SAF3R: Dynamic Sparse Attention for Feed-Forward 3D Reconstruction Transformers

Jianing Deng[1], Yuanzhe Li[2], Jialu Wang[3], Song Wang[4], Tianlong Chen[5], Huanrui Yang[2], and Jingtong Hu[1†]

[1] University of Pittsburgh, Pittsburgh, PA, USA
[2] University of Arizona, Tucson, AZ, USA
[3] Tongji University, Shanghai, China
[4] University of Central Florida, Orlando, FL, USA
[5] University of North Carolina at Chapel Hill, Chapel Hill, NC, USA
{jid70,jthu}@pitt.edu

Abstract. Feed-forward 3D reconstruction (F3R) transformers have recently achieved remarkable success. However, scaling them to long image sequences remains challenging, as the quadratic complexity of cross-view global attention quickly becomes the dominant computational bottleneck. While recent efforts attempt to improve efficiency through compressed or sparse attention, they fail to fully exploit the inherent sparsity and dynamic behavior of global attention. In this work, we present a comprehensive analysis of global attention across multiple F3R transformers and reveal that attention patterns are highly heterogeneous, dynamic, and extremely sparse across layers and attention heads. Motivated by these findings, we propose SAF3R, a training-free dynamic sparse attention framework tailored to F3R transformers. SAF3R integrates tailored sparse attention mechanisms with offline head profiling and an efficient online adaptation strategy to match input-dependent attention behaviors. Extensive experiments demonstrate that SAF3R achieves high sparsity ratios while preserving camera pose estimation and 3D reconstruction quality, translating into substantial end-to-end speedup on F3R transformers compared to existing methods. Code is available at `https://github.com/jndeng/SAF3R`.

Keywords: 3D Reconstruction · Transformers · Sparse Attention

1 Introduction

Recent advances in deep learning have catalyzed a paradigm shift in 3D geometric estimation, transitioning from iterative optimization-based pipelines [10, 20, 21, 24] to end-to-end neural architectures that directly infer geometry from raw images [14,32,34,42,43]. In particular, large-scale feed-forward 3D reconstruction (F3R) models [12, 15, 31, 36] have achieved remarkable progress in geometric reconstruction and multi-view scene understanding. These models adopt a unified

† Corresponding author.

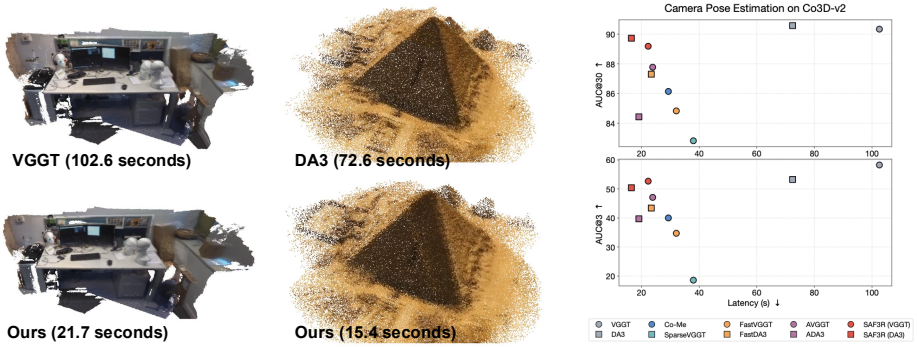

Fig. 1: Left: SAF3R accelerates feed-forward 3D reconstruction (F3R) models while preserving reconstruction quality. **Right:** SAF3R outperforms prior efficient F3R methods on camera pose estimation and achieves performance close to the original model.

transformer architecture [29] that jointly performs multiple geometric tasks, including camera pose estimation and dense point map prediction. A key characteristic of F3R transformers is the use of cross-view multi-head self-attention [9, 29] to model long-range geometric relationships across frames. By allowing tokens from all views to interact globally, they directly infer consistent scene geometry and camera poses in an end-to-end feed-forward manner, eliminating the need for explicit feature matching or global optimization. Fig. 2 illustrates the general architecture shared by existing F3R transformers [12, 15, 31, 36].

However, scaling F3R transformers to long image sequences remains challenging, as the quadratic complexity of cross-view global attention quickly dominates memory consumption and latency, severely limiting scalability and real-time deployment [23, 30]. To mitigate this issue, recent works explore efficient approximations to full global attention, including token compression [6, 23, 25, 35, 37] and sparse attention mechanisms [19, 26, 30]. A summary of these methods is provided in Tab. 1. While these approaches reduce computational cost, they typically adopt static or homogeneous sparsity configurations shared across layers and heads. Such designs overlook a critical property of F3R transformers: their unified multi-task architecture, which jointly models sparse camera pose estimation and dense point cloud reconstruction, induces inherently non-uniform and dynamic global attention patterns. Instead, our analysis reveals that global attention behavior varies significantly across layers, heads, and input scenes, suggesting that naïve sparsification may discard critical cross-view geometric information or waste computation on uninformative attention links.

Based on these insights, we introduce SAF3R, a training-free dynamic sparse attention inference framework tailored to F3R transformers. The SAF3R framework determines and applies the most appropriate sparse attention pattern for *each individual head* across all global attention layers, fully leveraging the heterogeneous and input-dependent nature of global attention. Through comprehensive analysis, we first categorize global attention heads into four types and design tai-

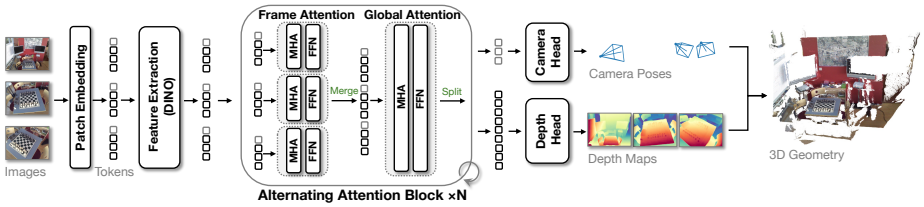

Fig. 2: Overview of the architecture shared by existing F3R transformers [12,15,31,36]. Image patches are first encoded by a DINO [5, 17] encoder and then processed by N alternating attention blocks of local and global attention, each consisting of multi-head self-attention (MHA) and feed-forward networks (FFN).

lored sparse attention kernels for each. Building on this design, SAF3R adopts a two-stage pipeline consisting of offline head profiling and online pattern adaptation. In the offline stage, we search for head-specific sparse configurations starting from a uniform-stride sparse attention baseline [23, 26], progressively refining it while ensuring no increase in approximation error on a calibration dataset. During inference, SAF3R executes these optimized configurations and dynamically adjusts input-dependent sparse patterns (*e.g.*, Top-K indices) through lightweight approximation and index caching. We evaluate SAF3R on multiple F3R models and benchmarks. As shown in Fig. 1, our method maintains both camera pose estimation and point cloud reconstruction quality while providing up to 7× end-to-end speedup on long input sequences.

In summary, our contributions are three-fold: (1) We conduct a comprehensive head-wise analysis of global attention in multiple F3R models, revealing heterogeneous and dynamic sparsity patterns across layers and heads. (2) We propose SAF3R, a training-free dynamic sparse attention framework that exploits head-specific sparsity patterns to accelerate model inference. (3) We demonstrate significant computational savings and substantial inference speedup on multiple F3R models while maintaining camera pose and 3D reconstruction accuracy.

2 Related Work

2.1 Feed-forward 3D Reconstruction Transformers

Early 3D reconstruction pipelines rely on multi-stage optimization methods such as Structure-from-Motion (SfM) [20] and Multi-View Stereo (MVS) [10], which have been widely adopted in practice [16, 20, 21] but are computationally expensive and prone to error accumulation. Recent advances in deep learning have enabled feed-forward 3D reconstruction (F3R) models that directly infer scene geometry from unseen images [12, 14, 15, 31–34, 36, 42, 43]. A key milestone is VGGT [31], which jointly predicts camera poses, depth maps, point maps, and tracking results within a unified transformer architecture. Subsequent works, including π^3 [36], MapAnything [12], and Depth Anything 3 [15], further improve permutation invariance, metric scale recovery, and reconstruction perfor-

Table 1: Comparison of efficient global attention mechanisms for F3R transformers.

Method	Compression Strategy	Policy Scope	Input-adaptive	Head-wise	Dynamic Sparsity
FastVGGT [23]	Token Merging	Model-level	✗	✗	✗
FlashVGGT [37]	Token Merging	Model-level	✓	✗	✗
Co-Me [6]	Token Merging	Model-level	✓	✗	✗
LiteVGGT [25]	Token Merging	Model-level	✓	✗	✗
HTTM [35]	Token Merging	Model-level	✗	✓	✗
SparseVGGT [30]	Sparse Attention	Model-level	✓	✗	✗
Speed3R [19]	Sparse Attention	Model-level	✓	✗	✗
AVGGT [26]	Sparse Attention	Stage-level	✗	✗	✗
SAF3R (Ours)	Sparse Attention	Head-level	✓	✓	✓

mance. Despite their strong performance, these models rely heavily on global self-attention to capture cross-view geometric relationships, whose quadratic complexity with respect to token length becomes a major bottleneck for long-sequence inference.

2.2 Leveraging Sparsity in Attention

Sparse attention has emerged as an effective strategy to alleviate the quadratic computational cost of attention-based architectures across various domains [11, 13, 38–41, 44, 46–48]. SpargeAttn [47] uses a low-resolution attention approximation to identify important blocks and computes attention only on them. DuoAttention [39] identifies two general sparse attention patterns in large language models and learns to assign each head to one of them using synthetic calibration data. NSA [46] trains the model to combine several predefined sparse attention patterns. MInference [11] uses offline profiling to assign head-specific sparse attention patterns for efficient inference. FlexPrefill [13] and SVG [38] use online profiling to determine input- or head-specific sparse attention patterns. However, unlike the relatively stable attention patterns in language models or the spatio-temporal locality in vision transformers, F3R models exhibit highly heterogeneous and task-dependent attention distributions across layers and heads due to their multi-task nature. This motivates systematic head-level analysis and tailored sparse kernel designs for F3R transformers.

2.3 Efficient Global Attention for F3R Transformers

Recent works explore token compression and sparse attention mechanisms to alleviate the computational bottleneck of global attention. FastVGGT [23] adopts token merging [3, 4] to reduce the number of tokens before attention based on patch-level similarity. Subsequent works improve token merging with better anchor token selection. FlashVGGT [37] computes global descriptors and performs token merging via interpolation rather than simple averaging. Co-Me [6] uses confidence maps from F3R models to guide merging, while LiteVGGT [25] further exploits geometric cues, such as edges, to select important anchor tokens. HTTM [35] introduces head-wise token merging and spatio-temporal block

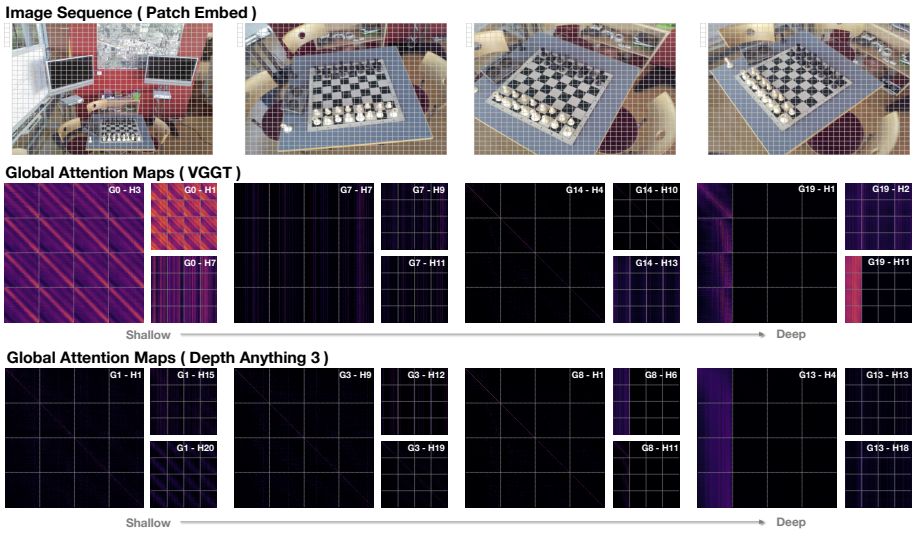

Fig. 3: Representative global attention patterns across layers (G) and heads (H). Gray dashed lines in the attention maps separate tokens from different images. Similar observations apply to MapAnything and π^3 (see the appendix). Zoom in for best view.

grouping to reduce similarity-computation overhead. However, merging-based methods heavily depend on anchor token selection, typically operate at relatively low sparsity ratios, and introduce non-trivial overhead for identifying tokens to merge. On the other hand, SparseVGGT [30] and AVGGT [26] approximate global attention with structured sparsity, yet their use of simple predefined sparsity patterns fails to model heterogeneous attention distributions. Speed3R [19] introduces trainable sparse attention to approximate global attention, but requires costly model retraining. As shown in Tab. 1, existing methods do not jointly capture the dynamic and head-level heterogeneous behavior of global attention, limiting their ability to fully exploit attention sparsity.

3 Analysis of Global Attention in F3R Transformers

In this section, we conduct a comprehensive study of cross-view global attention in four representative F3R models: VGGT [31], π^3 [36], MapAnything [12], and Depth Anything 3 (DA3) [15]. Since attention is computed in a multi-head manner, we perform head-level analysis to capture heterogeneous behaviors across heads. We use randomly sampled image sequences from the 7Scenes dataset [24]. Below, we summarize our observations, which provide key insights into the underlying mechanisms, sparsity characteristics, and opportunities for acceleration.

Observation#1: Global attention patterns are heterogeneous across models, layers, and heads. Fig. 3 visualizes global attention maps across layers and heads in different F3R models. We observe a clear three-stage progression

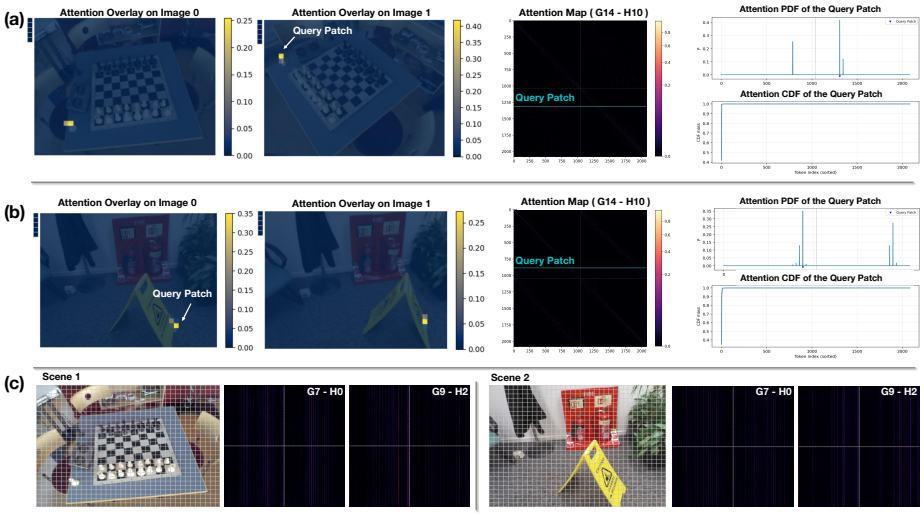

Fig. 4: (a) and (b) illustrate the high sparsity of correspondence heads, while (c) shows the content-dependent dynamic nature of vertical-line heads.

with distinct functional roles and sparsity patterns. In shallow layers, attention mainly performs local aggregation or focuses on a few salient patches, showing concentrated and often static sparsity with limited cross-view interaction. In middle layers, attention establishes point-to-point correspondences across views, where sparsity remains high but becomes semantically structured and content-dependent. In later layers, attention becomes more diffuse and mainly refines representations and camera poses relative to the anchor frame. This stage-wise behavior is consistent with prior findings [23, 26, 30].

Beyond this stage-wise trend, global attention exhibits more nuanced heterogeneity. Attention patterns vary substantially *across heads*, even within the same layer, suggesting that functional roles cannot be fully characterized at the coarse layer or stage level, as commonly assumed in prior work [23, 25, 26, 30]. Moreover, attention behaviors differ across F3R models. For example, DA3 does not show the positional-encoding-induced local attention patterns observed in the early and late layers of VGGT. These findings highlight the need to model heterogeneity across both layers and heads, motivating a head-wise heterogeneous compression strategy for F3R transformers.

Observation#2: Attention heads can be categorized into four types based on their behaviors. (1) **Position heads**, whose attention is dominated by positional encoding rather than semantic features, exhibiting patterns that are either fixed across all image blocks (*e.g.*, G0-H3 and G0-H1) or predominantly attend to the anchor image which defines the camera coordinate frame (*e.g.*, G19-H1 and G19-H11 in VGGT); (2) **Vertical-line heads**, where most queries consistently attend to a few critical keys (salient image patch tokens or sink tokens [8, 40]), forming prominent "lines" in the attention maps (*e.g.*, G7-H7

and G14-H4); (3) **Correspondence heads**, where attention is primarily driven by semantic features, and each query selectively attends to the key corresponding to the same 3D location as shown in Fig. 4 (*e.g.*, G14-H4 and G14-H10); (4) **Scanning heads**, which constitute the majority of attention heads, exhibit diverse attention patterns distributed across positions in all images. The head-level diversity also induces distinct sparsity behaviors across head types and layers. Position and scanning heads are typically less sparse, while correspondence heads are highly sparse, attending to only a few key tokens. Vertical-line heads exhibit layer-dependent sparsity, with early-layer heads being sparser than middle ones.

Observation#3: Some global attention heads are highly dynamic (*i.e.*, input-dependent). We observe that global attention heads exhibit both static and dynamic behaviors. While some heads produce attention patterns that remain largely consistent across inputs, others vary depending on the visual content. Fig. 4 visualizes attention maps of two types of dynamic heads. As shown, these heads follow a fixed attention *type* (*e.g.*, capturing correspondence relationships or attending to a few specific keys), while the exact attention patterns remain input-dependent. This suggests that sparse attention patterns should not be fixed, but should instead adapt dynamically to the input.

4 Proposed Dynamic Sparse Attention Framework

We present SAF3R, a two-stage dynamic Sparse Attention framework that exploits heterogeneous sparsity in F3R transformers for efficient inference. Fig. 5 illustrates the overall framework. We first introduce sparse attention mechanisms tailored to head-wise attention patterns (Sec. 4.1). We then present an offline profiling strategy to identify the appropriate sparse type for each head (Sec. 4.2), followed by an online adaptation mechanism that dynamically adjusts attention patterns based on head types and input content during inference (Sec. 4.3).

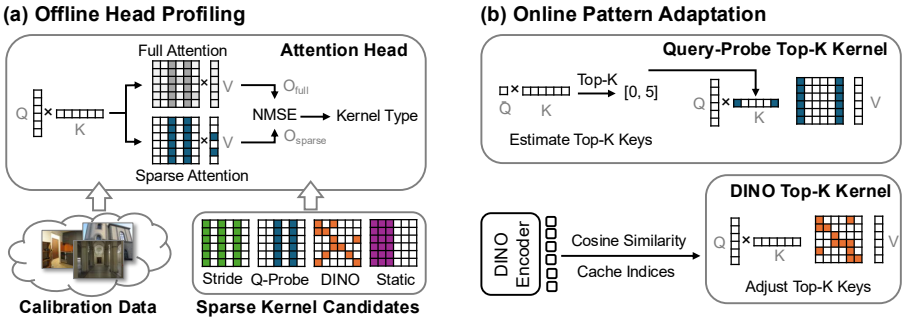

Fig. 5: Overview of the proposed SAF3R framework.

4.1 Tailored Sparse Attention Kernels

Based on the head categorization in Sec. 3, we develop tailored sparse attention kernels for each attention type to exploit their distinct sparsity patterns. Each kernel reduces the computational complexity of attention from $\mathcal{O}(N^2)$ to $\mathcal{O}(N)$.

Static Kernel. Since attention in positional heads is content-independent, we design a static kernel that computes fixed sparse attention patterns, *e.g.*, allowing all queries to attend to a designated anchor frame or broadcasting the first frame's local attention map to other frames for cross-view interaction.

Query-Probe Top-K Kernel. For vertical-line heads, where attention mass is concentrated on a few critical key tokens (forming several prominent lines in the attention map), we restrict computation to attention scores between all queries and these key tokens, omitting the remaining interactions. As analyzed in Sec. 3, the small set of key tokens is dynamic and input-dependent, which prevents the use of a fixed attention pattern. Therefore, we estimate key positions on the fly using a lightweight pre-computation, as detailed in Sec. 4.3.

DINO Top-K Kernel. Correspondence heads exhibit clear semantic behavior, attending to relevant patches at corresponding spatial locations across frames. We observe that DINO patch features already capture such semantic information and can serve as effective correspondence estimators, as visualized in the appendix. Therefore, we pre-compute the cosine similarity between each patch token and all other patch tokens using the DINO features, and select the Top-K most similar tokens from each frame as the keys and values for attention. Empirically, this strategy provides strong coverage of the full attention distribution with a relatively small K.

Uniform Sampling Kernel. For scanning heads, attention patterns are diverse and broadly distributed across tokens. Unlike other head types that focus on a small subset of keys, scanning heads aggregate information from a wide range of spatial locations without clear structural regularity, making pattern-specific key selection prone to missing important interactions. Therefore, following prior work [23, 26], we adopt a uniform sampling kernel that selects key positions with a fixed stride. This strategy provides an effective approximation by evenly covering the token space. In practice, it performs well at relatively high sampling ratios (*i.e.*, retaining 25%–50% of keys), but its performance degrades rapidly as the ratio decreases.

4.2 Offline Head Profiling

With tailored sparse kernels for each attention type, the remaining challenge is how to assign each attention head to the appropriate category to fully exploit sparsity while maintaining performance. To this end, we propose a simple yet effective sparsity search strategy based on progressive local substitution. Specifically, the search starts from a baseline configuration c_{base} that uses a uniform sampling kernel with sparsity ratio σ. For each head h, we then define a candidate set of configurations $\mathcal{C}_h = \{c_1, c_2, \ldots, c_S\}$ that includes alternating sparse attention kernels (*e.g.*, DINO Top-K and Query-Probe Top-K) under different

settings (*e.g.*, K). All configurations in the candidate set are designed to have *equal or higher sparsity* than the baseline uniform sampling kernel. Next, we compute the exact full-attention output $\mathbf{O}_{\text{full}}^{(h)} \in \mathbb{R}^{N\times D}$ and the sparse attention output $\mathbf{O}_{\text{sparse}}^{(h,c)} \in \mathbb{R}^{N\times D}$ under configuration c, and define the approximation cost using Normalized Mean Squared Error (NMSE):

$$E(c) = \mathbb{E}_{\mathcal{D}_{\text{calib}}}\left[\frac{\|\mathbf{O}_{\text{full}}^{(h)} - \mathbf{O}_{\text{sparse}}^{(h,c)}\|_2^2}{\|\mathbf{O}_{\text{full}}^{(h)}\|_2^2}\right], \tag{1}$$

where $\mathcal{D}_{\text{calib}}$ is a small calibration dataset used for profiling.

We then evaluate candidate sparse configurations using a progressive substitution strategy. Starting from the baseline configuration c_{base}, we iteratively update the current best configuration whenever a candidate achieves lower approximation error. Formally, the selected configuration minimizes the approximation error under the constraint that its computational complexity does not exceed that of the baseline:

$$c^* = \underset{c\in\mathcal{C}_h\cup\{c_{\text{base}}\}}{\arg\min}\ E(c) \quad \text{s.t.} \quad \Omega(c) \le \Omega(c_{\text{base}}), \tag{2}$$

where $\Omega(c)$ denotes the computational cost of configuration c (measured by floating-point operations). The progressive replacement produces a heterogeneous head-wise sparsity configuration that preserves the accuracy of the baseline while achieving equal or lower inference cost. Empirically, this simple greedy search significantly increases sparsity while incurring no accuracy loss.

4.3 Online Pattern Adaptation

In offline profiling stage, we determine the appropriate kernel and configuration for each attention head. Due to the dynamic nature of Query-probe and DINO Top-K heads, we further introduce an online adaptation mechanism to refine the attention patterns for each input sequence during inference.

Query-Probe Top-K Kernel. One intuitive approach is to compute the full pre-softmax attention matrix, average the scores across queries, and select the Top-K keys. However, this requires $\mathcal{O}(N^2)$ attention computation, eliminating any potential speedup. Instead, we show that selecting keys based on the column-wise average of the full attention scores admits an equivalent yet more efficient $\mathcal{O}(N)$ formulation. Let $Q, K \in \mathbb{R}^{N\times D}$ denote the query and key matrices, and $S = QK^\top/\sqrt{D}$ the pre-softmax attention scores. By the linearity of inner products, the column-wise average of S across queries can be written as the similarity between each key and the mean query $\bar{q} = \frac{1}{N}\sum_{i=1}^{N} q_i$. Therefore, selecting the Top-K columns according to the averaged attention scores is equivalent to computing a single probe vector $\bar{q}K^\top/\sqrt{D}$ and ranking keys accordingly. This preserves the exact ranking induced by column-wise averaging of the full attention scores while incurring only negligible overhead.

DINO Top-K Kernel. For each input sequence, we extract the output from the last DINO block and compute the global cosine similarity between all

patches, storing the indices of the per-frame Top-K matched patches. Although this introduces an $\mathcal{O}(N^2)$ computation, the indices are cached and reused by all subsequent DINO Top-K kernels, thereby amortizing the computational cost. We explore token-wise and block-wise implementations, which compute exact token-level similarities and block-level similarities averaged over multiple tokens, respectively, offering trade-offs between efficiency and retrieval accuracy.

5 Experiments

5.1 Experimental Setup

Evaluation benchmarks and metrics. We evaluate F3R transformers for camera pose estimation and point cloud reconstruction under both sparse and dense settings. For sparse evaluation, we uniformly sample 128 frames from each scene (or use all frames if fewer are available) in the following datasets: Co3D-v2 [18], RealEstate10K [49], 7Scenes [24], HiRoom [15], ETH3D [22], ScanNet++ [45], and DTU [1]. For dense evaluation, we uniformly sample 300 - 700 frames from each scene in 7Scenes [24] and ScanNet [7]. We follow the DA3-bench [15] evaluation protocol for both tasks across all datasets, except for ScanNet, where we adopt the evaluation protocol from FastVGGT [23].

For camera pose evaluation, we report Area Under the Curve (AUC) following [15, 31], where accuracy is defined as the minimum of relative rotation accuracy (RRA) and relative translation accuracy (RTA). Results at 3° and 30° are reported. We additionally report Absolute Trajectory Error (ATE), Absolute Rotation Error (ARE), and Relative Pose Error (RPE). For point cloud reconstruction, we report Accuracy, Completeness, Chamfer Distance (CD), and F1-score following [15].

Baselines. We evaluate on three representative large-scale F3R transformers: VGGT [31], π^3 [36], and Depth Anything 3 (DA3-Giant) [15], which contain 24, 18, and 14 global attention blocks, respectively. We compare against state-of-the-art token compression approaches designed for F3R models, including FastVGGT [23], Co-Me [6], as well as sparse attention methods SparseVGGT [30] and AVGGT [26] [6]. For VGGT, we adopt the memory-efficient implementation from FastVGGT [23]. The merging ratios are set to 0.9 and 0.5 for FastVGGT and Co-Me, respectively, following the official implementations. For SparseVGGT [30], we adopt a block sparsity ratio of 0.75, while for AVGGT we use a subsampling factor of 4. Except for Co-Me, all compared methods focus on optimizing global attention while leaving the rest of the model unchanged.

Implementation details. The proposed framework is implemented in PyTorch [2] with custom Triton kernels [28]. For offline head profiling, we sample 10 training sequences from ETH3D [22] for calibration, spanning indoor and

[6] Since no public code is available, we re-implement the method based on the paper.

Table 2: Comparisons of **camera pose estimation** accuracy under **sparse settings** (10 - 128 images). We report both AUC@3 (↑) and AUC@30 (↑) metrics. Results are in percentages (%). Full-attention baselines are marked in gray.

Methods	Co3D-v2 [18]		Re10K [49]		7Scenes [24]		ETH3D [22]		ScanNet++ [45]		HiRoom [15]		DTU [1]	
	AUC@30	AUC@3	AUC@30	AUC@3	AUC@30	AUC@3	AUC@30	AUC@3	AUC@30	AUC@3	AUC@30	AUC@3	AUC@30	AUC@3
VGGT [31]	90.34	58.26	80.11	23.99	84.64	23.66	81.68	29.42	94.78	61.70	88.07	48.05	85.26	79.66
FastVGGT [23]	84.84	34.73	76.99	18.39	83.01	18.67	**76.05**	19.83	90.15	39.81	75.36	25.23	81.77	52.95
SparseVGGT [30]	82.83	18.59	69.80	13.76	81.17	14.72	65.25	12.29	88.12	21.44	68.45	23.16	82.05	50.51
Co-Me [6]	86.15	40.02	75.25	17.64	80.70	16.19	70.87	18.25	88.76	33.84	76.38	28.33	84.10	72.31
AVGGT [26]	87.78	47.10	78.22	20.85	**84.51**	22.25	72.94	19.84	**92.31**	**51.57**	80.54	34.22	**85.20**	**79.60**
SAF3R (Ours)	**89.19**	**52.68**	**78.35**	**23.04**	84.49	**23.57**	75.65	**22.49**	91.33	47.65	73.93	**37.51**	84.82	77.71
π^3 [36]	90.91	60.83	90.56	57.03	86.15	24.09	85.57	32.70	93.83	57.05	94.47	64.73	94.85	62.95
Fastπ^3 [23]	87.56	40.49	88.40	47.20	85.30	23.80	79.32	24.20	91.46	44.28	87.24	37.92	93.03	54.24
Sparseπ^3 [30]	82.14	15.14	81.52	30.48	81.15	12.25	75.36	14.37	84.89	16.14	82.70	28.57	92.90	46.30
Aπ^3 [26]	89.27	50.69	89.06	52.11	**86.07**	**24.22**	78.03	22.74	92.68	50.42	90.40	46.92	93.45	55.30
SAF3R (Ours)	**90.14**	**53.98**	**89.69**	**52.44**	85.85	23.48	**83.03**	**24.30**	**93.48**	**50.76**	**92.13**	**51.96**	**94.79**	**61.46**
DA3 [15]	90.58	53.28	91.10	57.91	86.71	28.59	91.40	48.37	98.17	84.85	95.92	80.20	99.38	93.99
FastDA3 [23]	87.31	43.42	88.97	52.16	85.90	28.53	**88.53**	**37.89**	**94.58**	**76.36**	92.93	63.23	98.82	88.69
ADA3 [26]	84.44	39.72	83.95	37.44	85.15	25.05	64.13	16.60	71.26	30.84	85.36	49.32	97.36	77.91
SAF3R (Ours)	**89.73**	**50.42**	**89.94**	**52.92**	**86.80**	**28.84**	87.85	35.39	92.84	62.79	**94.68**	**79.15**	**98.99**	**90.39**

Table 3: Comparison of **point cloud reconstruction** performance under **sparse settings** (10 - 128 images). We report Accuracy (Acc.), Completeness (Comp.), Chamfer Distance (CD), and F1-score (F1). Full-attention baselines are marked in gray.

Methods	7Scenes [24]				ETH3D [22]				ScanNet++ [45]				HiRoom [15]			
	Acc.↓	Comp.↓	CD↓	F1↑	Acc.↓	Comp.↓	CD↓	F1↑	Acc.↓	Comp.↓	CD↓	F1↑	Acc.↓	Comp.↓	CD↓	F1↑
VGGT [31]	0.155	0.125	0.140	0.480	0.306	0.831	0.568	0.634	0.045	0.092	0.069	0.692	0.107	0.092	0.100	0.554
FastVGGT [23]	**0.127**	0.134	**0.130**	**0.478**	**0.355**	**0.926**	**0.640**	0.519	**0.055**	0.109	**0.082**	0.568	0.181	0.161	0.171	0.429
SparseVGGT [30]	0.156	0.161	0.159	0.371	0.426	1.779	1.102	0.460	0.078	0.120	0.099	0.447	0.246	0.203	0.225	0.279
Co-Me [6]	0.121	0.158	0.140	0.455	0.557	1.221	0.889	0.548	0.094	0.114	0.104	0.486	0.210	0.134	0.172	0.406
AVGGT [26]	0.152	0.127	0.139	0.447	0.414	1.301	0.858	0.486	0.067	0.108	0.088	**0.571**	**0.156**	**0.123**	**0.140**	**0.467**
SAF3R (Ours)	0.144	**0.127**	0.136	0.473	0.406	1.048	0.727	**0.576**	0.066	**0.108**	0.087	0.562	0.257	0.184	0.220	0.459
π^3 [36]	0.134	0.098	0.116	0.430	0.270	0.762	0.516	0.695	0.047	0.092	0.070	0.649	0.053	0.043	0.048	0.737
Fastπ^3 [23]	0.141	**0.094**	0.117	**0.473**	0.290	0.761	0.525	0.621	0.063	0.105	0.084	0.519	0.127	0.099	0.113	0.352
Sparseπ^3 [30]	0.163	0.113	0.138	0.332	0.811	0.879	0.845	0.435	0.166	0.164	0.165	0.231	0.234	0.145	0.190	0.203
Aπ^3 [26]	**0.132**	0.099	**0.115**	0.433	0.292	0.826	0.559	0.665	0.054	0.097	0.075	0.569	0.099	**0.063**	**0.081**	**0.561**
SAF3R (Ours)	0.150	0.101	0.125	0.424	**0.274**	**0.760**	**0.517**	**0.667**	**0.051**	**0.095**	**0.073**	**0.598**	**0.092**	0.074	0.083	0.501
DA3 [15]	0.119	0.128	0.124	0.524	0.237	0.667	0.452	0.793	0.035	0.080	0.057	0.796	0.048	0.044	0.046	0.859
FastDA3 [23]	0.123	0.145	0.134	0.529	0.290	0.711	0.501	**0.734**	**0.034**	**0.082**	**0.058**	**0.782**	0.060	0.060	0.060	0.762
ADA3 [26]	**0.120**	0.144	**0.132**	**0.533**	0.792	1.727	1.260	0.420	0.172	0.193	0.182	0.467	0.163	0.097	0.130	0.631
SAF3R (Ours)	**0.120**	0.150	0.134	0.514	**0.281**	**0.666**	**0.474**	0.710	0.057	0.090	0.074	0.673	**0.053**	**0.048**	**0.051**	**0.842**

outdoor scenes. We use a global sparsity ratio of $\sigma = 0.85$ as the search baseline. The profiled configurations for each model are provided in the appendix. During online inference, we set K between 16 and 128 per frame for the DINO Top-K kernel depending on the scene, balancing efficiency and reconstruction accuracy. All experiments are conducted on a single NVIDIA H100 GPU (80 GiB VRAM).

5.2 Main Results

Camera pose estimation. Tab. 2 presents the comparison of camera pose estimation performance of F3R models under sparse settings. As shown, our method outperforms existing efficient F3R models on most datasets. Notably, it preserves camera pose accuracy even under very strict error tolerances, resulting in only negligible degradation on the challenging AUC@3 metric, while other methods suffer noticeable performance drops. As shown in Tab. 4, our method

Table 4: Comparisons of **camera pose and point cloud estimation** on ScanNet-50 [7] dataset under **dense settings**. The average speedup (Spd.) measured relative to the baseline is reported. Full-attention baselines are marked in gray.

Method	300 Images					500 Images					700 Images				
	ATE$^{\downarrow}$	ARE$^{\downarrow}$	RPE-T/R$^{\downarrow}$	CD$^{\downarrow}$	Spd.$^{\uparrow}$	ATE$^{\downarrow}$	ARE$^{\downarrow}$	RPE-T/R$^{\downarrow}$	CD$^{\downarrow}$	Spd.$^{\uparrow}$	ATE$^{\downarrow}$	ARE$^{\downarrow}$	RPE-T/R$^{\downarrow}$	CD$^{\downarrow}$	Spd.$^{\uparrow}$
VGGT [31]	0.14	3.64	0.03/0.74	0.45	1.0×	0.15	4.14	0.04/0.98	0.47	1.0×	0.19	4.63	0.04/0.97	0.47	1.0×
Fast-VGGT [23]	0.13	3.48	0.03/0.77	0.44	2.8×	0.13	3.54	0.03/0.62	0.46	3.2×	0.14	3.72	0.03/0.73	0.47	3.4×
Sparse-VGGT [30]	0.16	4.51	0.04/0.96	0.48	2.5×	0.18	5.49	0.04/1.30	0.48	2.7×	0.21	5.85	0.06/1.65	0.48	2.8×
Co-Me [6]	0.16	3.96	0.05/1.24	0.45	2.6×	0.15	4.02	0.05/1.19	0.45	3.5×	0.18	4.58	0.06/1.20	0.49	4.0×
A-VGGT [26]	0.15	3.95	0.04/0.97	0.44	3.1×	0.16	4.31	0.04/1.11	0.45	4.3×	0.20	5.37	0.05/1.60	0.46	4.6×
SAF3R (Ours)	0.15	3.97	0.04/0.85	0.45	3.2×	0.15	4.13	0.03/0.91	0.45	4.6×	0.17	4.47	0.04/1.03	0.46	4.7×
π^3 [36]	0.08	2.16	0.02/0.52	0.59	1.0×	0.08	2.17	0.01/0.43	0.59	1.0×	0.08	2.15	0.01/0.37	0.59	1.0×
Fast-π^3 [23]	0.09	2.37	0.02/0.55	0.59	2.0×	0.09	2.35	0.02/0.41	0.59	2.2×	0.09	2.39	0.02/0.45	0.60	2.3×
Sparse-π^3 [30]	0.14	4.99	0.02/0.61	0.58	2.7×	0.14	4.99	0.02/0.49	0.60	2.9×	0.14	4.98	0.02/0.43	0.59	3.1×
A-π^3 [26]	0.10	2.39	0.02/0.55	0.59	4.9×	0.09	2.36	0.02/0.46	0.59	5.9×	0.09	2.37	0.02/0.40	0.60	6.7×
SAF3R (Ours)	0.09	2.36	0.02/0.54	0.58	5.3×	0.09	2.40	0.02/0.44	0.58	6.5×	0.09	2.44	0.01/0.39	0.59	7.3×
DA3 [15]	0.13	3.68	0.02/0.82	0.40	1.0×	0.13	3.73	0.02/0.77	0.40	1.0×	OOM	OOM	OOM	OOM	OOM
Fast-DA3 [23]	0.13	3.77	0.02/1.00	0.40	2.7×	0.13	3.94	0.02/1.00	0.40	3.1×	OOM	OOM	OOM	OOM	OOM
A-DA3 [26]	0.35	12.43	0.11/7.34	0.43	3.2×	0.43	13.42	0.10/7.67	0.45	3.8×	OOM	OOM	OOM	OOM	OOM
SAF3R (Ours)	0.15	5.19	0.03/1.40	0.40	3.3×	0.16	5.71	0.04/1.89	0.43	3.8×	OOM	OOM	OOM	OOM	OOM

shows a slight performance drop under dense settings on the ScanNet dataset, particularly in camera rotation estimation, while still achieving relatively low error compared to the baseline models.

Point cloud reconstruction. We compare the point cloud reconstruction performance of F3R models under sparse and dense settings in Tab. 3 and Tab. 4, respectively. As shown, FastVGGT achieves strong performance on several datasets, even outperforming the baseline models in some cases. However, it relies on dense anchor sampling and therefore achieves relatively limited speedup compared to the baseline. While our method may underperform FastVGGT on certain datasets, it generally achieves reconstruction quality comparable to the baseline models and remains competitive with other efficient approaches, while delivering substantially higher speedup due to dynamic sparse attention.

Efficiency benchmarks. Fig. 6 presents the efficiency benchmarking results of different methods across models when processing a single scene from the 7Scenes dataset. For short sequences, the bottleneck lies in other components of the model, such as feed-forward networks, and global attention does not dominate the computation. As a result, all models achieve similar performance. As the sequence length increases, full-attention baselines become dominated by the quadratic cost of global attention. In contrast, with acceptable memory overhead from computing and caching Top-K indices, our method exploits dynamic sparsity in attention computation and achieves end-to-end speedups of up to 5.8×, 6.8×, and 3.9× for VGGT, π^3, and DA3, respectively.

Visualization of offline profiling results. Fig. 7 presents the offline head profiling results. As shown, different models exhibit varying degrees of head-wise heterogeneity, and those with greater heterogeneity benefit more from dynamic

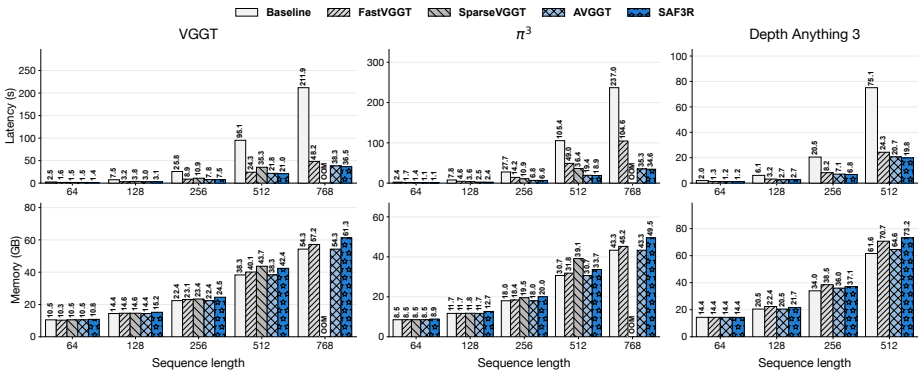

Fig. 6: Efficiency results for processing a single scene with different sequence lengths. Latency and memory are measured on scenes with fixed image resolution 480×640.

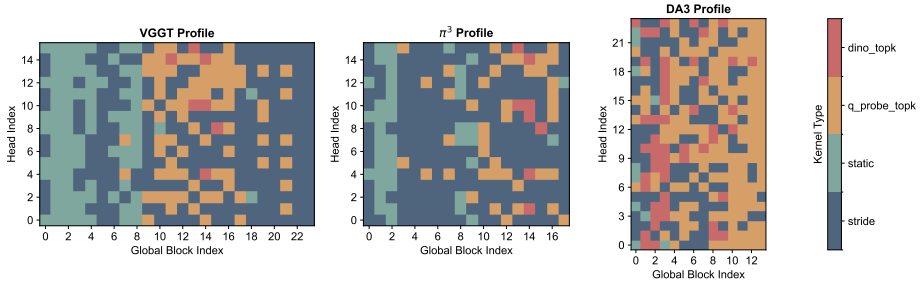

Fig. 7: Offline head profiling results for different F3R models. After profiling, each global attention head is assigned to one of four predefined kernel types.

sparse attention, highlighting the need for head-wise heterogeneous kernel configurations and the flexibility of our method.

5.3 Ablation Study

Effectiveness of tailored sparse kernels and offline search. We evaluate the effectiveness of the proposed offline search strategy in Tab. 5. Starting from a uniform sampling baseline with 85% sparsity, the search progressively replaces attention kernels with more suitable configurations for each head. The resulting model further increases the overall sparsity while preserving model accuracy. Specifically, the selected configurations reduce the number of attended keys by an additional 6% compared to the baseline, while slightly improving reconstruction quality. These results demonstrate that the proposed search strategy can effectively exploit heterogeneous attention behaviors across heads to achieve a better sparsity-accuracy trade-off.

Table 5: Effectiveness of the proposed tailored sparse kernels and profiling strategy (DINO Top-K=128). Camera pose estimation results (AUC@30 and AUC@3) are reported over five runs with different randomly sampled sets of scene images.

Model	Configs	Sparsity	7Scenes	ETH3D	ScanNet++	HiRoom	DTU
VGGT	Uniform	0.85	83.8±0.3 22.6±0.6	77.2±0.8 23.2±1.4	93.3±0.2 53.9±1.0	82.9±1.1 41.2±1.5	84.5±0.8 71.7±1.1
	+Static	0.90	84.4±0.3 23.3±0.5	76.9±0.8 23.5±1.2	93.9±0.3 54.3±0.9	82.2±1.2 40.9±1.1	84.8±1.0 72.5±1.4
	+Dynamic	0.91	84.8±0.4 24.2±0.6	78.1±0.9 24.9±1.6	93.6±0.3 54.8±0.9	78.8±1.2 39.1±0.9	85.4±0.8 74.7±1.7
DA3	Uniform	0.82	86.8±0.2 29.3±0.4	86.8±2.1 36.1±3.5	96.8±0.7 74.2±2.3	93.6±0.3 72.5±1.1	98.5±0.5 88.0±2.3
	+Static	0.85	86.8±0.3 29.3±0.6	88.4±1.5 38.4±3.2	96.8±0.4 75.3±2.2	93.9±0.7 71.9±1.3	99.1±0.5 88.5±2.1
	+Dynamic	0.87	86.8±0.2 29.4±0.5	89.1±2.6 39.3±3.4	97.5±0.4 79.8±2.6	94.9±0.9 81.8±1.9	98.5±0.5 87.2±2.8

Table 6: Effectiveness of DINO Top-K selection on 7Scenes dataset. Sparse attention is applied only to heads $\{4, 10, 14\}$ in layer 14 of VGGT.

Kernel Type	#Keys	AUC@30↑	AUC@3↑	CD↓
Full	100%	84.64	23.66	0.14
Skip	0%	2.56	0.00	0.97
Uniform Stride (s=64)	1.6%	82.13	19.94	0.15
Uniform Stride (s=32)	3.1%	84.45	22.64	0.14
DINO Top-K (K=2)	0.2%	84.40	22.92	0.14
DINO Top-K (K=4)	0.4%	84.51	23.34	0.14

Table 7: Effectiveness of DINO Top-K selection on Co3D-v2 dataset. Sparse attention is applied only to heads $\{4, 10, 14\}$ in layer 14 of VGGT.

Kernel Type	#Keys	AUC@30↑	AUC@3↑
Full	100%	90.34	58.26
Skip	0%	19.55	0.03
Uniform Stride (s=64)	1.6%	81.54	33.06
Uniform Stride (s=32)	3.1%	86.85	43.77
DINO Top-K (K=2)	0.2%	87.44	45.34
DINO Top-K (K=4)	0.4%	88.05	48.05

Effectiveness of DINO Top-K for correspondence heads. We compare the impact of different sparse attention kernels on the correspondence heads. Tab. 6 and Tab. 7 present the results of applying different sparse attention kernels to correspondence heads and their impact on model performance. As shown, skipping these heads results in severe performance degradation. While uniform sampling requires a large sampling ratio to cover the correspondence pattern, our DINO Top-K kernel achieves comparable results while attending to significantly fewer keys, resulting in a 16× increase in sparsity.

Further Analysis of the DINO Top-K kernel. DINO Top-K assumes that DINO features capture patches similar to those used by correspondence heads. This assumption can be violated in practice due to limited feature discriminability (*e.g.*, textureless regions, repeated patterns, illumination changes, and large viewpoint variations). When the assumption fails, correspondence heads may miss critical regions, leading to performance degradation, especially with small K. For example, we observe degradation on VGGT in certain HiRoom scenes, where small K performs worse than uniform sampling. Increasing K effectively mitigates this issue, albeit at the cost of reduced sparsity. We further evaluate

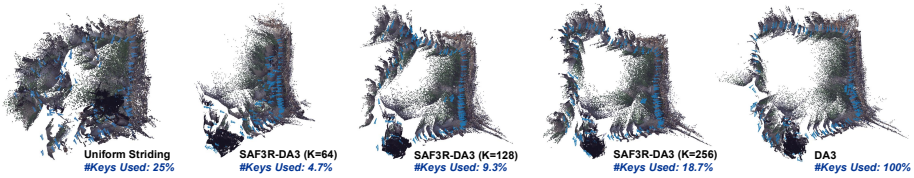

Fig. 8: Qualitative comparison across different values of K for the DINO Top-K kernel on the Oxford Spires dataset.

DA3 on a challenging large-scale outdoor scene with loop closures from the Oxford Spires dataset [27]. As shown in Fig. 8, increasing K progressively improves trajectory accuracy, whereas uniform striding performs poorly in this setting. Overall, the choice of K for DINO kernels reflects a trade-off between robustness and efficiency, and the optimal value may vary across scenes.

For future improvements to the DINO Top-K kernel, we identify two promising directions: (1) more accurate Top-K estimation by deriving indices directly from the first correspondence head instead of DINO features; (2) combining Top-K with uniform striding to reduce missing regions in challenging scenes.

6 Conclusions and Limitations

We present SAF3R, a training-free dynamic sparse attention framework for efficient inference in feed-forward 3D reconstruction (F3R) transformers. Our head-wise analysis reveals that global attention in F3R models exhibits strong heterogeneity and input-dependent sparsity. By exploiting these properties, SAF3R achieves significant end-to-end speedup while preserving reconstruction and pose estimation quality. We hope our findings provide new insights of global attention in F3R transformers and inspire future research on efficient F3R architectures.

Limitations. Our method primarily targets key-value sparsity in global attention, and further acceleration could be achieved by exploiting sparsity in other components, query-side redundancy, or integration with token merging. In addition, the DINO Top-K kernel may be less robust in challenging scenes where DINO features do not reliably localize correspondence regions, requiring larger K and thus reducing sparsity.

Acknowledgements

This work was supported in part by the National Science Foundation under Grants CNS-2328972, CCF-2324937, CNS-2122320, and CNS-2133267. This work was also supported in part by the TetraMem Inc. Research Award. We acknowledge the computational resources provided by the Pittsburgh Supercomputing Center and the National Center for Supercomputing Applications.

References

1. Aanæs, H., Jensen, R.R., Vogiatzis, G., Tola, E., Dahl, A.B.: Large-scale data for multiple-view stereopsis. IJCV **120**(2), 153–168 (2016)
2. Ansel, J., Yang, E., He, H., Gimelshein, N., Jain, A., Voznesensky, M., Bao, B., Bell, P., Berard, D., Burovski, E., et al.: Pytorch 2: Faster machine learning through dynamic python bytecode transformation and graph compilation. In: Proceedings of the 29th ACM international conference on architectural support for programming languages and operating systems, volume 2. pp. 929–947 (2024)
3. Bolya, D., Fu, C.Y., Dai, X., Zhang, P., Feichtenhofer, C., Hoffman, J.: Token merging: Your vit but faster. arXiv preprint arXiv:2210.09461 (2022)
4. Bolya, D., Hoffman, J.: Token merging for fast stable diffusion. In: CVPRW. pp. 4599–4603 (2023)
5. Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., Joulin, A.: Emerging properties in self-supervised vision transformers. In: ICCV. pp. 9650–9660 (2021)
6. Chen, Y., Qiu, Y., Li, R., Agha, A., Omidshafiei, S., Patrikar, J., Scherer, S.: Co-me: Confidence-guided token merging for visual geometric transformers. arXiv preprint arXiv:2511.14751 (2025)
7. Dai, A., Chang, A.X., Savva, M., Halber, M., Funkhouser, T., Nießner, M.: Scannet: Richly-annotated 3d reconstructions of indoor scenes. In: CVPR. pp. 5828–5839 (2017)
8. Darcet, T., Oquab, M., Mairal, J., Bojanowski, P.: Vision transformers need registers. arXiv preprint arXiv:2309.16588 (2023)
9. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020)
10. Furukawa, Y., Hernández, C.: Multi-view stereo: A tutorial. Foundations and Trends in Computer Graphics and Vision **9**(1-2), 1–148 (2015)
11. Jiang, H., Li, Y., Zhang, C., Wu, Q., Luo, X., Ahn, S., Han, Z., Abdi, A.H., Li, D., Lin, C.Y., et al.: Minference 1.0: Accelerating pre-filling for long-context llms via dynamic sparse attention. NeurIPS **37**, 52481–52515 (2024)
12. Keetha, N., Müller, N., Schönberger, J., Porzi, L., Zhang, Y., Fischer, T., Knapitsch, A., Zauss, D., Weber, E., Antunes, N., et al.: Mapanything: Universal feed-forward metric 3d reconstruction. arXiv preprint arXiv:2509.13414 (2025)
13. Lai, X., Lu, J., Luo, Y., Ma, Y., Zhou, X.: Flexprefill: A context-aware sparse attention mechanism for efficient long-sequence inference. arXiv preprint arXiv:2502.20766 (2025)
14. Leroy, V., Cabon, Y., Revaud, J.: Grounding image matching in 3d with mast3r. In: ECCV. pp. 71–91. Springer (2024)
15. Lin, H., Chen, S., Liew, J., Chen, D.Y., Li, Z., Shi, G., Feng, J., Kang, B.: Depth anything 3: Recovering the visual space from any views. arXiv preprint arXiv:2511.10647 (2025)
16. Mur-Artal, R., Tardós, J.D.: Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. IEEE Transactions on Robotics **33**(5), 1255–1262 (2017)
17. Oquab, M., Darcet, T., Moutakanni, T., Vo, H., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., et al.: Dinov2: Learning robust visual features without supervision. arXiv preprint arXiv:2304.07193 (2023)

18. Reizenstein, J., Shapovalov, R., Henzler, P., Sbordone, L., Labatut, P., Novotny, D.: Common objects in 3d: Large-scale learning and evaluation of real-life 3d category reconstruction. In: ICCV. pp. 10901–10911 (2021)
19. Ren, W., Tan, X., Han, K.: Speed3r: Sparse feed-forward 3d reconstruction models. arXiv preprint arXiv:2603.08055 (2026)
20. Schonberger, J.L., Frahm, J.M.: Structure-from-motion revisited. In: CVPR. pp. 4104–4113 (2016)
21. Schönberger, J.L., Zheng, E., Frahm, J.M., Pollefeys, M.: Pixelwise view selection for unstructured multi-view stereo. In: ECCV. pp. 501–518. Springer (2016)
22. Schops, T., Schonberger, J.L., Galliani, S., Sattler, T., Schindler, K., Pollefeys, M., Geiger, A.: A multi-view stereo benchmark with high-resolution images and multi-camera videos. In: CVPR. pp. 3260–3269 (2017)
23. Shen, Y., Zhang, Z., Qu, Y., Zheng, X., Ji, J., Zhang, S., Cao, L.: Fastvggt: Training-free acceleration of visual geometry transformer. arXiv preprint arXiv:2509.02560 (2025)
24. Shotton, J., Glocker, B., Zach, C., Izadi, S., Criminisi, A., Fitzgibbon, A.: Scene coordinate regression forests for camera relocalization in rgb-d images. In: CVPR. pp. 2930–2937 (2013)
25. Shu, Z., Lin, C., Xie, T., Yin, W., Li, B., Pu, Z., Li, W., Yao, Y., Cao, X., Guo, X., et al.: Litevggt: Boosting vanilla vggt via geometry-aware cached token merging. arXiv preprint arXiv:2512.04939 (2025)
26. Sun, X., Zhu, Z., Lou, Z., Yang, B., Tang, J., Zhang, L., Wang, H., Zhang, J.: Avggt: Rethinking global attention for accelerating vggt. arXiv preprint arXiv:2512.02541 (2025)
27. Tao, Y., Muñoz-Bañón, M.Á., Zhang, L., Wang, J., Fu, L.F.T., Fallon, M.: The oxford spires dataset: Benchmarking large-scale lidar-visual localisation, reconstruction and radiance field methods. The International Journal of Robotics Research **45**(6), 839–857 (2026)
28. Tillet, P., Kung, H.T., Cox, D.: Triton: an intermediate language and compiler for tiled neural network computations. In: Proceedings of the 3rd ACM SIGPLAN International Workshop on Machine Learning and Programming Languages. pp. 10–19 (2019)
29. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. NeurIPS **30** (2017)
30. Wang, C.S.B., Schmidt, C., Piekenbrinck, J., Leibe, B.: Faster vggt with block-sparse global attention. arXiv preprint arXiv:2509.07120 (2025)
31. Wang, J., Chen, M., Karaev, N., Vedaldi, A., Rupprecht, C., Novotny, D.: Vggt: Visual geometry grounded transformer. In: CVPR. pp. 5294–5306 (2025)
32. Wang, J., Karaev, N., Rupprecht, C., Novotny, D.: Vggsfm: Visual geometry grounded deep structure from motion. In: CVPR. pp. 21686–21697 (2024)
33. Wang, Q., Zhang, Y., Holynski, A., Efros, A.A., Kanazawa, A.: Continuous 3d perception model with persistent state. In: CVPR. pp. 10510–10522 (2025)
34. Wang, S., Leroy, V., Cabon, Y., Chidlovskii, B., Revaud, J.: Dust3r: Geometric 3d vision made easy. In: CVPR. pp. 20697–20709 (2024)
35. Wang, W., Meiner, L., Shubham, R., De La Parra, C., Kumar, A.: Httm: Head-wise temporal token merging for faster vggt. arXiv preprint arXiv:2511.21317 (2025)
36. Wang, Y., Zhou, J., Zhu, H., Chang, W., Zhou, Y., Li, Z., Chen, J., Pang, J., Shen, C., He, T.: π^3: Permutation-Equivariant Visual Geometry Learning. arXiv preprint arXiv:2507.13347 (2025)
37. Wang, Z., Xu, D.: Flashvggt: Efficient and scalable visual geometry transformers with compressed descriptor attention. arXiv preprint arXiv:2512.01540 (2025)

38. Xi, H., Yang, S., Zhao, Y., Xu, C., Li, M., Li, X., Lin, Y., Cai, H., Zhang, J., Li, D., et al.: Sparse videogen: Accelerating video diffusion transformers with spatial-temporal sparsity. arXiv preprint arXiv:2502.01776 (2025)
39. Xiao, G., Tang, J., Zuo, J., Guo, J., Yang, S., Tang, H., Fu, Y., Han, S.: Duoattention: Efficient long-context llm inference with retrieval and streaming heads. arXiv preprint arXiv:2410.10819 (2024)
40. Xiao, G., Tian, Y., Chen, B., Han, S., Lewis, M.: Efficient streaming language models with attention sinks. arXiv preprint arXiv:2309.17453 (2023)
41. Xu, R., Xiao, G., Huang, H., Guo, J., Han, S.: Xattention: Block sparse attention with antidiagonal scoring. In: ICML (2025)
42. Yang, J., Sax, A., Liang, K.J., Henaff, M., Tang, H., Cao, A., Chai, J., Meier, F., Feiszli, M.: Fast3r: Towards 3d reconstruction of 1000+ images in one forward pass. In: CVPR. pp. 21924–21935 (2025)
43. Yang, L., Kang, B., Huang, Z., Xu, X., Feng, J., Zhao, H.: Depth anything: Unleashing the power of large-scale unlabeled data. In: CVPR. pp. 10371–10381 (2024)
44. Yang, S., Xi, H., Zhao, Y., Li, M., Zhang, J., Cai, H., Lin, Y., Li, X., Xu, C., Peng, K., et al.: Sparse videogen2: Accelerate video generation with sparse attention via semantic-aware permutation. arXiv preprint arXiv:2505.18875 (2025)
45. Yeshwanth, C., Liu, Y.C., Nießner, M., Dai, A.: Scannet++: A high-fidelity dataset of 3d indoor scenes. In: ICCV. pp. 12–22 (2023)
46. Yuan, J., Gao, H., Dai, D., Luo, J., Zhao, L., Zhang, Z., Xie, Z., Wei, Y., Wang, L., Xiao, Z., et al.: Native sparse attention: Hardware-aligned and natively trainable sparse attention. In: Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 23078–23097 (2025)
47. Zhang, J., Xiang, C., Huang, H., Wei, J., Xi, H., Zhu, J., Chen, J.: Spargeattn: Accurate sparse attention accelerating any model inference. In: ICML (2025)
48. Zhang, P., Chen, Y., Su, R., Ding, H., Stoica, I., Liu, Z., Zhang, H.: Fast video generation with sliding tile attention. arXiv preprint arXiv:2502.04507 (2025)
49. Zhou, T., Tucker, R., Flynn, J., Fyffe, G., Snavely, N.: Stereo magnification: Learning view synthesis using multiplane images. arXiv preprint arXiv:1805.09817 (2018)