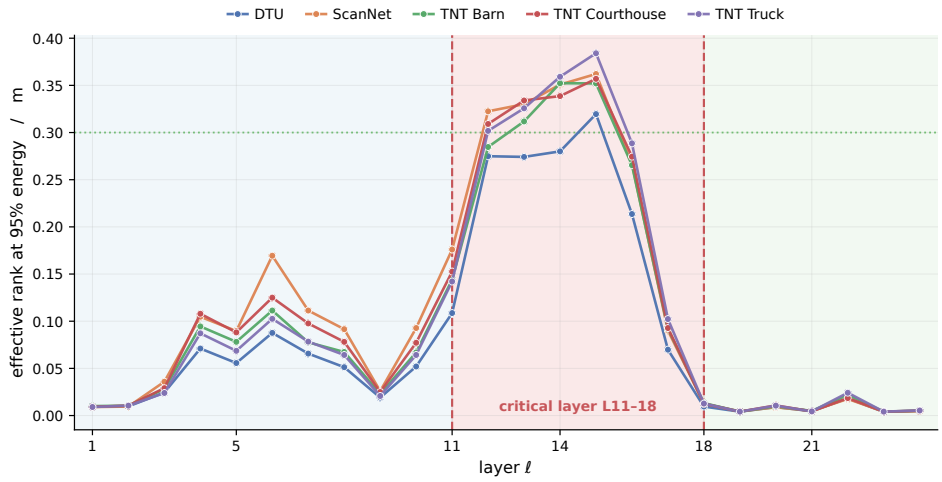


Figure 7: **Effective attention rank is universally inverted-U-shaped, peaking at the middle band L11–L18.** Effective rank of the global-attention matrix vs. layer depth, on five scenes (TNT Courthouse, TNT Barn, TNT Truck, DTU, ScanNet); m=256 sampled query rows per layer. Shallow ($\ell \leq 10$) and deep ($\ell \geq 19$) layers concentrate in the top singular component (eff-rank/m<0.05); the middle band $\ell \in [11, 18]$ peaks above the 0.30 go-criterion (green dotted line).

A.2 Token-Level Heterogeneity and Protection Targets

The layer-wise analyses in Appendix A.1.1–A.1.3 show where VGGT can be compressed across depth, but token merging also requires a second question: *which patch tokens should survive the merge?* This is not a uniform token-reduction problem. In VGGT, each patch token remains spatially tied to dense 3D prediction through the DPT heads, so merging the wrong tokens can erase local structure that cannot be recovered later.

Figure 2 in the main paper motivates this token-level view. It shows that the middle band L11–L18 is not only the regime where cross-view geometric information emerges, but also the regime where a small heterogeneous subset of patch tokens carries disproportionate 3D information. Ablating the top-10% geometry-boundary tokens disrupts the depth head much more than ablating random tokens, while the residual 3D signal accumulates across the same L11–L18 window identified by the probe and causal-intervention analyses. This appendix expands that observation in two steps. Appendix A.2.1 defines the token classes that should be protected during token merging. Appendix A.2.2 then compares practical importance scores for selecting such protected tokens during inference.

A.2.1 Geometry and Edge Tokens as Protection Targets for Token Merging

Motivation. Token merging removes redundancy by combining similar patch tokens, but dense 3D reconstruction is sensitive to local spatial failures. Smooth image regions often contain many redundant tokens and can tolerate merging, whereas tokens near depth changes, object boundaries, or high-frequency spatial structures are more fragile. We therefore separate the protection target into two complementary token types: *geometry tokens*, which denote patch locations whose depth cannot be explained by local 2D cues alone and therefore serve as a proxy for 3D-critical regions, and *edge tokens*, which mark high-frequency image boundaries that are vulnerable to merging.

Geometry tokens. We define geometry tokens through a residual-depth criterion, similar to Appendix A.1. For each patch token, we first regress the ground-truth patch depth from low-level 2D cues: normalized image coordinates, patch RGB values, and Sobel edge magnitude. This control regression explains the portion of depth that can already be predicted from image position, color, and local edge strength. We then rank tokens by the squared residual of this regression,

$$e_i = \left(y_i^{\mathrm{depth}} - \hat{y}_i^{\mathrm{2D}} \right)^2, \tag{7}$$

and define the top-10% residual tokens as geometry tokens. A high residual means that the token's depth cannot be explained by local 2D cues alone, so it is more likely to correspond to genuine 3D structure such as depth discontinuities or scene-layout changes.

Edge tokens. We define edge tokens independently using Sobel magnitude on the input image. These tokens capture high-frequency spatial structures such as object contours, texture boundaries, and possible depth edges. Edge tokens are not equivalent to geometry tokens: some strong image edges are appearance-only texture, while some geometry tokens correspond to 3D structure that is not the strongest 2D edge. The two sets therefore provide complementary protection targets. Geometry tokens target patch locations associated with residual 3D information, while edge tokens target spatial boundaries that are easy to blur under merging. Together, they define the desired protection target for token merging, whereas the practical policy must approximate this target using inference-time signals that do not rely on ground-truth depth.

Functional validation. To verify that geometry tokens are not an artifact of the residual definition, we perform a full layer-wise ablation experiment by applying the same token-ablation procedure across the aggregator layers. For each frame, we ablate the top-10% geometry tokens and compare against ablating the same fraction of randomly selected patch tokens. We measure the relative change in VGGT's DPT depth output outside the ablated regions, so that the metric reflects how perturbing a selected token set propagates to the surrounding dense geometry rather than only the directly masked locations. As reported in Figure 2, geometry-token ablation causes about $2.1\times$ larger depth disruption than random ablation on 3D-rich scenes. On geometrically flatter scenes, such as Flower, the gap narrows to about $1.1\times$, consistent with the smaller number of strong depth discontinuities in such scenes. This confirms that the residual-depth criterion identifies a functional token class whose removal disproportionately affects dense geometry.

Implication for token merging. These observations define the target that token merging should preserve. A safe merge policy should preferentially keep patch tokens associated with residual 3D information and tokens lying on sharp spatial boundaries, while allowing redundant tokens in smooth regions to be merged. The next section evaluates practical inference-time importance scores for approximating this protection target.

A.2.2 Importance Ablation

Given the protection target defined above, we next ask which inference-time per-token score should be used to approximate this target in token merging.

We compare four candidate per-token importance scores Ψ for the protection set used by token merging and the hybrid compression scheme: (i) DINOv2 [CLS] class-attention from the last DINOv2 block, head-averaged; (ii) aggregator frame-attention saliency at L17; (iii) Sobel edge magnitude on the input image; (iv) ℓ_2 norm of the L17 patch token. For each scoring rule, we run the full Saliency-Guided Banded Merging pipeline under the canonical configuration and report mean Acc / NC / Time / VRAM on 7 Scenes and NRGBD at strides 3 and 10 (Table 8).

Table 8: **DINOv2 `[CLS]` attention matches accuracy-best baselines and outperforms them on VRAM.** Importance scoring variants on 7 Scenes (18 scenes) and NRGBD (8 scenes) at stride 3/10. Memory reported as Mean / Max across scenes (GB).

	Stride 3								**Stride 10**							
	7 Scenes				NRGBD				7 Scenes				NRGBD			
Importance	Acc↓	NC↑	Time↓	VRAM↓	Acc↓	NC↑	Time↓	VRAM↓	Acc↓	NC↑	Time↓	VRAM↓	Acc↓	NC↑	Time↓	VRAM↓
`frame_attn`	0.017	**0.603**	14.19	32.7 / 72.4	0.019	0.663	22.73	38.2 / 49.1	0.017	0.611	2.97	10.9 / 11.6	**0.018**	0.670	4.15	13.3 / 16.4
`sobel`	0.017	0.601	13.99	30.6 / 33.3	0.019	0.661	22.55	38.2 / 49.1	0.017	0.609	2.94	10.8 / 11.6	0.021	0.666	4.10	13.1 / 16.4
`token_norm`	0.017	0.600	**13.94**	30.6 / 33.3	0.031	0.659	22.52	38.2 / 49.1	0.017	0.611	2.92	10.8 / 11.6	0.028	0.665	4.10	13.1 / 16.4
`dino_attn` (ours)	**0.017**	0.602	13.96	**30.6 / 33.3**	**0.019**	**0.663**	**22.52**	**38.2 / 49.1**	**0.017**	**0.611**	**2.92**	**10.8 / 11.6**	0.019	**0.673**	**4.10**	**13.1 / 16.4**

The DINOv2 score is computed once per input image and is therefore *input-fixed*: it does not depend on the aggregator's intermediate state, does not interact with the merge schedule, and does not accumulate noise across layers. This distinguishes it from frame-attention and token-norm scores, which drift as earlier merging changes the token representations they read from. Figure 8 further shows DINOv2 attention is *not edge-only*: it lands on salient objects rather than tracing every image edge, so unlike Sobel, which is input-fixed but captures only local 2D edge strength, it approximates the geometry- and edge-sensitive protection target defined above. We therefore use `dino_attn` as the default scoring rule for Ψ.

A.3 Empirical Path to Selectively Protected K/V Downsampling

This appendix documents the empirical path that produced our final K/V downsampling design and the failure modes that motivated each step along the way. The final configuration (Appendix A.3.3) keeps the camera and register tokens at full K/V density to preserve the cross-frame pose path, and partially updates the remaining patch tokens through phase-shifted K/V downsampling applied universally across the global stack, with a frame-0 anchor preserved at full density and the per-band rate set by the rank budget reported in A.1.3. The configuration was reached only after two more aggressive deep-band-only variants—direct skip of the deep global attention (Appendix A.3.1) and asymmetric queries restricted to the deep band (Appendix A.3.2)—each broke pose in a distinct way; we detail the variants and their failure modes below, with quantitative comparison in Table 10.

A.3.1 Step 1 — Skipping the deep global attention

Our first instinct was the simplest one consistent with the analysis: if deep global attention is near-rank-1 and contributes little to dense reconstruction, just skip it. We replaced the global block at L19–L24 with a no-op, so each frame in the deep band is processed in isolation through the existing frame-attention path, and left every other layer (including all of L1–L18) unchanged. The reconstruction numbers held—confirming that deep global attention is not where dense outputs live—but the camera-pose pipeline collapsed on long sequences (Table 10). Without deep-band global updates, the camera token at the top of the aggregator stops receiving fresh cross-frame information in those final six layers, and the resulting pose drift accumulates with sequence length until the trajectory becomes unusable. Deep global attention, despite contributing little to reconstruction, carries the cross-frame signal that lets the camera token resolve drift; we cannot remove that path wholesale.

A.3.2 Step 2 — Asymmetric queries inside the deep band

Step 1 told us that some cross-frame computation must remain in the deep band; the next question was which tokens still need to be updated. VGGT's camera head reads from the camera token, and the pose

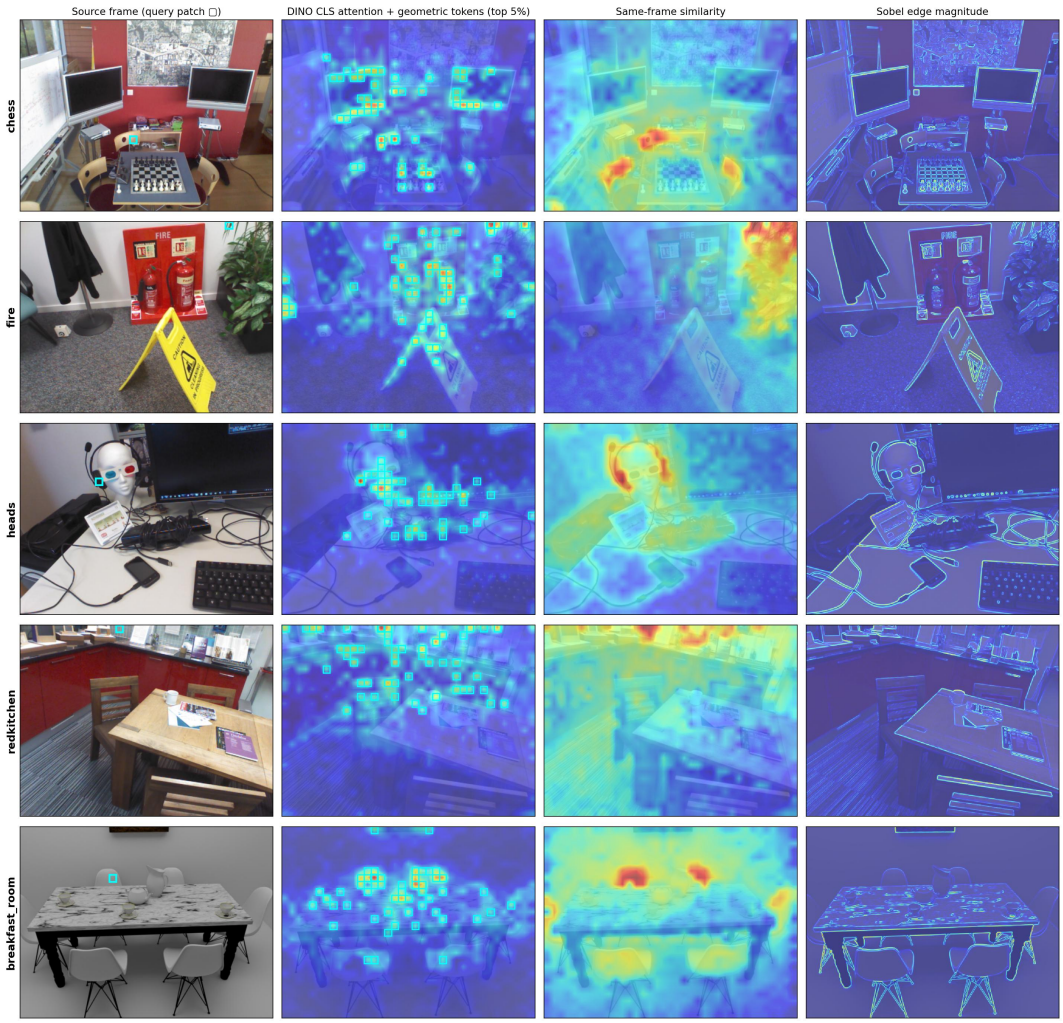


Figure 8: **DINOv2 [CLS] attention is edge-aware but not edge-only, motivating its use as the protection signal in RegimeVGGT.** Across five scenes (rows): **(col 1)** source frame with the top-1 CLS-attention patch outlined in cyan; **(col 2)** DINO CLS attention map from the last patch-embed block, with the top-5% patches (the geometry tokens we protect) outlined in cyan; **(col 3)** cosine similarity between the query patch and every other patch in the same frame; **(col 4)** Sobel edge magnitude.

probe in Appendix A.1.1 is regressed directly from that token, with register tokens accompanying it across frames. We therefore tried a strict asymmetric variant restricted to L19–L24: only camera and register tokens issue cross-frame queries, while patch tokens bypass the deep global-attention residual and MLP. Patches still update through within-frame attention and retain their global updates in L1–L18.

The important failure is that *recognizable point-cloud structure can coexist with severe camera-pose degradation*. Appendix D illustrates this intervention applied on top of RegimeVGGT on ScanNet. Although the deep-asymmetric reconstruction retains the overall scene layout, its predicted trajectory exhibits much larger deviations from ground truth (Figures 10 and 11). At 500 frames in this case study, ATE increases from 0.251 to 1.119 and ARE from 6.192 to 31.966 relative to RegimeVGGT. Thus, retaining camera/register queries alone does not preserve pose accuracy, even when the reconstructed scene still looks broadly plausible.

A plausible explanation is that the patch representations formed in earlier layers suffice to retain coarse scene structure, but freezing their deep global updates leaves the camera/register queries reading patch evidence that no longer follows the evolving cross-frame consensus. This interpretation is consistent with the observed pose failure; the visualizations do not by themselves establish that mechanism. The result motivates Step 3: retain global updates for patch queries while reducing their K/V context selectively, rather than treating deep-layer redundancy for dense geometry as a reason to freeze patch updates entirely.

A.3.3 Step 3 — Selectively Protected K/V Downsampling

Step 2's failure pinpointed the smallest fix that could rescue it: patches need to update through global attention, but their full K/V is not necessary—replacing it with a downsampled K/V leaves all attention paths intact (cam/reg still query, patches still query, patches still get updated) while reducing the cross-frame compute. The final design factors into three complementary mechanisms described next.

Downsampling. The mechanism, once written down, has no built-in dependence on the layer it lives in: K/V downsampling at deep is the same operation as K/V downsampling at middle or shallow, only with a different rate. Restricting it to L19–L24 would leave free compression on the table at every other layer, where the rank measurements in A.1.3 say there is also slack to take. We therefore apply the same K/V downsampling method at every layer, with the per-band rate set by the rank profile; the specific σ_b values and per-band kept fractions are given in Section 3.3. At every layer all tokens participate as queries—including patches, so the staleness mechanism behind Step 2 disappears. The K/V side is selectively protected: camera and register tokens, together with the geometry and edge tokens identified by Ψ, are kept at full density.

Phase shift. Within a single frame, downsampling at rate σ keeps patch K/V on a regular sub-grid whose kept fraction is approximately $1/\sigma^2$, instantiated on a discrete tile (t, k) with $k/t \approx 1/\sigma$ so that $(k/t)^2$ matches the continuous target. If every frame retained the same sub-grid, the cross-frame K/V union would inherit the same spatial blind spots, and queries arriving at any frame would systematically miss the suppressed coordinates. We avoid this by *phase-shifting* the grid as a function of the frame index: at frame f with tile size t (set by σ), the row and column offsets are $(f \operatorname{div} t) \bmod t$ and $f \bmod t$, so different frames retain different sub-grids. The union of S frames' kept positions therefore approaches full spatial coverage as S grows, at no extra per-frame cost. Phase-shifting is the mechanism that lets us downsample K/V aggressively per frame while keeping the cross-frame information set nearly intact.

Anchor frame. On top of phase-shifted per-frame downsampling, we keep a single anchor frame at full K/V density, fixed at frame 0. Its role is complementary to phase-shifting: phase-shifting guarantees that the cross-frame K/V union approaches full spatial coverage as S grows, but any single layer still has each query reading only $\approx 1/\sigma^2$ of every non-anchor frame. The anchor frame supplies a stable, fully resolved world-coordinate reference that every query at every layer can resolve against, which is precisely what the middle-band cross-view alignment regresses toward. Table 9 compares four anchor-frame configurations on Tanks & Temples 6-scene pose at the canonical σ_b=1.5 density. The single-anchor configuration at frame 0 matches or beats every alternative on AUC@30 at the lowest K/V budget. Two anchors do not improve accuracy because frame 0 already supplies the world reference the middle band aligns against; the second anchor only adds K/V tokens without adding alignment information. Rotating the anchor across layers *actively hurts* accuracy: the middle-band alignment computation regresses toward frame 0's coordinate frame, and rotating the full-density frame phase-shifts that target through layers, which is exactly the inconsistency the construction was designed to avoid.

Putting these pieces together, the final K/V downsampling configuration applies a phase-shifted per-frame sub-grid at every layer, with cam/reg tokens, the frame-0 anchor, and the Ψ-selected patches kept at full density. Combined with *Saliency-Guided Banded Merging* (Section 3.2), this configuration produces the $6.7\times$ speedup over VGGT* on ScanNet-1000 reported in the main paper. Table 10 quantifies the cost of each shortcut on Tanks & Temples pose: skipping deep global wholesale (Step 1) collapses AUC@30 from 0.9162 to 0.6330, the deep-band asymmetric variant (Step 2) only recovers to 0.7965, while selective K/V protection (Step 3) matches the baseline at 0.9185 with a $2.18\times$ speedup.

Table 9: **Single anchor at frame 0 outperforms other anchor configurations.** Anchor variants on Tanks & Temples 6-scene pose, with a uniform σ_b=1.5 across bands and variants to isolate the anchor configuration (the U-shape rates of the final design are not used here).

Variant	AUC@30↑	Speedup
no anchor (phase-shift only)	0.913	2.05×
two anchors (frames 0, $S/2$)	0.918	1.99×
rotating anchor (frame $\ell \bmod S$)	0.911	2.06×
single anchor at frame 0 (ours)	**0.919**	2.09×

Table 10: **Step 3 outperforms the deep-band-only variants on Tanks & Temples pose.** Row 1 is the VGGT* baseline; rows 2–4 are the three variants discussed in A.3.1–A.3.3. Mean AUC@30 across the 6 training scenes (kf_every=1, same protocol as Table 3); total wall-clock; speedup over VGGT*. Step 1 collapses pose (-0.28 AUC@30); Step 2 only partially recovers (-0.12); Step 3 matches the baseline at 2.18×.

Variant	Cross-frame attention spec	AUC@30 ↑	Time ↓	Speed ↑	Status
VGGT* (baseline)	global block unchanged at every layer	0.9162	556.1s	1.00×	ref.
Step 1: skip deep global	global block → ∅ in the deep band (L19–L24); L1–L18 unchanged	0.6330	169.4s	3.28×	FAIL
Step 2: deep-band asym.	in the deep band: cam/reg Q only, patches frozen in deep global; L1–L18 unchanged	0.7965	179.2s	3.10×	PARTIAL
Step 3: selectively protected *(ours)*	all Q at every layer; K/V phase-shift σ_b=1.5 across bands; cam/reg/frame-0/Ψ-top-α kept full	**0.9185**	**254.6s**	**2.18×**	**PASS**

B Complexity Derivation

In the original VGGT, every layer runs both frame and global attention. With 24 frame blocks and 24 global blocks (alternating schedule), the total cost is $\mathcal{O}_{\text{VGGT}} = 24 \cdot O(SL^2) + 24 \cdot O(S^2L^2)$, dominated by the $O(S^2L^2)$ term for large S.

Token merging. At each global block in band b, the merge ratio ρ_b retains $(1-\rho_b)\,T$ patch tokens per non-reference frame; the resulting per-block global-attention cost is $O\big(S^2 L_b'^2\big)$ with $L_b' \approx (1-\rho_b)\,T + 5$. With the canonical ratios ρ_s=0.99, ρ_m=0.5, ρ_d=0.99 and the per-band layer counts (10 shallow, 8 middle, 5 deep, 1 DPT-tap at L24), the global term drops by roughly 7×, translating to 2.75× end-to-end speedup on Tanks & Temples (S=20).

K/V downsampling. The query set is unchanged; per band, keys and values are restricted to a $1/\sigma_b^2$ fraction of each non-anchor frame, plus the full anchor frame. The per-block global-attention cost is $O\big(S^2L^2/\sigma_b^2\big) + O\big(SL^2\big)$ for the anchor contribution. At a uniform σ_b=1.5 across bands plus the frame-0 anchor (the configuration used for the anchor ablation in Table 9), this retains $\approx 49\%$ of the original K/V tokens and yields 2.09× end-to-end speedup on Tanks & Temples (S=20).

Both axes (RegimeVGGT). The two compressions multiply. The full RegimeVGGT block evaluates the phase-shifted K/V index set on the merged token grid, so the per-block cost is $O\big(S^2 L_b'^2/\sigma_b^2\big)$ in band b. With the canonical token-merge ratios ρ_s=0.99, ρ_m=0.5, ρ_d=0.99 and the U-shape K/V rates σ_s=1.5, σ_m=1.3, σ_d=1.7 (A.3.3) plus the frame-0 anchor, the global term drops by roughly 20×, translating to the 6.7× end-to-end speedup over VGGT* on ScanNet-1000 reported in the main paper.

C Comparison with AVGGT

AVGGT [28] is the closest concurrent method to ours: it identifies the same three-regime layer partition, converts early global layers to frame attention (t_{early}=9, compared to our L1–L10), and accelerates the remaining layers via K/V subsampling on a uniform spatial grid. Because AVGGT's code and configurations have not been publicly released, and the two papers use different hardware and evaluation baselines, a direct numerical comparison would be misleading. We instead compare along three hardware-independent dimensions.

Theoretical compression comparison. Both methods' speedups derive from reducing the global-attention cost $O(S^2L^2)$. AVGGT at subsampling factor σ=4 reduces the K/V set to $\approx 1/4$ of patch tokens per frame, yielding a $\sim$4× reduction in global-attention FLOPs; at σ=9, this rises

to $\sim 9\times$. Combined with G2F conversion of the first 9 layers, the total global-attention FLOPs reduction is approximately 10–$15\times$ (estimated from their reported layer counts and subsampling rates). RegimeVGGT's dual-axis compression reduces global-attention FLOPs by $\sim 20\times$ (Appendix B), achieving comparable or higher theoretical compression through the multiplicative composition of the two axes rather than a single aggressive subsampling factor. A higher single-axis subsampling factor ($\sigma \geq 6$) risks removing spatially coherent K/V coverage; the dual-axis design distributes compression across two complementary dimensions to mitigate this.

Coverage of evaluation scenarios. AVGGT reports reconstruction quality on 7 Scenes, NRGBD, and DTU, but does not evaluate long-sequence pose estimation (ScanNet-50 at 1000 frames). Our results in Table 2 show that preserving the cross-frame pose path—via uncompressed cam/reg tokens and the frame-0 anchor—is critical at this scale: RegimeVGGT achieves the best ATE (0.092) among all methods at 1000 frames, substantially improving over VGGT* (0.117) and FastVGGT (0.111). Whether AVGGT's uniform-grid K/V subsampling, which does not explicitly protect the cam/reg cross-frame path, can maintain pose consistency at this sequence length remains an open question that cannot be resolved without their released code.

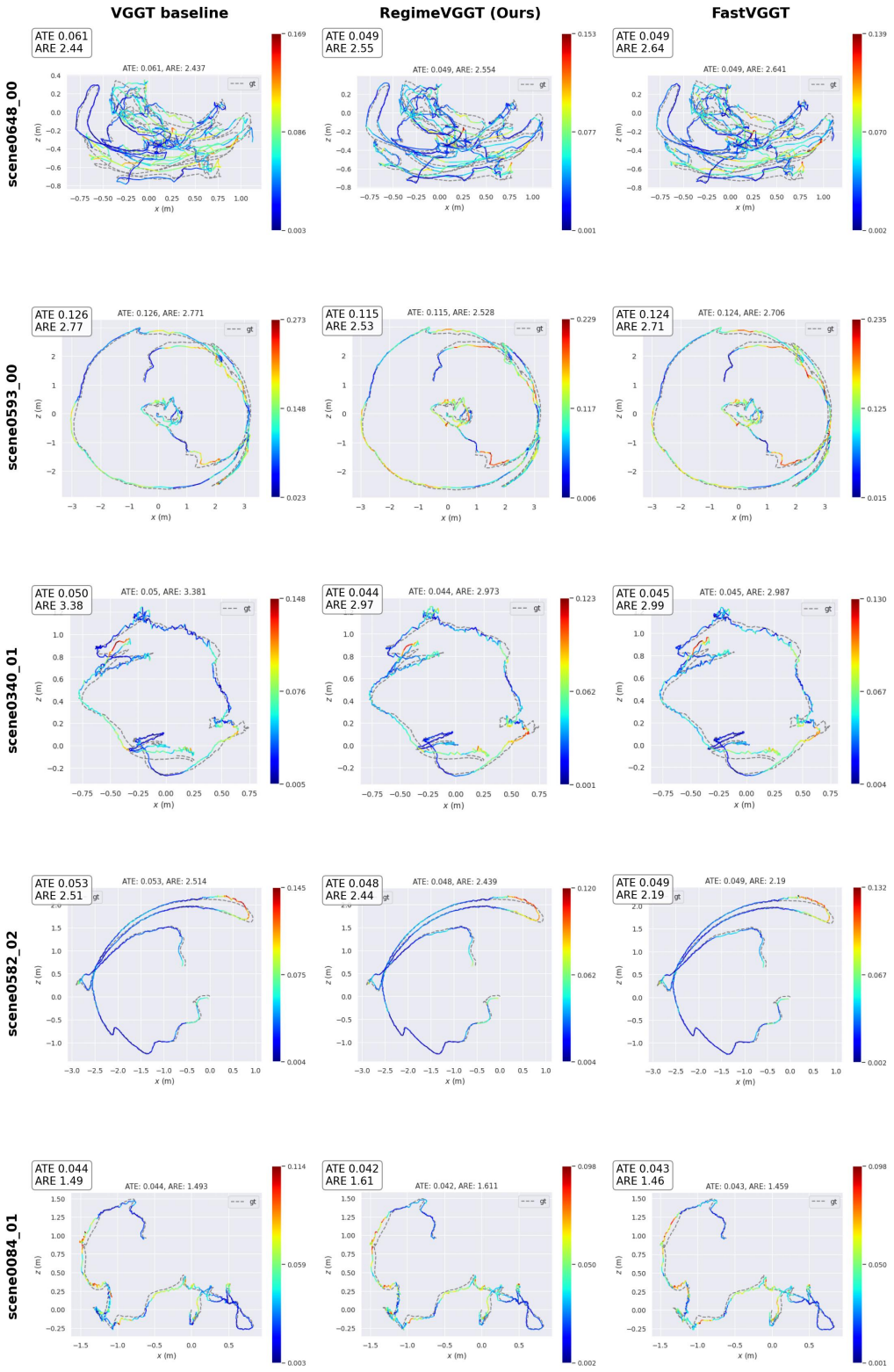


Figure 9: Additional visualizations of pose estimation results on the ScanNet dataset.

D Visualizing Deep-Band Asymmetric Updates

These visualizations extend the failure analysis of *Asymmetric queries inside the deep band* in Appendix A.3.2. We compare the VGGT baseline, RegimeVGGT, and RegimeVGGT with the deep-asymmetric intervention at 100, 300, 500, and 1000 input frames. In the latter setting, camera/register queries remain active in L19–L24, while patch tokens bypass the deep global-attention residual and MLP. The central observation is that preserving the recognizable appearance of a reconstructed point cloud does not ensure accurate camera poses.

Camera trajectories. Figure 10 makes the camera-pose failure visible. At 500 frames, the deep-asymmetric trajectory deviates substantially from ground truth, whereas RegimeVGGT follows it more closely. This difference is much more pronounced than the change in the overall scene appearance in Figure 11. Retaining camera/register queries therefore does not suffice to maintain pose accuracy when patch updates are frozen in the deep global blocks. At 1000 frames, all three methods exhibit substantial drift, including RegimeVGGT, showing that the protected design also has limits on long inputs.

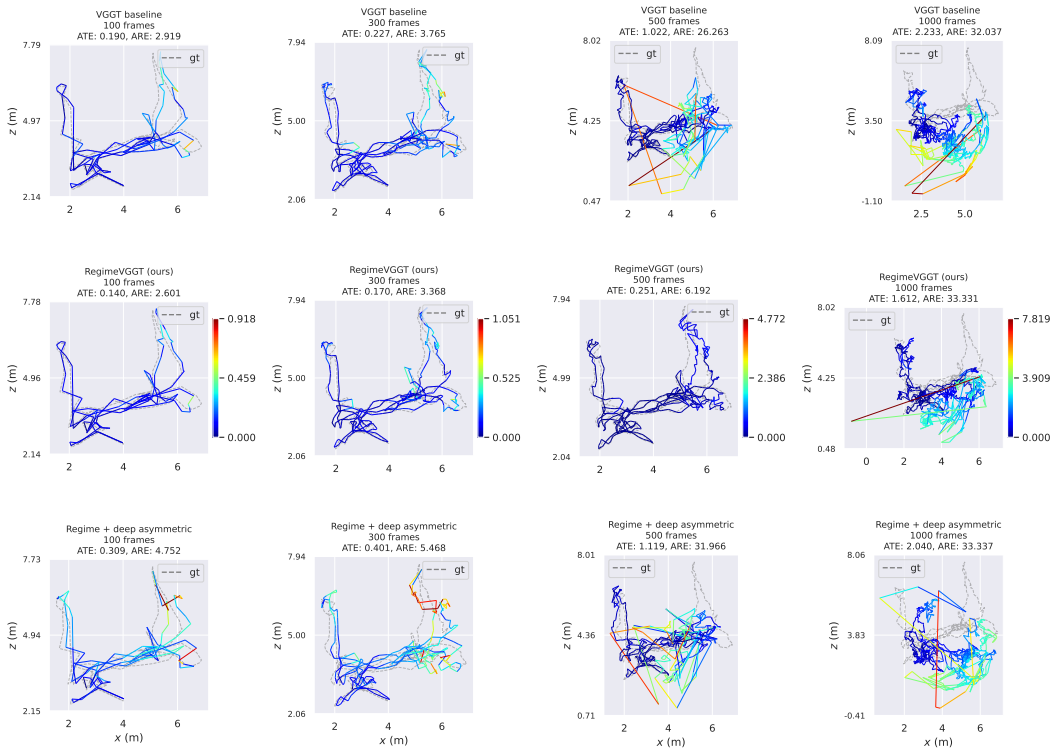


Figure 10: **Camera-pose degradation under deep-band asymmetric updates.** Rows show the VGGT baseline, RegimeVGGT, and RegimeVGGT with deep-asymmetric updates; columns show 100, 300, 500, and 1000 input frames. Gray dashed curves denote ground truth and colored curves denote predicted trajectories in the x–z plane. Each panel reports ATE and ARE. The deep-asymmetric variant exhibits substantial pose errors despite the recognizable reconstructions in Figure 11.

Point-cloud reconstructions. Figure 11 shows that the deep-asymmetric variant retains recognizable room structure through 500 frames, although its reconstruction is less coherent than that of RegimeVGGT. The visual change in coarse scene layout is relatively modest compared with the severe trajectory degradation in Figure 10. This separation between reconstruction appearance and pose accuracy is the key observation behind Step 2: visually plausible point clouds can conceal substantial camera-pose errors. At 1000 frames, visible reconstruction degradation is present in all three settings, so the relative preservation of scene structure should not be interpreted as lossless compression.

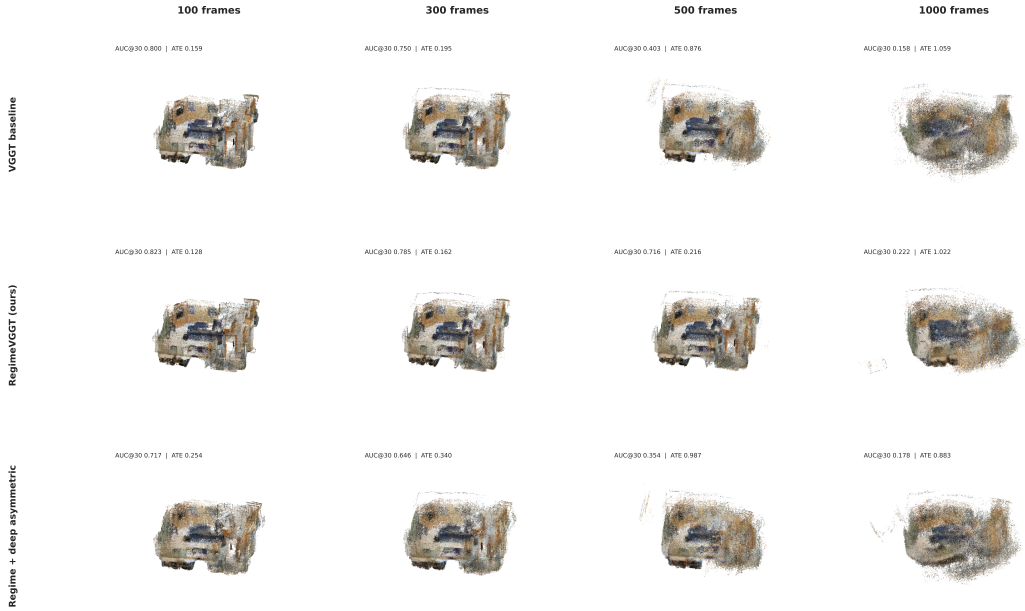


Figure 11: **Recognizable reconstruction despite camera-pose degradation.** Rows show the VGGT baseline, RegimeVGGT, and RegimeVGGT with deep-asymmetric updates; columns show 100, 300, 500, and 1000 input frames. AUC@30 and ATE are annotated above each reconstruction. The deep-asymmetric variant preserves the overall scene layout through 500 frames, even though the trajectory comparison in Figure 10 exposes large pose errors. All three settings show visible reconstruction degradation at 1000 frames.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction state both contributions (training-free Saliency-Guided Banded Merging and Selectively Protected K/V Downsampling guided by the three-band rank analysis) and the empirical results (up to 6.7× speedup over VGGT* on ScanNet-1000 at preserved reconstruction quality), all of which are supported by Sections 3–4 and the ablations in Section 4.4.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Limitations are discussed in Section 6: (i) RegimeVGGT uses fixed per-band compression rates (ρ_b, σ_b) chosen from scene-pooled diagnostics rather than adapting them per input, so input-adaptive rate selection could further improve the speed–quality trade-off without changing the overall design; (ii) the protected token set uses frozen DINOv2 `[CLS]` saliency, which is the strongest training-free proxy among the tested importance scores, and a learned geometry-aware selector under fine-tuning or additional supervision could identify geometry-critical tokens more directly and may further improve preservation under aggressive compression.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.

- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

   Justification: The paper contains no formal theorems requiring proof. The complexity accounting in Appendix B is direct arithmetic over the per-band attention costs, and the algebraic equivalences underlying the size-aware attention construction (Eq. 3) and the per-band scheduling (Eqs. 1, 4) are stated with their assumptions in Sections 3.2–3.3.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: All hyperparameters—band boundaries (L1–L10, L11–L18, L19–L24), merge ratios ρ_s=0.99, ρ_m=0.50, ρ_d=0.99 (Eq. 1), K/V rates σ_s=1.5, σ_m=1.3, σ_d=1.7 (Eq. 4), protection fraction α=0.1, cache anchors $\ell \in \{1, 11, 19\}$, which are specified in Sections 3.2–3.3 and Appendix A.3.3. Datasets, splits, evaluation protocols, and baselines are listed in Section 4.1; baselines use authors' released configurations. Code is publicly available.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.

- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: Code, configurations, and run scripts are publicly available at `https://github.com/Tracyou07/RegimeVGGT`. All evaluation datasets are standard public benchmarks (7-Scenes, NRGBD, ScanNet, Tanks & Temples, DTU) cited at first use in Section 4.1.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Section 4.1 specifies the datasets (7-Scenes, NRGBD, ScanNet-50, Tanks & Temples, DTU), evaluation metrics (Acc / Comp / NC / CD for reconstruction; AUC@5/15/30, ATE / ARE / RPE for pose), baselines (VGGT*, FastVGGT, S-VGGT, Fast3R, CUT3R), hardware (single NVIDIA H800 80GB), and input resolution (518×392). The method is training-free, so no optimizer or training schedule applies; per-band hyperparameters are listed in Sections 3.2–3.3 and Appendix A.3.3.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Tables 1 and 3 report cross-scene *mean and median* for each metric, providing both central tendency and an outlier-robust summary across the evaluated scenes (7-Scenes and NRGBD for Table 1; six Tanks & Temples scenes for Table 3); per-scene rows are listed alongside so that variability is fully recoverable. Tables 4 and 6 report aggregate values without per-scene breakdown (DTU as a 22-scan average; Table 6 ablations are scene-pooled). Long-sequence ScanNet-50 results (Tables 2, 5) report a single trajectory per sequence length and therefore do not carry a cross-scene summary at that scale; the analysis-level statistical reliability is verified by a permutation test (n=200) in Appendix A.1.1 and seed averaging (n_{perm}=5) in Appendix A.1.2.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: All inference experiments are run on a single NVIDIA H800 GPU with 80GB memory at input resolution 518×392 (Section 4.1). Per-method wall-clock times are reported in Tables 1–5, and peak GPU memory is reported in Table 7. The full main-paper evaluation suite takes approximately 50 GPU-hours on this hardware; analysis experiments in Appendix A (probing, causal intervention, rank spectrum, importance ablation, K/V step ablations) add roughly 30 GPU-hours.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.

- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We have reviewed the NeurIPS Code of Ethics and confirm that the work conforms with it in every respect: there are no human subjects, no scraped or unlicensed data, and no models trained from scratch; all datasets and pretrained models are standard 3D-reconstruction assets used under their published licenses, and the released code is anonymized.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: *Positive societal impacts.* By making 1000-frame inference feasible on a single 80 GB GPU (Table 7), RegimeVGGT removes the multi-GPU prerequisite for large-scale 3D reconstruction and lowers the access barrier for academic groups, small organizations, and users in compute-constrained regions. This broadens the reach of beneficial downstream applications including robotics safety, augmented reality, infrastructure inspection, and disaster-response mapping, and reduces per-job energy consumption for an increasingly compute-heavy class of vision models. *Negative societal impacts.* Faster, cheaper 3D reconstruction is intrinsically dual-use: the same efficiency gains that lower barriers for legitimate applications also lower them for unauthorized 3D mapping of private spaces and large-scale visual surveillance from publicly available video. Our acceleration recipe introduces no new generative capability or data product beyond VGGT itself; its societal contribution is to widen the deployment range of an existing capability, and the corresponding dual-use considerations of the underlying model transfer with it. We view explicit acknowledgment of this trade-off as the responsible default when releasing efficiency improvements to widely-used 3D foundation models.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to

generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.

- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper releases code for an inference-time accelerator and uses publicly available datasets and pretrained models without modifying their release. No new high-risk models, weights, or scraped datasets are introduced.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All used assets are cited at first use with their original publications. Pretrained models: VGGT [32] and DINOv2 [20], used under their published research-use terms. Concurrent acceleration baselines: FastVGGT [25], S-VGGT [17], Fast3R [39], CUT3R [33]. Datasets: 7-Scenes [26], NRGBD [2], ScanNet [6], Tanks & Temples [14], DTU [12], used under their respective research-use licenses.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.

- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We release the RegimeVGGT implementation as code at `https://github.com/Tracyou07/RegimeVGGT`. The release includes a README with installation, environment, and reproduction commands. No new datasets or pretrained model weights are introduced.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing or research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve human-subject research; no IRB approval is required.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLMs were used only for writing and editing assistance and were not part of the core methodology, experimental design, or analysis.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.