Good Token Hunting: A Hitchhiker's Guide to Token Selection for Visual Geometry Transformers

Shuhong Zheng[1,2] **Michael Oechsle**[3] **Erik Sandström**[3]
Marie-Julie Rakotosaona[3] **Federico Tombari**[3,4†] **Igor Gilitschenski**[1,2†]
[1]University of Toronto [2]Vector Institute [3]Google [4]Technical University of Munich
{shuhong, gilitschenski}@cs.toronto.edu
{michaeloechsle, sandstrom, mrakotosaona, tombari}@google.com

Abstract

Visual geometry transformers have become powerful architectures for multi-view 3D reconstruction, enabling joint prediction of multiple 3D attributes in a feed-forward manner. However, their computational cost grows quadratically with the input sequence length due to the global attention layers inside these models. This limits both their scalability and efficiency. In this work, we address this challenge with a simple yet general strategy: restricting the number of key/value tokens that each query interacts with during global attention. To achieve effective token selection, we introduce a two-stage framework. First, an inter-frame selection step operates at the frame level to identify frames that should be preserved. Second, an intra-frame selection step further discards more redundant tokens within the selected frames. Our analysis highlights the advantage of a diversity-based strategy for inter-frame selection, which ensures broad coverage of the scene. For intra-frame selection, we show that layer-aware sparsification is necessary, with the selection process guided by the entropy of the global attention pattern. Our approach offers a superior speed-accuracy trade-off compared to existing solutions. Extensive experiments show that it accelerates visual geometry transformers by over 85% for scenes with 500 images while maintaining, or even improving, baseline performance, which hints that how our token selection strategy can play a crucial role in future applications of visual geometry transformers. Our project website is available at https://zsh2000.github.io/good-token-hunting.github.io/.

1 Introduction

Visual geometry transformers [41, 53, 85, 89] are models capable of predicting key 3D attributes (*e.g.*, camera parameters, point maps, depth maps) from multiple views of a scene in a single forward pass. Although these models serve as substantially faster solutions than previous alternatives [70], they still suffer from prohibitively long inference time when increasing the number of processed frames. This limitation stems from the global attention layers inside these models. While these global attention layers enable effective information aggregation across views, they also exhibit quadratic computational complexity $\mathcal{O}(N^2L^2)$ in the number of input frames N and per-frame tokens L. As a result, global attention becomes the dominant bottleneck, causing inference cost to grow rapidly with the number of input images, and ultimately constraining the efficiency of visual geometry transformers, as illustrated in Figure 1.

[†]Joint Advising

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

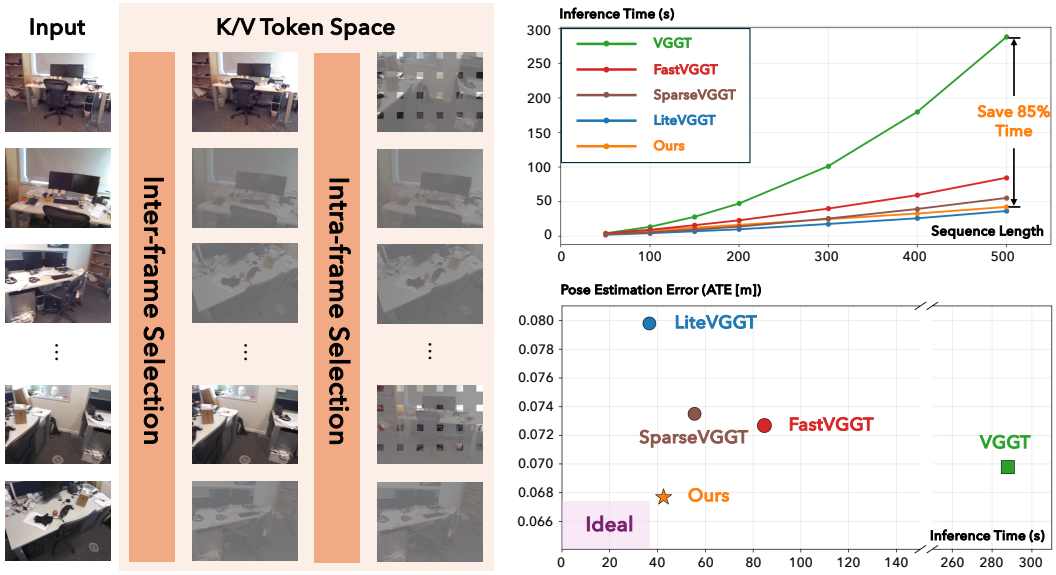

Figure 1: We accelerate visual geometry transformers via a two-stage hierarchical token selection scheme: inter-frame selection followed by intra-frame selection. Our training-free method scales near-linearly with the number of input frames, substantially improving the efficiency of visual geometry transformers with a comparable acceleration ratio with LiteVGGT [74], which requires costly full model training. Overall, our method achieves a superior trade-off between efficiency and accuracy.

To address this challenge in a principled and generalizable manner, we formulate our problem as follows: in the global attention layers of visual geometry transformers, *given a limited budget of key/value tokens with which each query can interact, how should these tokens be selected?* Our study, Good Token Hunting *(GoToHunt)*, investigates this question by exploring and analyzing various token selection strategies. Existing solutions [72] directly select tokens from the full set across all frames and require computationally heavy inspection of all tokens. In contrast, we leverage a two-stage hierarchical token selection scheme. The first stage performs *inter-frame selection* at the frame level, determining key/value tokens from which frames should be retained. This is a non-trivial task because several intuitive strategies, including similarity-based or activation-based criteria, incur significant performance degradation. Instead, inspired by keyframe-based SLAM [40, 49] systems, we propose selecting a collection of frames that are as diverse as possible to ensure broad scene coverage. Empirically, this diversity-driven approach proves to be an effective inter-frame selection strategy under tight token budgets, largely preserving the performance from base models while significantly reducing computational cost.

After completing token selection on the frame level, we perform *intra-frame selection* to further improve efficiency by discarding more key/value tokens within each selected frame. We first discover that uniformly downsampling across all global attention layers induces a non-negligible performance drop. To mitigate this issue, we conduct an analysis on the global attention patterns within each layer. We find that early layers exhibit heavily diluted attention, which is a phenomenon also found in language models [15, 44, 104, 113], while middle and late layers tend to display spiking values in the attention map. These observations motivate a layer-adaptive intra-frame selection strategy, in which different levels of token pruning are applied across different layers. In particular, in layers with highly activated tokens, we adopt more conservative strategies to avoid discarding important tokens before the actual attention scores are calculated. Combined with the preceding inter-frame selection stage, this two-stage hierarchical design, as demonstrated in Figure 1, substantially improves efficiency of visual geometry transformers. For example, on scenes with 500 input frames, our method reduces the inference time of the base VGGT [85] model by over 85%, while achieving a more favorable trade-off between inference speed and performance compared to existing acceleration approaches [72, 74, 82].

In summary, our work makes four contributions. **(1)** First, we cast the speedup of visual geometry transformers into a straightforward yet general formulation by constraining the number of key/value tokens each query interacts with in global attention layers. **(2)** Second, to solve this problem, we

introduce a novel hierarchical token selection strategy consisting of inter-frame and intra-frame selection for global attention layers. **(3)** Third, we provide a systematic exploration of token selection strategies, showing that diversity-based solutions are well-suited for inter-frame selection, while layer-adaptive strategies with different levels of token pruning is critical for intra-frame selection. These findings offer practical guidance for improving both efficiency and performance of visual geometry transformers. **(4)** Finally, comprehensive experimental results demonstrate that our training-free *GoToHunt* solution achieves superior trade-off between efficiency and performance for accelerating visual geometry transformers compared to existing methods, delivering competitive inference speed improvement with minimal performance compromise.

2 Related works

Feed-forward 3D reconstruction. Multi-view 3D reconstruction tasks, like Structure-from-Motion (SfM) and Multi-view Stereo (MVS), are traditionally solved using complex pipelines involving optimization [70]. While these methods achieve high accuracy under favorable conditions, they rely on iterative non-linear optimization steps like bundle adjustment [1]. Recent emergence of feed-forward 3D reconstruction models mark a fundamental departure from solving for geometry through optimization. DUSt3R [87] and its follow-up works [5, 22, 34, 48, 57, 80, 108, 112] pioneered this paradigm by predicting pairwise 3D point maps from image pairs using neural networks [92, 93]. More recently, *Visual Geometry Transformers* such as VGGT [85] further broadened this paradigm to jointly predict key 3D attributes like cameras, depth, or point maps from multiple images. This formulation inspired subsequent works [28, 31, 66, 83, 91], including π^3 [89], MapAnything [41], and Depth Anything 3 [53], which explore alternative architectural design choices. This line of work has already been applied in multiple areas [2, 12, 26, 51, 56, 60, 94, 96, 106]. For example, streaming reconstruction methods [25, 29, 42, 55, 58, 86, 101, 107] are realized through causal and autoregressive models [13, 45, 102, 117], preserved geometric context [6, 52, 68], recurrent memory [11, 54], and adaptive state updates [8, 21, 38, 115]. Dynamic scene reconstruction methods [36, 77, 84, 95, 99] leverage attention-derived motion cues [7, 19, 33, 71, 98], dynamic-aware aggregation and uncertainty modeling [24, 109, 116], dynamic point maps and temporal matching [32, 76, 111], and explicit scene flow and motion representations [39, 67]. Besides, there are also works focusing on human-centric reconstruction [9, 114], autonomous driving [35, 105, 118], visual relocalization [18, 20, 62, 100], and odometry [16, 64]. This breadth of applications demonstrates the growing importance and versatility of visual geometry transformers. The substantial computational cost when using a large number of input images is one of the main obstacles towards even broader impact for these models.

Efficiency improvement on visual geometry transformers. To address this challenge, a growing body of research [43, 47, 50, 61, 74, 88, 103] aims at improving efficiency to make visual geometry transformers practical at scale. For example, FastVGGT [72] introduced a training-free token merging scheme that preserves reference and salient tokens while merging the rest. Other approaches such as SparseVGGT [78] inspect the behavior of global attention and introduce specific attention calculation and token pruning mechanisms to speed up inference. AVGGT [78] replaces early global attention with frame attention and subsamples key/value tokens with diagonal preservation and mean filling. Compression-based approaches reduce the inference cost through low-bit compression, including quantized VGGT [27] and tail-aware quantization [65]. In contrast, methods [90] like LiteVGGT [74] and Speed3R [69] improve efficiency by retraining the model with additional priors or architectural constraints, so that the global attention layers operate on fewer tokens. Unlike these works, we adopt a training-free approach that allows for selecting a few key/value tokens within a limited budget that each query can interact with.

3 *GoToHunt*: Token selection for global attention

3.1 Preliminaries and problem formulation

Visual geometry transformers take in N images $\mathcal{I} = \{I_n\}_{i=1}^{N}$ capturing a scene as input, and predicts geometric properties for each frame such as camera pose $[\mathbf{R}_i|\mathbf{t}_i]$, point maps $\mathbf{P}_i$, *etc.*, depending on the model design. Specifically, each image is first patchified into L spatial tokens, optionally concatenated with special tokens (*e.g.*, camera tokens). These tokens are then processed

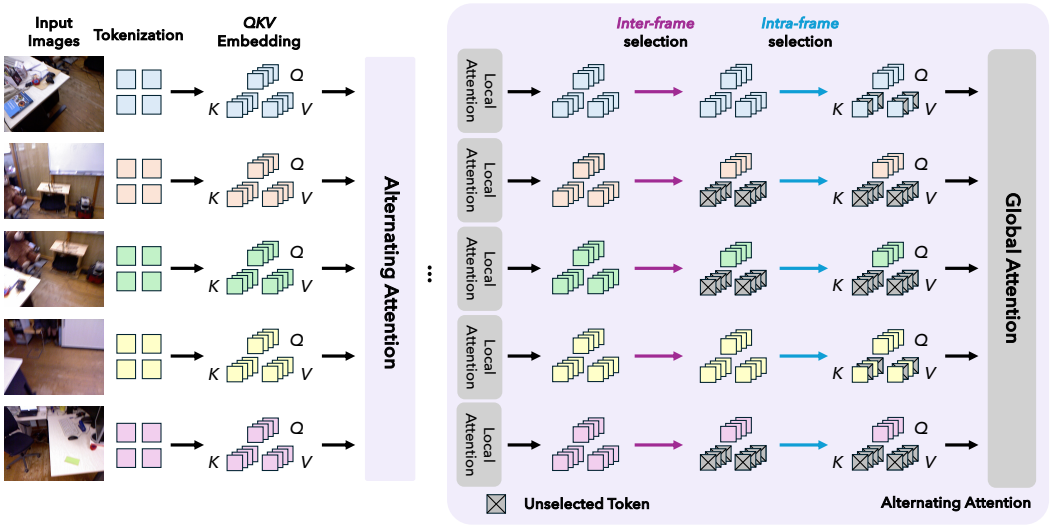

Figure 2: Pipeline of *GoToHunt*. Token selection is performed in the *K/V* space prior to the global attention layers, to determine which key/value tokens each query token interacts with. Our approach follows a two-stage hierarchical design: *inter-frame selection* first conducts frame-level selection, while *intra-frame selection* subsequently discard more tokens within each selected frame.

by a stack of frame-wise attention layers, which operate independently within each frame, and global attention layers, which jointly operate across all tokens from all frames. After cross-view information aggregation, dedicated task-specific heads decode each geometric property from the processed representations.

Computational bottleneck. As also illustrated in prior work [72, 74], the inference efficiency of visual geometry transformers is primarily constrained by the global attention layers, which compute attention over the entire set of $N \times L$ tokens (N is the number of input frames, and L is the number of tokens per frame), along with any additional special tokens. This results in a quadratic computational complexity of $\mathcal{O}(N^2L^2)$, which is the central bottleneck addressed in this work.

Problem formulation. To address this challenge in a general and principled manner, we adopt the following simple formulation: restricting the number of key/value tokens that each query attends to within each global attention layer. Rather than directly selecting tokens from the entire set across all frames, which is inefficient and suboptimal as it requires computationally heavy scan on all tokens, we employ a hierarchical selection strategy. We first perform inter-frame selection (Section 3.2) to select a set of frames. Then, we apply intra-frame selection (Section 3.3) within each selected frame to further discard more tokens. This two-stage design enables efficient token selection under the budget constraint while preserving essential information.

Preliminary experiment setting. In Sections 3.2 and 3.3, we conduct preliminary experiments to systematically analyze inter-frame and intra-frame token selection strategies. We evaluate camera pose estimation on the 7-Scenes [73] dataset. We sample every 2 frames of the image sequences, resulting in 500 frames per scene (with the exception of two scenes containing 250 frames). We follow previous works [72, 89] and adopt the metrics of Absolute Trajectory Error (ATE), Relative Pose Error in rotation (RPE-rot) and translation (RPE-trans).

3.2 Inter-frame selection: Hunting for good frames

Intuitive strategies. The first stage in token selection is inter-frame selection, to determine what frames to keep for further processing. First, we evaluate several intuitive strategies: **(1)** selecting temporally adjacent frames (only applicable to ordered sequences); **(2)** selecting frames based on co-visibility, with variants of **(2a)** selecting frames that are *most* co-visible with the current frame; **(2b)** selecting frames that are *least* co-visible with the current frame; **(3)** selecting frames based on attention activation, split into **(3a)** selecting based on the *maximum* attention score and **(3b)** selecting based on the *mean* attention score; and **(4)** selecting frames following existing work

Table 1: Comparison of intuitive inter-frame selection strategies on 7-Scenes [73] for camera pose estimation. The budget restricts each query to attend to key/value tokens from $K = 25$ frames.

Strategy	ATE (↓)	RPE-rot (↓)	RPE-trans (↓)
(1) Temporal proximity			
Nearest	0.7588	1.8485	0.0563
(2) Co-visibility-based selection			
(2a) High co-visibility	0.3813	2.9934	0.1197
(2b) Low co-visibility	0.1840	2.4761	0.1038
(3) Attention-based selection			
(3a) Max pooling	0.3879	7.2494	0.1257
(3b) Mean pooling	0.3627	7.2988	0.0988
(4) Selection strategy from existing work *(ordered input)*			
VGGT-SLAM	0.0741	0.5958	0.0203
Diversity-based (Ours)	**0.0676**	**0.4421**	**0.0167**
VGGT *(Base Model)*	0.0698	0.4953	0.0178

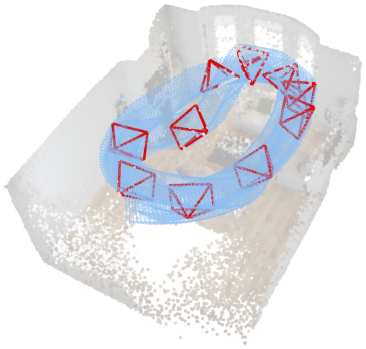

Figure 3: Illustration of inter-frame selection with $K = 10$: the selected views (red) form a diverse subset of the full set of views (blue), maximizing view-space coverage under a limited budget.

VGGT-SLAM [59]. Specifically, as VGGT-SLAM requires sequentially ordered input frames, it is evaluated on ordered sequences while the other settings do not necessarily require. For co-visibility approximation, we utilize the place recognition model [4] to extract features for each input image. The similarity between features serves as the proxy for the frame overlap, indicating the co-visibility between image pairs. For this preliminary analysis, we set the budget of selected frames to be $K = 25$ from the 7-Scenes sequences of 250/500 frames, meaning that we allow each query to only interact with key/value tokens from 25 frames in the global attention layers. At this level of sparsification, maintaining decent performance after frame selection is non-trivial. As reported in Table 1, the intuitive strategies from **(1)** to **(3)** all lead to substantial performance degradation. For **(4)**, it selects frames that have disparity only with the last selected keyframe, ignoring other already selected frames in the set. In contrast, our strategy optimizes for a diverse set of frames where each frame in the selected set is targeted at capturing a different portion of the scene from any other selected frame. Also, we point out that the purpose of frame selection in our paper is fundamentally different from the VGGT-SLAM paper, as in VGGT-SLAM, keyframe selection is for obtaining submaps that run VGGT independently, after which alignment between submaps are performed. In our approach, frame selection is used to determine key/value tokens from which frames need to be preserved, for accelerating the global attention layers inside VGGT, and all frames are still processed by VGGT in a single pass. Therefore, the metrics reported in Table 1 differ from the VGGT-SLAM paper.

Diversity-based frame selection. In contrast to the above strategies, our intuition is to select a set of frames, within a given budget, that can maximize view-space coverage. Formally, given N images with d-dimensional features $\{f_i\}_{i=1}^N$ extracted by the aforementioned place recognition model, we define the *cosine distance* between two images as

$$d(i,j) \;=\; 1 - \frac{\langle f_i, f_j \rangle}{\|f_i\|_2 \, \|f_j\|_2}. \tag{1}$$

Under a budget that allows each query to attend to tokens from K frames, we seek the subset $S^\star \subseteq \{1, \ldots, N\}$ with $|S^\star| = K$ that minimizes the largest distance from any frame to its nearest selected frame:

$$S^\star \;=\; \underset{S \subseteq \{1,\ldots,N\}, |S|=K}{\arg\min} \; \max_{i \in \{1,\ldots,N\}} \; \min_{j \in S} \; d(i,j). \tag{2}$$

Since Equation 2 is the classical NP-hard "K-center" objective, we adopt the similar greedy farthest point sampling (FPS) heuristic [30], widely used in point cloud processing, which iteratively selects the frame farthest from the current selected set. For details, we refer to Algorithm A in the Appendix.

From the results in Table 1, we can observe that our inter-frame selection strategy greatly outperforms the intuitive alternatives. These selected frames serve as "anchors", as illustrated in Figure 3, providing broad view-space coverage of the scene with a set of views within a limited budget. Moreover, these "anchors" supporting the whole scene are expected to be consistent across different

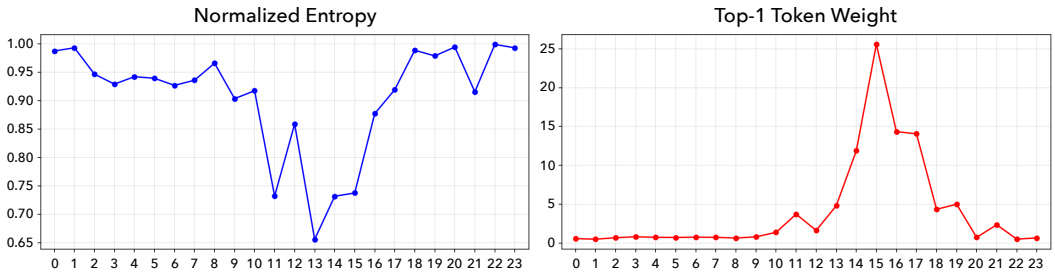

Figure 4: Attention pattern analysis of global attention layers (0-23) within VGGT [85]. Early layers show diluted, near-uniform attention distributions, whereas middle layers have spiking values.

queries, suggesting that a common set of reference views across all tokens is beneficial for cross-view representation processing within visual geometry transformers.

3.3 Intra-frame token selection: Preserving necessary tokens

Performance drop with intra-frame downsampling. Having determined which frames to retain, we turn to the second stage of token selection: identifying which tokens within each selected frame can be further discarded. Existing work [78] suggests that we can apply intra-frame downsampling within all global attention layers by subsampling token maps. Concretely, tokens are downsampled by a factor of σ along both the height and width dimensions, reducing a feature map with the original size $h \times w$ to $\lfloor \frac{h}{\sigma} \times \frac{w}{\sigma} \rfloor$. Following their approach, we perform downsampling across all global attention layers. However, we observe a noticeable performance drop, as reported in Table 2. Even a modest downsampling factor of $\sigma = 2$ leads to measurable performance degradation.

Table 2: Experimental results with the uniform intra-frame token selection strategy across all global attention layers following AVGGT [74].

K	σ	ATE (↓)	RPE-rot (↓)	RPE-trans (↓)
10	2	0.1031	1.4347	0.0295
10	3	0.2414	3.7025	0.0564
25	2	0.0831	0.6630	0.0213
25	3	0.1393	1.8891	0.0434
VGGT *(Base Model)*		0.0698	0.4953	0.0178

Attention pattern analysis. To understand the reason behind the performance degradation after intra-frame downsampling, we inspect the attention patterns within the global attention layers. Specifically, we report two statistics in Figure 4: *normalized entropy* $\mathcal{H}_{\text{norm}} \in [0, 1]$ and *top-1 token weight*, computed over a set of sampled query tokens and attention heads for each layer. The *normalized entropy* is formalized as

$$\mathcal{H}_{\text{norm}} = \frac{\sum_{0 \leqslant h < H, 0 \leqslant q < Q} \mathcal{H}(h, q)}{H \cdot Q \cdot \mathcal{H}_{\max}}, \tag{3}$$

where $\mathcal{H}_{\max} = \log(NL)$ is the maximum possible entropy over all key tokens, with N being the number of frames and L the number of tokens per frame. $\mathcal{H}(h, q)$ represents the entropy of the attention scores on attention head h and query q. $H = 4$ and $Q = 50$ are the number of sampled attention heads and query tokens for calculating these statistics.

As shown in Figure 4, early global attention layers exhibit a diluted, near-uniform attention pattern, whereas middle and later layers showcase a sharp attention pattern with spiking attention values. This observation suggests that, if the token downsampling in the middle and late layers discard tokens that are highly activated, their attention pattern will be severely disrupted, resulting in a performance degradation. This hypothesis is also supported by the comparison between the *Standard* and *Activation* strategies in Table 3, where the *Activation* preserves the same fraction of tokens ($\frac{1}{4}$

Table 3: Performance analysis on different intra-frame strategies applied on different sets of VGGT [85] layers, with $K = 25$.

σ	Strategy	Layers	ATE (↓)	RPE rot (↓)	RPE trans (↓)
2	*Standard*	0-8	0.0676	0.4427	0.0168
2	*Standard*	9-16	0.0792	0.9539	0.0239
2	*Activation*	9-16	0.0687	0.4664	0.0172
2	*Standard*	17-23	0.0715	0.4463	0.0168
3	*Standard*	0-8	0.0679	0.4486	0.0168
3	*Standard*	9-16	0.1234	1.6722	0.0416
3	*Activation*	9-16	0.0711	0.9163	0.0207
3	*Standard*	17-23	0.0743	0.4527	0.0172
VGGT [85] *(Base Model)*			0.0698	0.4953	0.0178

for $\sigma = 2$, $\frac{1}{9}$ for $\sigma = 3$) by selecting tokens with the highest attention activations, while *Standard* uniformly drops tokens in both height and width dimensions. The substantially reduced performance compromise of *Activation* for the middle layers indicates that token selection in layers with spiking attention values needs to be carefully designed. However, since identifying highly activated tokens requires computing attention scores in advance, which is time-consuming, *Activation* can only serve as a validation for our hypothesis, instead of a practical and efficient solution. In contrast, since the attention is diluted in the early layers without highly activated tokens, more aggressive intra-frame downsampling can be safely applied in these layers while still largely preserving the performance, as supported by the results in Table 3.

Layer-adaptive intra-frame strategy. The attention patterns in Figure 4 reveal that early layers of visual geometry transformers tend to have diluted attention patterns, where we can safely perform intra-frame downsampling without concerning about dropping highly activated tokens. Furthermore, we observe that the very first few layers have normalized entropy values close to 1, indicating that global attention in these layers can barely function for cross-view interaction. Following [78], we can replace these global attention layers with local attention operating within each frame to further save compute. Therefore, to formalize this design, we introduce two thresholds, l_{local} and l_{sample}, to determine the intra-frame strategies applied to each layer. For layers with index $l < l_{\text{local}}$, we replace global attention with local attention, which is the more aggressive intra-frame strategy to speed up the inference. For layers with index $l_{\text{local}} \leqslant l < l_{\text{sample}}$, we apply intra-frame downsampling with a selected factor. This layer-adaptive strategy balances efficiency and accuracy by aligning the levels of token pruning with the underlying attention characteristics for different global attention layers.

4 Experiments

4.1 Experimental setup

Implementation details. We choose two representative visual geometry transformers VGGT [85] and π^3 [89] as base models for evaluation. For comparisons with other methods in Section 4.2, we choose $K = 25$ and $\sigma \in \{2, 3\}$ for a relatively fixed budget of selected tokens. In the analysis in Section 4.3, we further show the model performance under different budgets. Unless otherwise specified, we set the layer thresholds to $l_{\text{local}} = 2$ and $l_{\text{sample}} = 9$, but also demonstrate in Section 4.3 that the performance is robust to these thresholds. All experiments are conducted on a single NVIDIA L40S GPU with 48GB CUDA memory.

Tasks, metrics, and datasets. Beyond the camera pose estimation task already introduced in Section 3.1, we also evaluate our method on 3D point cloud reconstruction and video depth estimation. Following previous works [89], we adopt the mean and median values of Accuracy (Acc), Completion (Comp), and Normal Consistency (NC) as evaluation metrics for 3D reconstruction, and Absolute Relative Error (Abs Rel), Root Mean Squared Error (RMSE), Log RMSE, Squared Relative Error (Sq Rel), and prediction accuracy at the threshold of $\delta < 1.25$ for video depth estimation. Detailed explanations on the metrics of all three tasks can be referred in Section F in the Appendix. Experiments are conducted on a diverse set of benchmarks, including 7-Scenes [73], Neural RGB-D [3], TUM-Dynamics [75], and Bonn [63].

Baseline methods. We compare against the state-of-the-arts for accelerating visual geometry transformers, including FastVGGT [72], SparseVGGT [82], Co-Me [10], LiteVGGT [74], and Speed3R [69]. We follow the default sparsification settings adopted in these methods. For SparseVGGT, we report results with sparsity ratio (SR) of 50% and 75% using a CDF threshold of 0.9. Among these methods, LiteVGGT and Speed3R require full model retraining, typically taking several days on a multi-GPU setup (8 high-performance GPUs). Co-Me involves lightweight training under 1 hour on an NVIDIA A100 GPU. In contrast, FastVGGT, SparseVGGT, and our method are training-free. SparseVGGT can also be applied to π^3 (denoted as Sparse-π^3), whereas Speed3R is only available with the π^3 checkpoint.

4.2 Comparisons with existing methods

Camera pose estimation. We evaluate our method on 7-Scenes, Neural RGB-D, and TUM-Dynamics. Following [14], sequences in 7-Scenes are sampled every two frames, resulting in each scene containing 500 frames except for two scenes with 250 frames, while Neural RGB-D is sampled every

Table 4: Quantitative comparisons on camera pose estimation together with the 500-frame sequence inference time on a single NVIDIA L40S GPU. Best is **bold** and the second best is underlined, excluding the base model row.

Method	7-Scenes			Neural RGB-D			TUM-Dynamics			Efficiency
	ATE (↓)	RPE-rot (↓)	RPE-trans (↓)	ATE (↓)	RPE-rot (↓)	RPE-trans (↓)	ATE (↓)	RPE-rot (↓)	RPE-trans (↓)	500-frame Inference Time (↓)
VGGT [85] *(Base Model)*	0.0698	0.4953	0.0178	0.0374	0.2934	0.0186	0.0118	0.3083	0.0098	288.0s
FastVGGT [72]	0.0727	**0.4254**	**0.0159**	0.0377	0.1985	0.0168	0.0127	0.3154	0.0108	84.6s
SparseVGGT [82] (SR: 50%)	0.0723	0.4608	0.0167	0.0402	0.2946	0.0202	0.0125	0.3114	0.0102	80.3s
SparseVGGT [82] (SR: 75%)	0.0735	0.4583	0.0169	0.0462	0.2717	0.0192	0.0127	0.3120	0.0103	55.4s
Co-Me [10]	0.0870	0.8105	0.0340	0.0626	0.4567	0.0336	0.0156	0.3438	0.0146	84.2s
LiteVGGT [74]	0.0798	0.6888	0.0238	0.0531	0.3311	0.0247	0.0145	0.3250	0.0119	**36.5s**
***GoToHunt* (Ours)** ($\sigma = 2$)	**0.0673**	0.4471	0.0165	**0.0267**	**0.1794**	**0.0162**	**0.0115**	0.3087	**0.0101**	44.6s
***GoToHunt* (Ours)** ($\sigma = 3$)	0.0677	0.4495	0.0166	0.0270	0.2409	0.0176	0.0119	**0.3075**	0.0102	41.2s
π^3 [89] *(Base Model)*	0.0573	0.3389	0.0105	0.0251	0.1031	0.0098	0.0140	0.3073	0.0088	106.5s
Sparse-π^3 [82] (SR: 50%)	0.0580	**0.3369**	**0.0106**	0.0313	**0.1182**	**0.0115**	**0.0140**	**0.3068**	0.0090	76.4s
Sparse-π^3 [82] (SR: 75%)	0.0594	0.3387	0.0108	0.0478	0.1250	0.0124	0.0141	0.3094	0.0092	74.1s
Speed3R [69]	0.0591	0.3800	0.0133	0.0391	0.1735	0.0145	0.0193	0.3152	0.0103	**17.9s**
***GoToHunt* (Ours)** ($\sigma = 2$)	0.0579	0.3445	0.0113	**0.0292**	0.1190	0.0123	0.0142	0.3075	**0.0089**	26.4s
***GoToHunt* (Ours)** ($\sigma = 3$)	**0.0570**	0.3428	0.0112	**0.0292**	0.1192	0.0123	0.0144	0.3083	**0.0089**	26.1s

Table 5: Quantitative comparisons on point map estimation. Best is **bold** and the second best is underlined, excluding the base model row.

Method	7-Scenes						Neural RGB-D					
	Acc (↓)		Comp (↓)		NC (↑)		Acc (↓)		Comp (↓)		NC (↑)	
	Mean	Med.	Mean	Med.	Mean	Med.	Mean	Med.	Mean	Med.	Mean	Med.
VGGT [85] *(Base Model)*	0.0171	0.0038	0.0184	0.0043	0.5568	0.5851	0.0160	0.0099	0.0112	0.0028	0.7508	0.8917
FastVGGT [72]	0.0166	0.0042	**0.0182**	0.0034	0.5554	0.5830	0.0181	0.0131	0.0115	0.0031	0.7196	0.8640
SparseVGGT [82] (SR: 50%)	0.0172	0.0039	0.0191	0.0042	0.5563	0.5846	0.0160	0.0099	**0.0112**	**0.0027**	0.7384	0.8787
SparseVGGT [82] (SR: 75%)	0.0174	0.0040	0.0189	0.0042	0.5561	0.5842	0.0363	0.0258	0.0169	0.0041	0.6907	0.8353
Co-Me [10]	**0.0147**	0.0061	0.0234	0.0060	**0.5826**	**0.6271**	0.0167	0.0091	0.0115	0.0033	**0.7716**	**0.9104**
LiteVGGT [74]	0.0185	0.0059	0.0232	**0.0033**	0.5542	0.5815	0.0264	0.0154	0.0152	0.0030	0.6833	0.8009
***GoToHunt* (Ours)** ($\sigma = 2$)	0.0152	**0.0036**	0.0188	0.0043	0.5567	0.5850	0.0127	0.0075	**0.0112**	**0.0027**	0.7552	0.8946
***GoToHunt* (Ours)** ($\sigma = 3$)	0.0152	**0.0036**	0.0189	0.0043	0.5568	0.5854	**0.0126**	**0.0074**	0.0113	0.0028	0.7582	0.8973
π^3 [89] *(Base Model)*	0.0105	0.0034	0.0141	0.0054	0.5677	0.6027	0.0128	0.0057	0.0101	0.0024	0.7503	0.8806
Sparse-π^3 [82] (SR: 50%)	0.0109	0.0036	0.0142	0.0053	0.5664	0.6006	0.0150	0.0078	0.0112	**0.0027**	0.7361	0.8672
Sparse-π^3 [82] (SR: 75%)	0.0117	0.0038	0.0152	0.0055	0.5659	0.5997	0.0189	0.0100	0.0149	0.0040	0.7350	0.8663
Speed3R [69]	0.0120	0.0040	**0.0137**	**0.0044**	0.5661	0.6006	0.0208	0.0137	0.0149	0.0040	0.7256	0.8617
***GoToHunt* (Ours)** ($\sigma = 2$)	0.0105	0.0034	0.0142	0.0052	**0.5666**	**0.6009**	**0.0148**	**0.0074**	**0.0107**	**0.0027**	**0.7463**	**0.8781**
***GoToHunt* (Ours)** ($\sigma = 3$)	**0.0104**	**0.0033**	0.0139	0.0050	**0.5666**	0.6008	0.0149	0.0073	**0.0107**	**0.0027**	0.7461	0.8778

five frames. As shown in Table 4, our method achieves an overall superior performance compared to existing approaches across all datasets, both on the long-sequence 7-Scenes and Neural RGB-D and the standard TUM-Dynamics benchmark, in several cases even improving upon the base model. These results highlight the effectiveness of our proposed token selection strategy, particularly as a *training-free* approach, which can serve as a flexible plug-in that can be applied to general types of visual geometry transformers.

3D point cloud reconstruction. Unlike prior evaluations that often focus on sparse reconstruction from only 3-5 views per scene, we evaluate on dense multi-view settings to better assess the overall behavior on both performance and computational efficiency. Specifically, we follow the same protocol as in camera pose estimation and evaluate 3D point cloud reconstruction on 7-Scenes and Neural RGB-D with up to 500 frames per scene. As reported in Table 5, our method achieves superior overall performance compared to existing solutions, demonstrating its effectiveness and robustness in large-scale reconstruction scenarios.

Video depth estimation. To further evaluate performance on more tasks with long sequences, we conduct video depth estimation experiments on the full-length Bonn dataset [63], where sequences range from 332 to 895 frames per scene. We adopt π^3 as the base model so that the baseline method can also operate within the 48GB memory. As shown in Table 6, Sparse-π^3 encounters CUDA out-of-memory issues on these long sequences within 48GB memory, even with a high sparsification rate of 75%. In contrast, our method scales reliably to sequences exceeding 800 frames, and even outperforms the base model. Moreover, compared to Speed3R which requires costly model retraining, our training-free approach still achieves superior performance on most metrics.

Inference efficiency. We present the inference speed comparison across different methods in Table 7. We can observe that our approach shows near-linear inference time scaling with respect to the number of input images, resulting in an improved efficiency for long sequences, as we select a constant budget

Table 6: Quantitative comparisons on video depth estimation. The best accelerated model is highlighted in **bold**. Our method outperforms the base model even in terms of quality.

Method	Bonn				
	Abs Rel (↓)	Log RMSE (↓)	RMSE (↓)	Sq Rel (↓)	$\delta < 1.25$ (↑)
π^3 *(Base Model)*	0.0333	0.0746	0.1623	0.0123	0.9886
Sparse-π^3 [82] (SR: 50%)	*OOM*	*OOM*	*OOM*	*OOM*	*OOM*
Sparse-π^3 [82] (SR: 75%)	*OOM*	*OOM*	*OOM*	*OOM*	*OOM*
Speed3R [69]	0.0314	0.0680	0.1525	0.0103	**0.9909**
***GoToHunt* (Ours)** ($\sigma = 3$)	**0.0288**	**0.0668**	**0.1501**	**0.0100**	0.9893

Table 7: Inference time comparison on an NVIDIA L40S GPU. As above, we use $K = 25$ and $\sigma = 3$. The fastest result is in **bold** and the second fastest is underlined.

Input Sequence Length	100	200	300	400	500
VGGT [85] *(Base Model)*	13.6s	47.4s	101.3s	179.8s	288.0s
FastVGGT [72]	9.4s	23.1s	40.0s	59.5s	84.6s
Sparse-π^3 [82] (SR: 50%)	6.8s	18.3s	35.0s	56.2s	80.3s
Sparse-π^3 [82] (SR: 75%)	5.7s	14.0s	25.6s	39.5s	55.4s
LiteVGGT [74]	**4.5s**	**10.1s**	**17.8s**	**26.0s**	**36.5s**
Co-Me [10]	5.5s	16.4s	32.2s	53.3s	84.2s
***GoToHunt* (Ours)**	7.8s	15.9s	23.8s	31.7s	41.2s

of key/value tokens for the global attention layers. Although our method remains slightly slower than LiteVGGT, which achieves superior efficiency from expensive model retraining, our method consistently has smaller performance compromise. Combined with the comparisons on various tasks above, it demonstrates that our method achieves an overall better trade-off between efficiency and performance compared to existing approaches.

4.3 Ablation study and sensitivity analysis

Beyond the ablations already presented in Sections 3.2 and 3.3 about the methodology design of both inter-frame and intra-frame strategies, we provide additional analysis below for the key parameters to offer practical guidance for applying our method.

Budget for inter-frame selection. In our default experimental setting, we use a fixed inter-frame budget of $K = 25$ for scenes containing hundreds of frames. A natural concern is that using as few as 25 frames would be inevitably struggle to cover the full scene. To investigate this, we vary the budget K allocated for inter-frame selection, and report the results in Table 8. We observe that when the budget increases to 40-60 frames, which is rougly 10% of the total frames, the performance gets further boosted compared to our default choice of $K = 25$. What is more interesting is that the performance does not monotonically increase with larger budgets. On one hand, this is expected as when K approaches the total number of input frames, the results should converge to the performance of the base model, which can be outperformed by many of our variants. On the other hand, it is still counterintuitive, as attending to more frames should, in principle, make the model visible to more information.

There are many factors that can possibly cause this counterintuitive phenomena to happen, below are two of our proposed potential reasons: ***(1) More selected frames indicate more retained tokens for global attention***, and this may potentially aggravate attention dilution if this issue exists. This is consistent with our observation and solution for intra-frame token selection. As illustrated in Figure 4, early layers show diluted, near-uniform attention patterns, and downsampling key/value tokens that suffer from dilution can help improve the performance (Table 3 when downsampling the heavily diluted 0-8 layers). Similarly, if the tokens in the original VGGT suffer from a certain level of attention dilution, then the inter-frame downsampling is also alleviating the dilution issue, which accounts for the non-monotonic behavior of the inter-frame budget K. ***(2) The base models VGGT and π^3 are trained with the maximum sequence length of*** $l = 24$, and demonstrate the generalization capability to longer sequences containing hundreds of frames. However, it is possible that the generalization capability can gradually decrease when the number of input frames grows, and the bottleneck for the generalization capability lies in the global attention layer that we are optimizing. Therefore, when the number of frames grows, on one hand, the available multi-view information increases for cross-view information aggregation; while on the other hand, the generalization capability may gradually diminish as the number of frames increases. These two factors may ultimately reach a "sweet spot" at K being around 40-60, as demonstrated from the results in Table 8, which causes the non-monotonic behavior of K.

Layer thresholds for intra-frame selection. In Section 3.3, we set the thresholds l_{local} and l_{sample} to determine the layer-adaptive strategies with different levels of token pruning across all the global attention layers. We vary these thresholds in Table 9 and find that the performance remains stable across a broad range of configurations, which is consistent with the observed global attention patterns

Table 8: Analysis on the impact of the inter-frame selection budget K with $\sigma = 3$ for the pose estimation task on 7-Scenes. Increasing K initially improves performance, indicating that a moderate expansion of the selected frame set enhances scene coverage. However, the trend is not monotonic: beyond a certain range, further increasing K occasionally leads to an eventual performance degradation, with the performance approaching the results of the full model. Inference time is measured on an NVIDIA L40S GPU with 500 frames.

K	ATE (↓)	RPE-rot (↓)	RPE-trans (↓)	Inference Time (↓)
VGGT *(Base Model)*	0.0698	0.4953	0.0178	288.0s
10	0.0722	0.7614	0.0198	32.3s
25	0.0677	0.4495	0.0166	41.2s
40	0.0674	0.4204	0.0155	51.9s
60	0.0677	0.4203	0.0153	63.7s
80	0.0684	0.4211	0.0152	77.8s
100	0.0685	0.4229	0.0153	89.3s
π^3 *(Base Model)*	0.0573	0.3389	0.0105	110.1s
10	0.0575	0.3652	0.0126	21.9s
25	0.0570	0.3428	0.0112	26.1s
40	0.0563	0.3384	0.0108	30.5s
60	0.0561	0.3379	0.0107	36.1s
80	0.0567	0.3348	0.0105	42.2s
100	0.0566	0.3359	0.0106	47.8s

Table 9: Sensitivity analysis on the layer partition thresholds l_{local} and l_{sample} for intra-frame selection with $K = 25$, with the *blue* row indicating our current parameter choice. Performance remains stable across a wide range of threshold choices, demonstrating that our method is robust to these hyperparameter choices, as long as they are consistent with the observed layer-wise attention patterns.

l_{local}	l_{sample}	ATE (↓)	RPE-rot (↓)	RPE-trans (↓)
π^3 *(Base Model)*		0.0573	0.3389	0.0105
1	8	0.0567	0.3425	0.0112
1	9	0.0568	0.3425	0.0112
1	10	0.0569	0.3406	0.0111
2	8	0.0569	0.3432	0.0112
2	9	0.0570	0.3428	0.0112
2	10	0.0570	0.3409	0.0112
3	8	0.0570	0.3415	0.0112
3	9	0.0571	0.3418	0.0112
3	10	0.0571	0.3396	0.0112
4	8	0.0577	0.3442	0.0113
4	9	0.0578	0.3446	0.0113
4	10	0.0578	0.3425	0.0113

for each layer. This indicates that our method is robust to these hyperparameter choices, which further enhances the soundness and reliability of our method.

5 Discussions

Although our approach is introduced as a training-free acceleration method, the analysis and findings in our paper point to a more fundamental observation: the improved performance after token selection indicates that current visual geometry transformers are not yet perfectly trained with the optimal architecture. Our solution thus provides guidelines for future research on how to improve the network design and training strategies of visual geometry transformers. For example, our study on inter-frame selection points toward the potential of routing-based mechanisms for attention layers inside these models, while our intra-frame analysis suggests that global attention layers at early stages which suffer from attention dilution may be skipped even in the training process.

Limitations. Our inter-frame selection strategy relies on features extracted by a place recognition model, which may become less reliable in challenging scenarios such as object-centric scenes or environments with highly symmetric and ambiguous structures. However, these settings are inherently difficult for visual geometry transformers in general, and remain challenging even for conventional optimization-based structure-from-motion and multi-view stereo pipelines. In addition, the capability of our approach may get bounded by the capacity of the underlying base model. While our method substantially improves efficiency through hierarchical token selection, it does not alter the overall inference paradigm of visual geometry transformers. Consequently, it still faces difficulties on extremely large-scale environments, such as kilometer-scale scenes considered in VGGT-Long [17], where additional techniques such as chunk-based processing or recurrent inference are required. Nevertheless, our method is orthogonal and complementary to these approaches, and can still potentially be integrated with them to further improve their efficiency.

6 Conclusions

We present *GoToHunt*, a study that formulates the efficiency improvement of visual geometry transformers as a token selection problem. To enable effective selection, we introduce a two-stage hierarchical framework consisting of inter-frame and intra-frame selection. Through comprehensive analysis, we show that diversity-based strategies are well-suited for inter-frame selection, while layer-adaptive strategies with different levels of token pruning are needed for intra-frame selection. Extensive experiments demonstrate that our method achieves a superior trade-off between inference

efficiency and reconstruction quality compared to existing approaches, even occasionally outperforming the base models. We believe that our training-free solution can serve as a general and easy-to-use algorithm for accelerating various visual geometry transformers.

References

[1] Sameer Agarwal, Noah Snavely, Steven M. Seitz, and Richard Szeliski. Bundle adjustment in the large. In *ECCV*, 2010. 3

[2] Vivek Alumootil and Tuan-Anh Vu. DePT3R: Joint dense point tracking and 3D reconstruction of dynamic scenes in a single forward pass. *arXiv preprint arXiv:2512.13122*, 2025. 3

[3] Dejan Azinović, Ricardo Martin-Brualla, Dan B. Goldman, Matthias Nießner, and Justus Thies. Neural RGB-D surface reconstruction. In *CVPR*, 2022. 7, 27, 28, 29, 30, 33

[4] Gabriele Berton and Carlo Masone. MegaLoc: One retrieval to place them all. In *CVPR Workshops*, 2025. 5, 24

[5] Yohann Cabon, Lucas Stoffl, Leonid Antsfeld, Gabriela Csurka, Boris Chidlovskii, Jerome Revaud, and Vincent Leroy. MUSt3R: Multi-view network for stereo 3D reconstruction. In *CVPR*, 2025. 3

[6] Lin-Zhuo Chen, Jian Gao, Yihang Chen, Ka Leong Cheng, Yipengjing Sun, Liangxiao Hu, Nan Xue, Xing Zhu, Yujun Shen, Yao Yao, and Yinghao Xu. Geometric context transformer for streaming 3D reconstruction. *arXiv preprint arXiv:2604.14141*, 2026. 3

[7] Xingyu Chen, Yue Chen, Yuliang Xiu, Andreas Geiger, and Anpei Chen. Easi3R: Estimating disentangled motion from DUSt3R without training. In *ICCV*, 2025. 3

[8] Xingyu Chen, Yue Chen, Yuliang Xiu, Andreas Geiger, and Anpei Chen. TTT3R: 3D reconstruction as test-time training. In *ICLR*, 2026. 3

[9] Yue Chen, Xingyu Chen, Yuxuan Xue, Anpei Chen, Yuliang Xiu, and Gerard Pons-Moll. Human3R: Everyone everywhere all at once. In *ICLR*, 2026. 3

[10] Yutian Chen, Yuheng Qiu, Ruogu Li, Ali Agha, Shayegan Omidshafiei, Jay Patrikar, and Sebastian Scherer. Co-Me: Confidence-guided token merging for visual geometric transformers. In *CVPR*, 2026. 7, 8, 9

[11] Zhuoguang Chen, Minghui Qin, Tianyuan Yuan, Zhe Liu, and Hang Zhao. LONG3R: Long sequence streaming 3D reconstruction. In *ICCV*, 2025. 3

[12] Ziyang Chen, Yansong Qu, You Shen, Xuan Cheng, and Liujuan Cao. StereoVGGT: A training-free visual geometry transformer for stereo vision. *arXiv preprint arXiv:2603.29368*, 2026. 3

[13] Chong Cheng, Xianda Chen, Tao Xie, Wei Yin, Weiqiang Ren, Qian Zhang, Xiaoyang Guo, and Hao Wang. LongStream: Long-sequence streaming autoregressive visual geometry. In *CVPR*, 2026. 3

[14] Leo Kaixuan Cheng, Abdus Shaikh, Ruofan Liang, Zhijie Wu, Yushi Guan, and Nandita Vijaykumar. MERG3R: A divide-and-conquer approach to large-scale neural visual geometry. In *CVPR*, 2026. 7

[15] Ta-Chung Chi, Ting-Han Fan, and Alexander Rudnicky. Attention alignment and flexible positional embeddings improve transformer length extrapolation. In *NAACL Findings*, 2024. 2

[16] Weichen Dai, Wenhan Su, Da Kong, Yuhang Ming, and Wanzeng Kong. Keyframe-based feed-forward visual odometry. *arXiv preprint arXiv:2601.16020*, 2026. 3

[17] Kai Deng, Zexin Ti, Jiawei Xu, Jian Yang, and Jin Xie. VGGT-Long: Chunk it, Loop it, Align it – Pushing VGGT's limits on kilometer-scale long RGB sequences. In *ICRA*, 2026. 10

[18] Tianchen Deng, Wenhua Wu, Kunzhen Wu, Guangming Wang, Siting Zhu, Shenghai Yuan, Xun Chen, Guole Shen, Zhe Liu, and Hesheng Wang. Reloc-VGGT: Visual re-localization with geometry grounded transformer. *arXiv preprint arXiv:2512.21883*, 2025. 3

[19] Tianye Ding, Yiming Xie, Yiqing Liang, Moitreya Chatterjee, Pedro Miraldo, and Huaizu Jiang. LASER: Layer-wise scale alignment for training-free streaming 4D reconstruction. In *CVPR*, 2026. 3

[20] Gergely Dinya, Péter Halász, András Lőrincz, Kristóf Karacs, and Anna Gelencsér-Horváth. Building temporally coherent 3D maps with VGGT for memory-efficient semantic SLAM. *arXiv preprint arXiv:2511.16282*, 2025. 3

[21] Jiacheng Dong, Huan Li, Sicheng Zhou, Wenhao Hu, Weili Xu, and Yan Wang. MeMix: Writing less, remembering more for streaming 3D reconstruction. *arXiv preprint arXiv:2603.15330*, 2026. 3

[22] Bardienus Pieter Duisterhof, Lojze Zust, Philippe Weinzaepfel, Vincent Leroy, Yohann Cabon, and Jerome Revaud. MASt3R-SfM: A fully-integrated solution for unconstrained structure-from-motion. In *3DV*, 2025. 3

[23] Sven Elflein, Ruilong Li, Sérgio Agostinho, Zan Gojcic, Laura Leal-Taixé, Qunjie Zhou, and Aljosa Osep. VGG-T^3: Offline feed-forward 3D reconstruction at scale. In *CVPR*, 2026. 34

[24] Juntong Fang, Zequn Chen, Weiqi Zhang, Donglin Di, Xuancheng Zhang, Chengmin Yang, and Yu-Shen Liu. MoRe: Motion-aware feed-forward 4D reconstruction transformer. In *CVPR*, 2026. 3

[25] Keyu Fang, Changchun Zhou, Yuzhe Fu, Hai Helen Li, and Yiran Chen. IncVGGT: Incremental VGGT for memory-bounded long-range 3D reconstruction. In *ICLR*, 2026. 3

[26] Xianze Fang, Jingnan Gao, Zhe Wang, Zhuo Chen, Xingyu Ren, Jiangjing Lyu, Qiao-Mu Ren, Zhonglei Yang, Xiaokang Yang, Yichao Yan, and Chengfei Lv. Dens3R: A foundation model for 3D geometry prediction. In *ICLR*, 2026. 3

[27] Weilun Feng, Haotong Qin, Mingqiang Wu, Chuanguang Yang, Yuqi Li, Xiangqi Li, Zhulin An, Libo Huang, Yulun Zhang, Michele Magno, and Yongjun Xu. Quantized visual geometry grounded transformer. In *ICLR*, 2026. 3

[28] Jingnan Gao, Zhe Wang, Xianze Fang, Xingyu Ren, Zhuo Chen, Shengqi Liu, Yuhao Cheng, Jiangjing Lyu, Xiaokang Yang, and Yichao Yan. MoRE: 3D visual geometry reconstruction meets mixture-of-experts. In *CVPR*, 2026. 3

[29] Anna Gelencsér-Horváth, Gergely Dinya, Dorka Boglárka Erős, Péter Halász, Islam Muhammad Muqsit, and Kristóf Karacs. SceneVGGT: VGGT-based online 3D semantic SLAM for indoor scene understanding and navigation. *arXiv preprint arXiv:2602.15899*, 2026. 3

[30] Teofilo F. Gonzalez. Clustering to minimize the maximum intercluster distance. *Theoretical Computer Science*, 38:293–306, 1985. 5

[31] Jisang Han, Sunghwan Hong, Jaewoo Jung, Wooseok Jang, Honggyu An, Qianqian Wang, Seungryong Kim, and Chen Feng. Emergent outlier view rejection in visual geometry grounded transformers. In *CVPR*, 2026. 3

[32] Zhuolin He, Jing Li, Guanghao Li, Xiaolei Chen, Jiacheng Tang, Siyang Zhang, Zhounan Jin, Feipeng Cai, Bin Li, Jian Pu, Jia Cai, and Xiangyang Xue. DynamicVGGT: Learning dynamic point maps for 4D scene reconstruction in autonomous driving. *arXiv preprint arXiv:2603.08254*, 2026. 3

[33] Yu Hu, Chong Cheng, Sicheng Yu, Xiaoyang Guo, and Hao Wang. VGGT4D: Mining motion cues in visual geometry transformers for 4D scene reconstruction. *arXiv preprint arXiv:2511.19971*, 2025. 3

[34] Wonbong Jang, Philippe Weinzaepfel, Vincent Leroy, Lourdes Agapito, and Jerome Revaud. Pow3R: Empowering unconstrained 3d reconstruction with camera and scene priors. In *CVPR*, 2025. 3

[35] Xiaosong Jia, Yanhao Liu, Junqi You, Renqiu Xia, Yu Hong, and Junchi Yan. DriveVGGT: Visual geometry transformer for autonomous driving. *arXiv preprint arXiv:2511.22264*, 2025. 3

[36] Zeren Jiang, Chuanxia Zheng, Iro Laina, Diane Larlus, and Andrea Vedaldi. Geo4D: Leveraging video generators for geometric 4D scene reconstruction. In *ICCV*, 2025. 3

[37] Haian Jin, Rundi Wu, Tianyuan Zhang, Ruiqi Gao, Jonathan T. Barron, Noah Snavely, and Aleksander Holynski. ZipMap: Linear-time stateful 3D reconstruction via test-time training. In *CVPR*, 2026. 34

[38] Seonghyun Jin and Jong Chul Ye. FILT3R: Latent state adaptive Kalman filter for streaming 3D reconstruction. *arXiv preprint arXiv:2603.18493*, 2026. 3

[39] Jay Karhade, Nikhil Keetha, Yuchen Zhang, Tanisha Gupta, Akash Sharma, Sebastian Scherer, and Deva Ramanan. Any4D: Unified feed-forward metric 4D reconstruction. In *CVPR*, 2026. 3

[40] Anton Kasyanov, Francis Engelmann, Jörg Stückler, and Bastian Leibe. Keyframe-based visual-inertial online SLAM with relocalization. In *IROS*, 2017. 2

[41] Nikhil Keetha, Norman Müller, Johannes Schönberger, Lorenzo Porzi, Yuchen Zhang, Tobias Fischer, Arno Knapitsch, Duncan Zauss, Ethan Weber, Nelson Antunes, Jonathon Luiten, Manuel Lopez-Antequera, Samuel Rota Bulò, Christian Richardt, Deva Ramanan, Sebastian Scherer, and Peter Kontschieder. MapAnything: Universal feed-forward metric 3D reconstruction. In *3DV*, 2026. 1, 3

[42] Ramil Khafizov, Artem Komarichev, Ruslan Rakhimov, Peter Wonka, and Evgeny Burnaev. G-CUT3R: Guided 3D reconstruction with camera and depth prior integration. *arXiv preprint arXiv:2508.11379*, 2025. 3

[43] Yongsung Kim, Wooseok Song, Jaihyun Lew, Hun Hwangbo, Jaehoon Lee, and Sungroh Yoon. HeSS: Head sensitivity score for sparsity redistribution in VGGT. In *CVPR*, 2026. 3

[44] Zhenglun Kong, Yize Li, Fanhu Zeng, Lei Xin, Shvat Messica, Xue Lin, Pu Zhao, Manolis Kellis, Hao Tang, and Marinka Zitnik. Token reduction should go beyond efficiency in generative models – From vision, language to multimodality. *arXiv preprint arXiv:2505.18227*, 2025. 2

[45] Yushi Lan, Yihang Luo, Fangzhou Hong, Shangchen Zhou, Honghua Chen, Zhaoyang Lyu, Shuai Yang, Bo Dai, Chen Change Loy, and Xingang Pan. STream3R: Scalable sequential 3D reconstruction with causal transformer. In *ICLR*, 2026. 3

[46] Brandon Leblanc and Charalambos Poullis. Distill3R: A pipeline for democratizing 3D foundation models on commodity hardware. *arXiv preprint arXiv:2602.00865*, 2026. 34

[47] Jungho Lee, Minhyeok Lee, Sunghun Yang, Minseok Kang, and Sangyoun Lee. SwiftVGGT: A scalable visual geometry grounded transformer for large-scale scenes. In *CVPR Findings*, 2026. 3

[48] Vincent Leroy, Yohann Cabon, and Jerome Revaud. Grounding image matching in 3D with MASt3R. In *ECCV*, 2024. 3

[49] Stefan Leutenegger, Simon Lynen, Michael Bosse, Roland Siegwart, and Paul Furgale. Keyframe-based visual-inertial odometry using nonlinear optimization. *IJRR*, 2015. 2

[50] Huan Li, Longjun Luo, Yuling Shi, and Xiaodong Gu. Analyzing the mechanism of attention collapse in VGGT from a dynamics perspective. *arXiv preprint arXiv:2512.21691*, 2025. 3

[51] Hao Li, Zhengyu Zou, Fangfu Liu, Xuanyang Zhang, Fangzhou Hong, Yukang Cao, Yushi Lan, Manyuan Zhang, Gang Yu, Dingwen Zhang, and Ziwei Liu. IGGT: Instance-grounded geometry transformer for semantic 3D reconstruction. In *ICLR*, 2026. 3

[52] Zizun Li, Jianjun Zhou, Yifan Wang, Haoyu Guo, Wenzheng Chang, Yang Zhou, Haoyi Zhu, Junyi Chen, Chunhua Shen, and Tong He. WinT3R: Window-based streaming reconstruction with camera token pool. In *ICLR*, 2026. 3

[53] Haotong Lin, Sili Chen, Jun Hao Liew, Donny Y. Chen, Zhenyu Li, Yang Zhao, Sida Peng, Hengkai Guo, Xiaowei Zhou, Guang Shi, Jiashi Feng, and Bingyi Kang. Depth Anything 3: Recovering the visual space from any views. In *ICLR*, 2026. 1, 3, 29

[54] Changkun Liu, Jiezhi Yang, Zeman Li, Yuan Deng, Jiancong Guo, and Luca Ballan. Mem3R: Streaming 3D reconstruction with hybrid memory via test-time training. *arXiv preprint arXiv:2604.07279*, 2026. 3

[55] Xuanyi Liu, Chunan Yu, Deyi Ji, Qi Zhu, Lingyun Sun, Xuanfu Li, Jin Ma, Tianrun Chen, and Lanyun Zhu. StreamCacheVGGT: Streaming visual geometry transformers with robust scoring and hybrid cache compression. *arXiv preprint arXiv:2604.15237*, 2026. 3

[56] Yang Liu, Chuanchen Luo, Zimo Tang, Junran Peng, and Zhaoxiang Zhang. VGGT-X: When VGGT meets dense novel view synthesis. *arXiv preprint arXiv:2509.25191*, 2025. 3

[57] Jiahao Lu, Tianyu Huang, Peng Li, Zhiyang Dou, Cheng Lin, Zhiming Cui, Zhen Dong, Sai-Kit Yeung, Wenping Wang, and Yuan Liu. Align3R: Aligned monocular depth estimation for dynamic videos. In *CVPR*, 2025. 3

[58] Si-Yu Lu, Po-Ting Chen, Hui-Che Hsu, Sin-Ye Jhong, Wen-Huang Cheng, and Yung-Yao Chen. OVGGT: O(1) constant-cost streaming visual geometry transformer. *arXiv preprint arXiv:2603.05959*, 2026. 3

[59] Dominic Maggio, Hyungtae Lim, and Luca Carlone. VGGT-SLAM: Dense RGB SLAM optimized on the SL(4) manifold. In *NeurIPS*, 2025. 5

[60] Mohammad Mahdavian, Gordon Tan, Binbin Xu, Yuan Ren, Dongfeng Bai, and Bingbing Liu. UniScale: Unified scale-aware 3D reconstruction for multi-view understanding via prior injection for robotic perception. *arXiv preprint arXiv:2602.23224*, 2026. 3

[61] Soroush Mahdi, Fardin Ayar, Ehsan Javanmardi, Manabu Tsukada, and Mahdi Javanmardi. Evict3R: Training-free token eviction for memory-bounded streaming visual geometry transformers. *arXiv preprint arXiv:2509.17650*, 2025. 3

[62] Xingyu Miao, Weiguang Zhao, Tao Lu, Linning Xu, Mulin Yu, Yang Long, Jiangmiao Pang, and Junting Dong. TrajVG: 3D trajectory-coupled visual geometry learning. *arXiv preprint arXiv:2602.04439*, 2026. 3

[63] Emanuele Palazzolo, Jens Behley, Philipp Lottes, Philippe Giguère, and Cyrill Stachniss. ReFusion: 3D reconstruction in dynamic environments for RGB-D cameras exploiting residuals. In *IROS*, 2019. 7, 8, 26, 33

[64] Junxiang Pan, Lipu Zhou, and Baojie Chen. HyVGGT-VO: Tightly coupled hybrid dense visual odometry with feed-forward models. *arXiv preprint arXiv:2604.02107*, 2026. 3

[65] Sicheng Pan, Chen Tang, Shuzhao Xie, Ke Yang, Weixiang Zhang, Jiawei Li, Bin Chen, Shu-Tao Xia, and Zhi Wang. Tail-aware post-training quantization for 3D geometry models. *arXiv preprint arXiv:2602.01741*, 2026. 3

[66] Haosong Peng, Hao Li, Yalun Dai, Yushi Lan, Yihang Luo, Tianyu Qi, Zhengshen Zhang, Yufeng Zhan, Junfei Zhang, Wenchao Xu, and Ziwei Liu. OmniVGGT: Omni-modality driven visual geometry grounded transformer. In *CVPR*, 2026. 3

[67] Shenhan Qian, Ganlin Zhang, Shangzhe Wu, and Daniel Cremers. Flow4R: Unifying 4D reconstruction and tracking with scene flow. *arXiv preprint arXiv:2602.14021*, 2026. 3

[68] Zhicheng Qiu, Jiarui Meng, Tong an Luo, Yican Huang, Xuan Feng, Xuanfu Li, and Zhan Xu. SLARM: Streaming and language-aligned reconstruction model for dynamic scenes. *arXiv preprint arXiv:2603.22893*, 2026. 3

[69] Weining Ren, Xiao Tan, and Kai Han. Speed3R: Sparse feed-forward 3D reconstruction models. In *CVPR Findings*, 2026. 3, 7, 8, 9

[70] Johannes L. Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *CVPR*, 2016. 1, 3

[71] Guole Shen, Tianchen Deng, Xingrui Qin, Nailin Wang, Jianyu Wang, Yanbo Wang, Yongtao Chen, Hesheng Wang, and Jingchuan Wang. MUT3R: Motion-aware updating transformer for dynamic 3D reconstruction. *arXiv preprint arXiv:2512.03939*, 2025. 3

[72] You Shen, Zhipeng Zhang, Yansong Qu, Xiawu Zheng, Jiayi Ji, Shengchuan Zhang, and Liujuan Cao. FastVGGT: Fast visual geometry transformer. In *ICLR*, 2026. 2, 3, 4, 7, 8, 9, 29

[73] Jamie Shotton, Ben Glocker, Christopher Zach, Shahram Izadi, Antonio Criminisi, and Andrew Fitzgibbon. Scene coordinate regression forests for camera relocalization in RGB-D images. In *CVPR*, 2013. 4, 5, 7, 26, 29, 30, 31, 33

[74] Zhijian Shu, Cheng Lin, Tao Xie, Wei Yin, Ben Li, Zhiyuan Pu, Weize Li, Yao Yao, Xun Cao, Xiaoyang Guo, and Xiao-Xiao Long. LiteVGGT: Boosting vanilla VGGT via geometry-aware cached token merging. In *CVPR*, 2026. 2, 3, 4, 6, 7, 8, 9, 28

[75] Jürgen Sturm, Nikolas Engelhard, Felix Endres, Wolfram Burgard, and Daniel Cremers. A benchmark for the evaluation of RGB-D SLAM systems. In *IROS*, 2012. 7, 33

[76] Edgar Sucar, Zihang Lai, Eldar Insafutdinov, and Andrea Vedaldi. Dynamic point maps: A versatile representation for dynamic 3D reconstruction. In *ICCV*, 2025. 3

[77] Shuo Sun, Unal Artan, Malcolm Mielle, Achim J. Lilienthaland, and Martin Magnusson. Dense dynamic scene reconstruction and camera pose estimation from multi-view videos. *arXiv preprint arXiv:2603.12064*, 2026. 3

[78] Xianbing Sun, Zhikai Zhu, Zhengyu Lou, Bo Yang, Jinyang Tang, Liqing Zhang, He Wang, and Jianfu Zhang. AVGGT: Rethinking global attention for accelerating VGGT. In *CVPR*, 2026. 3, 6, 7

[79] Yu Sun, Xinhao Li, Karan Dalal, Jiarui Xu, Arjun Vikram, Genghan Zhang, Yann Dubois, Xinlei Chen, Xiaolong Wang, Sanmi Koyejo, Tatsunori Hashimoto, and Carlos Guestrin. Learning to (learn at test time): RNNs with expressive hidden states. In *ICML*, 2025. 34

[80] Zhenggang Tang, Yuchen Fan, Dilin Wang, Hongyu Xu, Rakesh Ranjan, Alexander Schwing, and Zhicheng Yan. MV-DUSt3R+: Single-stage scene reconstruction from sparse views in 2 seconds. In *CVPR*, 2025. 3

[81] Chen Wang, Hao Tan, Wang Yifan, Zhiqin Chen, Yuheng Liu, Kalyan Sunkavalli, Sai Bi, Lingjie Liu, and Yiwei Hu. tttLRM: Test-time training for long context and autoregressive 3D reconstruction. In *CVPR*, 2026. 34

[82] Chung-Shien Brian Wang, Christian Schmidt, Jens Piekenbrinck, and Bastian Leibe. Block-sparse global attention for efficient multi-view geometry transformers. In *CVPR*, 2026. 2, 7, 8, 9, 29

[83] Hengyi Wang and Lourdes Agapito. AMB3R: Accurate feed-forward metric-scale 3D reconstruction with backend. In *CVPR*, 2026. 3

[84] Haonan Wang, Hanyu Zhou, Haoyue Liu, and Luxin Yan. 4D-VGGT: A general foundation model with spatiotemporal awareness for dynamic scene geometry estimation. *arXiv preprint arXiv:2511.18416*, 2025. 3

[85] Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. VGGT: Visual geometry grounded transformer. In *CVPR*, 2025. 1, 2, 3, 6, 7, 8, 9, 24, 29, 33

[86] Runze Wang, Yuxuan Song, Youcheng Cai, and Ligang Liu. STAC: Plug-and-play spatio-temporal aware cache compression for streaming 3D reconstruction. In *CVPR*, 2026. 3

[87] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. DUSt3R: Geometric 3D vision made easy. In *CVPR*, 2024. 3

[88] Weitian Wang, Lukas Meiner, Rai Shubham, Cecilia De La Parra, and Akash Kumar. HTTM: Head-wise temporal token merging for faster VGGT. In *CVPR*, 2026. 3

[89] Yifan Wang, Jianjun Zhou, Haoyi Zhu, Wenzheng Chang, Yang Zhou, Zizun Li, Junyi Chen, Jiangmiao Pang, Chunhua Shen, and Tong He. π^3: Permutation-equivariant visual geometry learning. In *ICLR*, 2026. 1, 3, 4, 7, 8, 24, 29, 33

[90] Zipeng Wang and Dan Xu. FlashVGGT: Efficient and scalable visual geometry transformers with compressed descriptor attention. In *CVPR*, 2026. 3

[91] Zichen Wang, Ang Cao, Liam J. Wang, and Jeong Joon Park. MoE3D: A mixture-of-experts module for 3D reconstruction. *arXiv preprint arXiv:2601.05208*, 2026. 3

[92] Philippe Weinzaepfel, Vincent Leroy, Thomas Lucas, Romain Brégier, Yohann Cabon, Vaibhav Arora, Leonid Antsfeld, Boris Chidlovskii, Gabriela Csurka, and Jérôme Revaud. CroCo: Self-supervised pre-training for 3D vision tasks by cross-view completion. In *NeurIPS*, 2022. 3

[93] Philippe Weinzaepfel, Thomas Lucas, Vincent Leroy, Yohann Cabon, Vaibhav Arora, Romain Brégier, Gabriela Csurka, Leonid Antsfeld, Boris Chidlovskii, and Jérôme Revaud. CroCo v2: Improved cross-view completion pre-training for stereo matching and optical flow. In *ICCV*, 2023. 3

[94] Changli Wu, Haodong Wang, Jiayi Ji, Yutian Yao, Chunsai Du, Jihua Kang, Yanwei Fu, and Liujuan Cao. MVGGT: Multimodal visual geometry grounded transformer for multiview 3D referring expression segmentation. In *CVPR*, 2026. 3

[95] Xianfeng Wu, Yajing Bai, Minghan Li, Xianzu Wu, Xueqi Zhao, Zhongyuan Lai, Wenyu Liu, and Xinggang Wang. 4DLangVGGT: 4D language-visual geometry grounded transformer. *arXiv preprint arXiv:2512.05060*, 2025. 3

[96] Mochu Xiang, Zhelun Shen, Xuesong Li, Jiahui Ren, Jing Zhang, Chen Zhao, Shanshan Liu, Haocheng Feng, Jingdong Wang, and Yuchao Dai. RnG: A unified transformer for complete 3D modeling from partial observations. In *CVPR*, 2026. 3

[97] Tao Xie, Peishan Yang, Yudong Jin, Yingfeng Cai, Wei Yin, Weiqiang Ren, Qian Zhang, Wei Hua, Sida Peng, Xiaoyang Guo, and Xiaowei Zhou. Scal3R: Scalable test-time training for large-scale 3D reconstruction. In *CVPR*, 2026. 34

[98] Zhuang Xiong, Chen Zhang, Qingshan Xu, and Wenbing Tao. VGGT-Motion: Motion-aware calibration-free monocular SLAM for long-range consistency. *arXiv preprint arXiv:2602.05508*, 2026. 3

[99] Kai Xu, Tze Ho Elden Tse, Jizong Peng, and Angela Yao. DAS3R: Dynamics-aware gaussian splatting for static scene reconstruction. *arXiv preprint arXiv:2412.19584*, 2024. 3

[100] Yangfan Xu, Lilian Zhang, Xiaofeng He, Pengdong Wu, Wenqi Wu, and Jun Mao. GPA-VGGT: Adapting VGGT to large scale localization by self-supervised learning with geometry and physics aware loss. *arXiv preprint arXiv:2601.16885*, 2026. 3

[101] Zhisong Xu and Takeshi Oishi. FrameVGGT: Frame evidence rolling memory for streaming VGGT. *arXiv preprint arXiv:2603.07690*, 2026. 3

[102] Yibin Yan, Jilan Xu, Shangzhe Di, Haoning Wu, and Weidi Xie. OmniStream: Mastering perception, reconstruction and action in continuous streams. *arXiv preprint arXiv:2603.12265*, 2026. 3

[103] 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 *CVPR*, 2025. 3

[104] Shunyu Yao, Binghui Peng, Christos Papadimitriou, and Karthik Narasimhan. Self-attention networks can process bounded hierarchical languages. In *ACL*, 2021. 2

[105] Haibao Yu, Kuntao Xiao, Jiahang Wang, Ruiyang Hao, Yuxin Huang, Guoran Hu, Haifang Qin, Bowen Jing, Yuntian Bo, and Ping Luo. ReconDrive: Fast feed-forward 4D gaussian splatting for autonomous driving scene reconstruction. *arXiv preprint arXiv:2603.07552*, 2026. 3

[106] Jiayi Yuan, Haobo Jiang, De Wen Soh, and Na Zhao. VGGT-360: Geometry-consistent zero-shot panoramic depth estimation. *arXiv preprint arXiv:2603.18943*, 2026. 3

[107] Shuai Yuan, Yantai Yang, Xiaotian Yang, Xupeng Zhang, Zhonghao Zhao, Lingming Zhang, and Zhipeng Zhang. InfiniteVGGT: Visual geometry grounded transformer for endless streams. *arXiv preprint arXiv:2601.02281*, 2026. 3

[108] Yuheng Yuan, Qiuhong Shen, Shizun Wang, Xingyi Yang, and Xinchao Wang. Test3R: Learning to reconstruct 3D at test time. In *NeurIPS*, 2025. 3

[109] Ying Zang, Yidong Han, Chaotao Ding, Yuanqi Hu, Deyi Ji, Qi Zhu, Xuanfu Li, Jin Ma, Lingyun Sun, Tianrun Chen, and Lanyun Zhu. Robust 4D visual geometry transformer with uncertainty-aware priors. *arXiv preprint arXiv:2604.09366*, 2026. 3

[110] Junyi Zhang, Charles Herrmann, Junhwa Hur, Chen Sun, Ming-Hsuan Yang, Forrester Cole, Trevor Darrell, and Deqing Sun. LoGeR: Long-context geometric reconstruction with hybrid memory. *arXiv preprint arXiv:2603.03269*, 2026. 34

[111] Songyan Zhang, Yongtao Ge, Jinyuan Tian, Guangkai Xu, Hao Chen, Chen Lv, and Chunhua Shen. POMATO: Marrying pointmap matching with temporal motions for dynamic 3D reconstruction. In *ICCV*, 2025. 3

[112] Shangzhan Zhang, Jianyuan Wang, Yinghao Xu, Nan Xue, Christian Rupprecht, Xiaowei Zhou, Yujun Shen, and Gordon Wetzstein. FLARE: Feed-forward geometry, appearance and camera estimation from uncalibrated sparse views. In *CVPR*, 2025. 3

[113] Xuechen Zhang, Xiangyu Chang, Mingchen Li, Amit Roy-Chowdhury, Jiasi Chen, and Samet Oymak. Selective attention: Enhancing transformer through principled context control. In *NeurIPS*, 2024. 2

[114] Zetong Zhang, Manuel Kaufmann, Lixin Xue, Jie Song, and Martin R. Oswald. ODHSR: Online dense 3D reconstruction of humans and scenes from monocular videos. In *CVPR*, 2025. 3

[115] Zhijie Zheng, Xinhao Xiang, and Jiawei Zhang. TTSA3R: Training-free temporal-spatial adaptive persistent state for streaming 3D reconstruction. *arXiv preprint arXiv:2601.22615*, 2026. 3

[116] Kaichen Zhou, Yuhan Wang, Grace Chen, Xinhai Chang, Gaspard Beaudouin, Fangneng Zhan, Paul Pu Liang, and Mengyu Wang. PAGE-4D: Disentangled pose and geometry estimation for VGGT-4D perception. In *ICLR*, 2026. 3

[117] Dong Zhuo, Wenzhao Zheng, Jiahe Guo, Yuqi Wu, Jie Zhou, and Jiwen Lu. Streaming visual geometry transformer. In *ICLR*, 2026. 3

[118] Sicheng Zuo, Zixun Xie, Wenzhao Zheng, Shaoqing Xu, Fang Li, Shengyin Jiang, Long Chen, Zhi-Xin Yang, and Jiwen Lu. DVGT: Driving visual geometry transformer. In *CVPR*, 2026. 3

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 claims made in the abstract and introduction accurately reflect the paper's contributions and scope.

   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: The limitations of the work have been discussed in Section 5 in the main paper.

   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 does not include theoretical results.

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: We have discussed all the details to reproduce all experimental results in our paper.

   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: We have published our code at https://github.com/zsh2000/gotohunt.

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: The inference details including data processing and hyperparameter choices have been specified in the paper, which are enough to understand the results.

   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: We have measured the variance of different seeds in Section B in the Appendix.

   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: The compute resources required for reproduce the is specified in Section 4 in the paper.

   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: The conducted research conforms with the NeurIPS Code of Ethics in every respect.

   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: Both positive and negative societal impacts have been discussed in Section I in the Appendix.

    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: Our paper does not have such risks.

    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: We have properly credited and listed the license in Section G in the Appendix. Please note that licenses for some datasets are extremely lengthy as each scene in these datasets have different licenses. Therefore, we include the links for reference instead of directly listing all the 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: [N/A]

    Justification: At the submission time, we have not released the new assets used in the paper.

    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 nor 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 crowdsourcing nor research with human subjects.

    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: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

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.

Good Token Hunting: A Hitchhiker's Guide to Token Selection for Visual Geometry Transformers

Technical Appendices and Supplementary Material

In the appendix, we provide additional clarfication, analyses and experiments to further validate the effectiveness and provide guidelines of our method along with more discussions. First, in Section A and Section B, we illustrate the details of the proposed diversity-based inter-frame selection algorithm, and the robustness of random seed selection in this process. Section C displays the exploration of several intra-frame strategies that we attempted during our research, but ultimately chose not to be applied on our final method, in order to provide more guidelines for future research. Then, in Section D, we provide additional experiments on the sensitivity analysis of the hyperparameters used in our method, demonstrating the robustness of our method to parameter choices. In addition, we illustrate in Section E that our method can be easily applied to other visual geometry transformers with different architecture. Afterwards, in Section F, we elaborate the evaluation metrics used in the tasks. Section G include the license and terms of use for the models and data used in the paper. Finally, Section H and Section I provide additional discussions and societal impact of our work.

A Algorithm details for inter-frame selection

In Section 3.2, we propose the diversity-based inter-frame selection solution. We demonstrate the algorithm details in Algorithm A. Given image-level features extracted by a place recognition model [4], we first construct a pairwise similarity matrix using cosine similarity, serving as approximation for co-visibility between frames, which is then converted into a distance metric for the following sampling process. The selection problem is formulated as a "K-center" objective, aiming to choose a subset of frames that maximizes coverage of the scene. To efficiently approximate this objective, we adopt the greedy farthest point sampling (FPS) algorithm. Starting from a randomly initialized frame, FPS iteratively selects the frame that is farthest from the current selected set, thereby encouraging diversity among selected frames. The resulting subset is shared across all query tokens and serves as the inter-frame support "anchor" set for global attention. In practice, this procedure is computationally efficient, requiring only a single $N \times N$ similarity computation and K iterative updates.

Additionally, we would like to point out that for VGGT [85], the selected reference set needs to include the first frame, which serves as the reference view of the scene from the model design. In our implementation, if the above selection process is done but the first frame is still absent in the selected set, we replace the final selected frame with the first frame to preserve the budget K. No corresponding rules for the reference need to be applied for the permutation-equivariant architectures like π^3 [89].

B Variance analysis on random seeds for inter-frame selection

In Section A, we mention that inter-frame selection starts from a randomly initialized frame and performs the FPS algorithm afterwards. To demonstrate the robustness of model performance regarding random seed selection, we use 5 random seeds and report the mean and standard deviation for the camera pose estimation task in Table A. We can observe that the performance of our method is generally robust to the randomness from the first frame pick. This randomness stems from the nature of our diversity-based frame selection algorithm. As long as the number of selected frames is not too small to cover most parts of the 3D scene, the selected frames from our Algorithm A, inspired by farthest point sampling, can excel in finding a set of frames that cover most of the 3D scene to achieve satisfactory results.

C Additional analysis on intra-frame strategy

C.1 Token-level diversity-based selection

In Section 3.3, we adopt an intra-frame strategy to downsample the token map along the two spatial dimensions, which is independent to the token content. One intuitive consideration is that, what if

Algorithm A Diversity-based Inter-frame Selection

Input: Feature map $F \in \mathbb{R}^{N \times d}$ (row f_i is the d-dim feature of image i); number of neighbors K; random seed σ
Output: Selected index set S with $|S| = K$ (shared by all query frames)

// Step 1: build the covisibility matrix from features (cosine similarity).
$\tilde{f}_i \leftarrow f_i / \|f_i\|_2, \quad \forall\, i \in \{0, \ldots, N-1\}$ ▷ $\mathcal{L}_2$-normalized each row
$\tilde{F} \leftarrow [\,\tilde{f}_0;\ \tilde{f}_1;\ \ldots;\ \tilde{f}_{N-1}\,] \in \mathbb{R}^{N \times d}$ ▷ stack normalized rows
$C \leftarrow \tilde{F}\tilde{F}^{\top}, \quad C_{i,j} \in [-1, 1]$ ▷ covisibility matrix with cosine similarity between features

// Step 2: Convert covisibility to a distance.
$D_{i,j} \leftarrow \max(C) - C_{i,j}, \quad \forall\, i, j$ ▷ larger covisibility ⇒ closer distance

// Step 3: FPS starts from a random first pick.
$b_1 \sim \mathrm{Uniform}\{0, \ldots, N-1\}$ ▷ random selection with seed σ
$S \leftarrow \{b_1\}$ ▷ add frame to the selected set
$d_{\min} \leftarrow D_{b_1,:}$ ▷ $d_{\min} \in \mathbb{R}^N$ stores the distance of each frame to the selected set
$d_{\min}[b_1] \leftarrow -\infty$ ▷ prevent re-selecting the already selected frame

// Step 4: FPS for the remaining $K-1$ picks.
for $k \leftarrow 2$ **to** K **do**
  $b \leftarrow \arg\max_j d_{\min}[j]$ ▷ select the farthest frame from the currently selected set
  $S \leftarrow S \cup \{b\}$ ▷ add frame to the selected set
  $d_{\min}[b] \leftarrow -\infty$ ▷ prevent re-selecting the already selected frame
  $d_{\min} \leftarrow \min\big(d_{\min},\ D_{b,:}\big)$ ▷ update the distance change due to the newly added frame
end for

return S

Table A: Quantitative results camera pose estimation on 5 random seeds with mean and standard deviation. $K = 25$ is selected for the experimental settings.

Backbone	σ	ATE (↓)	RPE-rot (↓)	RPE-trans (↓)
7-Scenes				
VGGT	2	0.0669 ± 0.0009	0.4372 ± 0.0135	0.0164 ± 0.0003
VGGT	3	0.0672 ± 0.0009	0.4386 ± 0.0153	0.0165 ± 0.0004
π^3	2	0.0576 ± 0.0006	0.3447 ± 0.0019	0.0112 ± 0.0001
π^3	3	0.0576 ± 0.0006	0.3441 ± 0.0018	0.0112 ± 0.0001
Neural RGB-D				
VGGT	2	0.0273 ± 0.0010	0.1946 ± 0.0464	0.0161 ± 0.0003
VGGT	3	0.0272 ± 0.0009	0.2109 ± 0.0231	0.0175 ± 0.0005
π^3	2	0.0292 ± 0.0011	0.1184 ± 0.0016	0.0122 ± 0.0001
π^3	3	0.0296 ± 0.0009	0.1199 ± 0.0018	0.0122 ± 0.0001

we adopt a similar diversity-based strategy as inter-frame selection, in the current intra-frame stage, leading to a diversity-based intra-frame selection strategy. However, the total number of tokens across frames can be prohibitively large for directly applying such a method even after inter-frame selection. To address this, we design an approximate solution. Specifically, for each selected frame, we first over-select twice the target budget using FPS. Then, we compute a *redundancy score* for each token, which is its maximum cosine similarity to the mean feature of every other frame. Finally, we retain the least redundant tokens from the selected tokens. This procedure directly guarantees diversity within each frame and, to some extent, promotes token diversity across frames. We refer to this Token-level Diversity-based strategy as *TLD*.

We observe that *TLD* performs well in certain scenarios, such as video depth estimation in Table B, where it consistently outperforms the *standard* strategy in nearly all configurations. However, in

other tasks, including pose estimation in Table C and 3D reconstruction in Table D, its performance is comparable to or sometimes worse than the *standard* approach.

In addition, unlike diversity-based inter-frame sampling, which is performed only once in the image space, *TLD* requires FPS within each selected frame even before advancing to the second selection stage, resulting in non-negligible computational overhead. It typically takes roughly 5 seconds for a scene with 500 frames. Therefore, this strategy is not applied in our official solution, but we report it here to potentially inspire future research.

Table B: Quantitative comparison on the token-level diversity-based *(TLD)* selection strategy for intra-frame downsampling with video depth prediction on the full length Bonn [63] dataset.

K	σ	**Strategy**	**Layers**	**Abs Rel (↓)**	**Log RMSE (↓)**	**RMSE (↓)**	$\delta < 1.25$ **(↑)**
10	2	*TLD*	0-10	0.0305	0.0686	0.1512	0.9889
10	2	*Standard*	0-10	0.0352	0.0748	0.1593	0.9882
10	3	*TLD*	0-10	0.0292	0.0666	0.1480	0.9892
10	3	*Standard*	0-10	0.0346	0.0717	0.1553	0.9892
10	3	*TLD*	11-17	0.0416	0.0779	0.1653	0.9841
10	3	*Standard*	11-17	0.0436	0.0815	0.1675	0.9824
25	2	*TLD*	0-10	0.0290	0.0667	0.1484	0.9892
25	2	*Standard*	0-10	0.0340	0.0729	0.1565	0.9887
25	3	*TLD*	0-10	0.0290	0.0651	0.1457	0.9896
25	3	*Standard*	0-10	0.0303	0.0664	0.1481	0.9897
100	2	*TLD*	0-10	0.0283	0.0666	0.1495	0.9895
100	2	*Standard*	0-10	0.0336	0.0729	0.1568	0.9889
100	3	*TLD*	0-10	0.0275	0.0648	0.1605	0.9901
100	3	*Standard*	0-10	0.0291	0.0653	0.1702	0.9901
π^3 *(Base Model)*				0.0346	0.0763	0.1642	0.9880

Table C: Quantitative comparison on the token-level diversity-based *(TLD)* selection strategy for intra-frame downsampling with camera pose estimation on 7-Scenes [73] dataset.

K	σ	**Strategy**	**Layers**	**ATE (↓)**	**RPE-rot (↓)**	**RPE-trans (↓)**
10	2	*TLD*	0-10	0.0578	0.3602	0.0123
10	2	*Standard*	0-10	0.0589	0.3675	0.0125
10	3	*TLD*	0-10	0.0581	0.3584	0.0122
10	3	*Standard*	0-10	0.0587	0.3747	0.0130
25	2	*TLD*	0-10	0.0576	0.3423	0.0111
25	2	*Standard*	0-10	0.0580	0.3430	0.0112
25	3	*TLD*	0-10	0.0570	0.3415	0.0111
25	3	*Standard*	0-10	0.0568	0.3415	0.0112
100	2	*TLD*	0-10	0.0573	0.3363	0.0105
100	2	*Standard*	0-10	0.0577	0.3369	0.0106
100	3	*TLD*	0-10	0.0571	0.3384	0.0106
100	3	*Standard*	0-10	0.0566	0.3384	0.0106
π^3 *(Base Model)*				0.0573	0.3389	0.0105

C.2 Replacing global attention with mean pooling

Replacing global attention with local attention adopted in the main paper is already a fairly radical strategy. An even more aggressive alternative is to skip the attention mechanism entirely and replace it with mean pooling over all value tokens. The justification of this simplification comes from the attention formulation, where the attention output for each query q_i is $o_i = \sum_j \mathrm{softmax}\left(\frac{q_i \cdot k_j}{\sqrt{d}}\right) v_j$. If the attention activations $q_i \cdot k_j$ are nearly identical across all j, then the softmax weights approach $\frac{1}{N}$, where N is the number of total K/V tokens. In this regime, the attention calculation collapse to a simple averaging of the all value tokens.

Table D: Quantitative comparison on the token-level diversity-based *(TLD)* selection strategy for intra-frame downsampling with point map estimation on Neural RGB-D [3] dataset.

K	σ	**Strategy**	**Layers**	**Acc (↓)**	**Comp (↓)**	**NC (↑)**
10	2	*TLD*	0-8	0.0158	0.0119	0.7486
10	2	*Standard*	0-8	0.0153	0.0114	0.7477
10	3	*TLD*	0-8	0.0161	0.0119	0.7485
10	3	*Standard*	0-8	0.0145	0.0112	0.7470
25	2	*TLD*	0-8	0.0126	0.0112	0.7588
25	2	*Standard*	0-8	0.0126	0.0110	0.7561
25	3	*TLD*	0-8	0.0127	0.0111	0.7571
25	3	*Standard*	0-8	0.0126	0.0111	0.7564
100	2	*TLD*	0-8	0.0128	0.0113	0.7609
100	2	*Standard*	0-8	0.0130	0.0111	0.7596
100	3	*TLD*	0-8	0.0130	0.0113	0.7602
100	3	*Standard*	0-8	0.0129	0.0112	0.7626
VGGT *(Base Model)*				0.0160	0.0112	0.7508

We implement this variant with the results reported in Table E. The performance degradation is noticeable across many configurations, suggesting that this approximation is too crude and is not robust enough for accelerating visual geometry transformers.

C.3 Layer partitioning with entropy thresholds

To determine the thresholds for intra-frame selection strategies, we introduce two layer indices, l_{local} and l_{sample}, which partition the global attention layers of visual geometry transformers into groups adopting different intra-frame strategies. A natural question to ask is that, *can these partitioning rules can be made adaptive to the input sequence?*

Since the choice of l_{local} and l_{sample} is motivated by observations of the entropy of global attention patterns, a straightforward extension is to use entropy on each layer to adaptively group layers for different intra-frame strategy. Concretely, we define two entropy thresholds, τ_1 and τ_2. Starting from the first layer, we apply the most aggressive strategy, which is to replace global attention with local attention, until reaching the first layer where $\mathcal{H} < \tau_1$. From that point, we switch to the *Standard* intra-frame downsampling strategy, and continue until encountering the first layer where the inter-frame entropy drops to $\mathcal{H} < \tau_2$.

The results are reported in Table F and Table G. While this adaptive scheme achieves competitive performance and serves as a viable alternative for determining layer-wise strategies, we do not adopt it in our final model for reasons similar to the *TLD* strategy in Section C.1. Specifically, computing attention entropy on-the-fly during the global attention pass introduces significant overhead, adding approximately 7 seconds for a 500-frame scene. Therefore, although promising, we present this approach as a reference to motivate future work on more efficient ways of leveraging entropy to inform intra-frame strategy selection.

C.4 Intra-frame strategy for late layers

Observing from the attention pattern analysis in Figure 4, we notice that several of the latest layers also have the diluted attention patterns, which motivates us to consider: *can intra-frame selection be applied on these final layers as well?* To investigate on this question, we introduce another threshold l_{late}, such that intra-frame downsampling is also applied to layers with indices $l \geqslant l_{\text{late}}$. Results are presented in Table H and Table I. While performance can improve in certain cases, it is generally more sensitive to the choice of l_{late}, not as robust as to l_{local} and l_{sample} as shown in Table 9 and Table J. We attribute this sensitivity to the proximity of these layers to the final output, where small changes can have a large impact on the final performance.

Table E: Quantitative comparison on the more extreme approximation to replace global attention layers with mean pooling. The experiments are conducted with $K = 25$ and $\sigma = 2$ on the camera pose estimation on Neural RGB-D [3]. Essentially, we replace the local attention *(Local)* in $l \leqslant l_{\text{local}}$ with mean pooling *(Pool)* layers. The inferior performance with the *Pool* strategy demonstrates that it is not robust enough to be instantiated in our method.

l_{local}	l_{global}	**Strategy**	**ATE (↓)**	**RPE-rot (↓)**	**RPE-trans (↓)**
1	7	*Local*	0.0268	0.2118	0.0170
1	7	*Pool*	0.0288	0.3353	0.0198
1	8	*Local*	0.0264	0.1946	0.0165
1	8	*Pool*	0.0286	0.3235	0.0195
1	9	*Local*	0.0266	0.1907	0.0164
1	9	*Pool*	0.0286	0.3163	0.0193
1	10	*Local*	0.0266	0.1659	0.0158
1	10	*Pool*	0.0280	0.2809	0.0184
2	7	*Local*	0.0269	0.2000	0.0169
2	7	*Pool*	0.0324	0.4391	0.0242
2	8	*Local*	0.0266	0.1847	0.0163
2	8	*Pool*	0.0316	0.4261	0.0236
2	9	*Local*	0.0267	0.1794	0.0162
2	9	*Pool*	0.0318	0.4248	0.0235
2	10	*Local*	0.0267	0.1615	0.0158
2	10	*Pool*	0.0310	0.3936	0.0223
3	7	*Local*	0.0268	0.1652	0.0160
3	7	*Pool*	0.0335	0.4444	0.0247
3	8	*Local*	0.0266	0.1548	0.0158
3	8	*Pool*	0.0328	0.4336	0.0240
3	9	*Local*	0.0267	0.1538	0.0156
3	9	*Pool*	0.0330	0.4348	0.0242
3	10	*Local*	0.0267	0.1492	0.0153
3	10	*Pool*	0.0322	0.4089	0.0229
VGGT *(Base Model)*			0.0374	0.2934	0.0186

D Additional experimental results for sensitivity analysis

D.1 Threshold hyperparameters

In Section 4.3, we demonstrate that the threshold hyperparameters l_{local} and l_{sample} are robust when with π^3 as the base model. We further verify this robustness with VGGT as the base model in Table J, further enhancing the reliability and soundness of our method.

D.2 Performance under varying sequence lengths

As the goal of our method is to accelerate feed-forward 3D reconstruction, with the primary part of our work demonstrating how the inference time of our model can grow near-linearly with minimized performance compromise. To better complement our evaluation with varying length sequences, we evaluate chronological prefixes of 100-400 frames with the same sampling interval of 2 as the main experiments in Section 4.2. After sampling, since two scenes in 7-Scenes only contain 250 frames, we average results over all 18 scenes for the 100/200-frame settings and over the 16 eligible scenes for the 300/400-frame settings. As shown in Table K, when the sequence length becomes shorter, our method can still remain competitive with the baseline and other training-free acceleration methods, although it may falls short to methods like LiteVGGT [74] which requires additional training for the accelerated model.

Table F: Quantitative results of using entropy thresholds to determine the intra-frame strategies for each layer. Experiments are conducted on camera pose estimation with $K = 25$ and $\sigma = 3$ on Neural RGB-D [3].

τ_1	τ_2	**ATE (↓)**	**RPE-rot (↓)**	**RPE-trans (↓)**
0.99	0.95	0.0278	0.2948	0.0185
0.99	0.92	0.0279	0.3455	0.0198
0.97	0.95	0.0271	0.1871	0.0165
0.97	0.92	0.0270	0.2405	0.0176
0.97	0.90	0.0263	0.1523	0.0155
0.95	0.92	0.0264	0.1911	0.0164
0.95	0.90	0.0262	0.1469	0.0153
0.95	0.88	0.0261	0.1455	0.0153
0.93	0.90	0.0273	0.1497	0.0156
0.93	0.88	0.0271	0.1491	0.0155
0.91	0.88	0.0444	0.2820	0.0223
***GoToHunt* (Ours)**		0.0270	0.2409	0.0176
VGGT *(Base Model)*		0.0374	0.2934	0.0186

Table G: Quantitative results of using entropy thresholds to determine the intra-frame strategies for each layer. Experiments are conducted on camera pose estimation with $K = 25$ and $\sigma = 2$ on 7-Scenes [73].

τ_1	τ_2	**ATE (↓)**	**RPE-rot (↓)**	**RPE-trans (↓)**
0.99	0.95	0.0675	0.4446	0.0168
0.99	0.92	0.0675	0.4428	0.0167
0.97	0.95	0.0672	0.4471	0.0164
0.97	0.92	0.0671	0.4454	0.0164
0.97	0.90	0.0673	0.4471	0.0165
0.95	0.92	0.0672	0.4454	0.0164
0.95	0.90	0.0673	0.4471	0.0165
0.95	0.88	0.0682	0.4461	0.0161
0.93	0.90	0.0673	0.4517	0.0165
0.93	0.88	0.0682	0.4506	0.0161
0.91	0.88	0.0715	0.4880	0.0176
***GoToHunt* (Ours)**		0.0673	0.4471	0.0165
VGGT *(Base Model)*		0.0698	0.4953	0.0178

E Application on other architecture of visual geometry transformers

The visual geometry transformers, VGGT [85] and π^3 [89] that we mainly experiment with in the main paper, adopt global attention layers for cross-view information processing, which are also the major architecture that existing methods like FastVGGT [72], SparseVGGT [82] target to accelerate. To demonstrate the broad application of our method as a general plug-in solution for various types of visual geometry transformers, we additionally illustrate below that base models like Depth Anything 3 [53] which uses a different architecture (input-adaptive cross-view self-attention by token rearrangement) for cross-view information processing can also easily benefit from our solution. In Table L, we show that our method can also effectively boost the inference speed for Depth Anything 3 without performance compromise. In contrast, existing methods like SparseVGGT that require inspection on global attention maps for token compression and merging are not very straightforward to be adapted to models like Depth Anything 3.

F Detailed explanation on evaluation metrics

We provide detailed explanations of the evaluation metrics in the three tasks: camera pose estimation, 3D reconstruction, and video depth estimation as below.

Table H: Quantitative results on applying intra-frame downsampling on selected late layers determined by l_{late}. The experiments are conducted with $K = 25$ on the camera pose estimation on 7-Scenes [73]. l_{late} demonstrates inferior robustness than the choices of l_{local} and l_{sample}.

σ	l_{late}	**ATE (↓)**	**RPE-rot (↓)**	**RPE-trans (↓)**
VGGT *(Base Model)*		0.0698	0.4953	0.0178
2	18	0.0718	0.4496	0.0166
2	19	0.0687	0.4448	0.0165
2	20	0.0686	0.4460	0.0165
2	21	0.0681	0.4453	0.0165
2	22	0.0683	0.4458	0.0165
2	23	0.0683	0.4462	0.0165
3	18	0.0758	0.4585	0.0171
3	19	0.0702	0.4481	0.0167
3	20	0.0710	0.4494	0.0167
3	21	0.0699	0.4471	0.0167
3	22	0.0699	0.4473	0.0167
3	23	0.0698	0.4478	0.0166

Table I: Quantitative results on applying intra-frame downsampling on selected late layers determined by l_{late}. The experiments are conducted with $K = 25$ on the camera pose estimation on Neural RGB-D [3]. l_{late} demonstrates inferior robustness than the choices of l_{local} and l_{sample}.

σ	l_{late}	**ATE (↓)**	**RPE-rot (↓)**	**RPE-trans (↓)**
VGGT *(Base Model)*		0.0374	0.2934	0.0186
2	18	0.0315	0.2008	0.0170
2	19	0.0286	0.1842	0.0162
2	20	0.0277	0.1823	0.0163
2	21	0.0274	0.1818	0.0163
2	22	0.0273	0.1816	0.0163
2	23	0.0271	0.1820	0.0163
3	18	0.0412	0.3229	0.0236
3	19	0.0321	0.2540	0.0180
3	20	0.0311	0.2519	0.0181
3	21	0.0300	0.2463	0.0180
3	22	0.0297	0.2457	0.0179
3	23	0.0293	0.2451	0.0179

F.1 Camera pose estimation

Absolute Trajectory Error (ATE). ATE measures the global error between the estimated trajectory and the ground truth trajectory. Given estimated poses $\{\hat{T}_i\}$ and ground truth poses $\{T_i\}$, we compute the Root Mean Square Error (RMSE) between aligned camera positions:

$$\text{ATE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N} \|\mathbf{p}_i - \hat{\mathbf{p}}_i\|^2}, \tag{A}$$

where $\mathbf{p}_i$ and $\hat{\mathbf{p}}_i$ denote the ground truth and estimated camera centers after alignment. ATE captures long-term drift over the trajectory.

Relative Pose Error (RPE). RPE evaluates local motion consistency over a fixed interval Δ. In evaluation, we use interval $\Delta = 1$, indicating that we report the relative pose error between adjacent frames. Let $T_{i,i+\Delta} = T_i^{-1} T_{i+\Delta}$ and $\hat{T}_{i,i+\Delta} = \hat{T}_i^{-1} \hat{T}_{i+\Delta}$. Each relative transformation $T_{i,i+\Delta} \in$

Table J: Sensitivity analysis on the layer partition thresholds l_{local} and l_{sample} for intra-frame selection with $K = 25$ on 7-Scenes [73] using VGGT, with the *blue* row indicating our current parameter choice. Performance remains stable across a wide range of threshold choices, demonstrating that our method is robust to these hyperparameter choices, as long as they are consistent with the observed layer-wise attention patterns.

l_{local}	l_{sample}	**ATE (↓)**	**RPE-rot (↓)**	**RPE-trans (↓)**
VGGT *(Base Model)*		0.0698	0.4953	0.0178
1	8	0.0670	0.4433	0.0165
1	9	0.0676	0.4476	0.0167
1	10	0.0675	0.4494	0.0166
2	8	0.0671	0.4454	0.0164
2	9	0.0677	0.4495	0.0166
2	10	0.0677	0.4518	0.0165
3	8	0.0672	0.4501	0.0164
3	9	0.0678	0.4532	0.0166
3	10	0.0680	0.4539	0.0166
4	8	0.0674	0.4555	0.0167
4	9	0.0680	0.4589	0.0169
4	10	0.0681	0.4581	0.0167

$\mathrm{SE}(3)$ can be decomposed as:

$$T_{i,i+\Delta} = \begin{bmatrix} R_{i,i+\Delta} & \mathbf{t}_{i,i+\Delta} \\ 0 & 1 \end{bmatrix}, \tag{B}$$

where $R_{i,i+\Delta} \in \mathrm{SO}(3)$ denotes the relative rotation matrix and $\mathbf{t}_{i,i+\Delta} \in \mathbb{R}^3$ denotes the relative translation vector. **RPE-rot** captures the rotation error of the relative pose:

$$\text{RPE-rot} = \frac{1}{N} \sum_i \angle \left(R_{i,i+\Delta}^{-1} \hat{R}_{i,i+\Delta} \right), \tag{C}$$

where $\angle(\cdot)$ computes the rotation angle. **RPE-trans** captures the translation error of the relative pose:

$$\text{RPE-trans} = \frac{1}{N} \sum_i \|\mathbf{t}_{i,i+\Delta} - \hat{\mathbf{t}}_{i,i+\Delta}\|. \tag{D}$$

F.2 3D reconstruction

Accuracy (Acc). Accuracy measures how close the predicted point cloud $\hat{P}$ is to the ground truth point cloud P:

$$\text{Acc} = \frac{1}{|\hat{P}|} \sum_{\hat{p} \in \hat{P}} \min_{p \in P} \|\hat{p} - p\|. \tag{E}$$

Completeness (Comp). On the contrary to accuracy, completeness evaluates how well the ground truth geometry is covered by the prediction:

$$\text{Comp} = \frac{1}{|P|} \sum_{p \in P} \min_{\hat{p} \in \hat{P}} \|p - \hat{p}\|. \tag{F}$$

Normal Consistency (NC). Normal consistency measures the alignment between predicted and ground-truth surface normals. For each predicted point $\hat{p} \in \hat{P}$, we find its nearest neighbor $p \in P$ in the ground truth point cloud, and compute the cosine similarity between their normals:

$$\text{NC} = \frac{1}{|\hat{P}|} \sum_{\hat{p} \in \hat{P}} \langle \mathbf{n}(p), \mathbf{n}(\hat{p}) \rangle, \tag{G}$$

where $\mathbf{n}(\hat{p})$ and $\mathbf{n}(p)$ denote the predicted and ground truth normals at $\hat{p}$ and its nearest neighbor p, respectively.

Table K: Comparison on camera pose estimation and inference time on 7-Scenes with varying sequence lengths. Inputs are chronological prefixes of the scenes. The two shorter sequences are excluded at 300/400 frames. Best and second-best accelerated methods are **bold** and underlined. Our *GoToHunt* uses $K = 25$, $\sigma = 3$.

Method	ATE (↓)	RPE-rot (↓)	RPE-trans (↓)	Time (↓)
100 frames				
VGGT *(Base Model)*	0.0293	0.3179	0.0095	8.12s
FastVGGT	0.0301	0.3399	0.0107	8.15s
SparseVGGT (SR: 50%)	0.0329	0.3683	0.0121	5.88s
SparseVGGT (SR: 75%)	0.0345	0.4596	0.0156	5.03s
Co-Me	0.0367	0.4594	0.0187	3.70
LiteVGGT	**0.0285**	0.3648	0.0122	**3.02s**
***GoToHunt* (Ours)**	0.0289	**0.3223**	**0.0099**	6.96s
200 frames				
VGGT *(Base Model)*	0.0563	0.3919	0.0131	25.57s
FastVGGT	0.0582	0.4055	0.0140	18.51s
SparseVGGT (SR: 50%)	0.0624	0.4691	0.0164	15.81s
SparseVGGT (SR: 75%)	0.0659	0.5977	0.0218	12.25s
Co-Me	0.0742	0.6494	0.0281	10.38s
LiteVGGT	0.0528	0.4344	0.0160	**7.23s**
***GoToHunt* (Ours)**	**0.0525**	**0.3780**	**0.0130**	14.10s
300 frames				
VGGT *(Base Model)*	0.0651	0.4898	0.0147	51.83s
FastVGGT	0.0687	0.4585	0.0153	31.58s
SparseVGGT (SR: 50%)	0.0733	0.5543	0.0176	30.48s
SparseVGGT (SR: 75%)	0.0829	0.7012	0.0237	22.26s
Co-Me	0.0883	0.7937	0.0294	20.27s
LiteVGGT	**0.0590**	0.4977	0.0175	**12.84s**
***GoToHunt* (Ours)**	0.0624	**0.4450**	**0.0146**	21.29s
400 frames				
VGGT *(Base Model)*	0.0724	0.5066	0.0165	87.42s
FastVGGT	0.0763	0.4627	0.0163	46.62s
SparseVGGT (SR: 50%)	0.0862	0.5959	0.0192	49.29s
SparseVGGT (SR: 75%)	0.0932	0.7539	0.0256	34.36s
Co-Me	0.0956	0.9304	0.0337	33.26s
LiteVGGT	0.0685	0.7061	0.0202	**19.23s**
***GoToHunt* (Ours)**	**0.0678**	**0.4482**	**0.0154**	28.49s

Table L: Application to Depth Anything 3 Large (DA3-Large) on 7-Scenes. *GoToHunt* uses $K = 25$ and $\sigma = 2$. Inference time is measured with 500-frame sequences.

Method	ATE (↓)	RPE-rot (↓)	RPE-trans (↓)	Time (↓)
DA3-Large	0.0692	0.4259	**0.0136**	96.5s
***GoToHunt* on DA3-Large (Ours)**	**0.0677**	**0.4177**	0.0139	**29.2s**

F.3 Video depth estimation

Let d_i and $\hat{d}_i$ denote the ground-truth and predicted depths at pixel i, respectively, and N be the number of valid pixels.

Absolute Relative Error (Abs Rel). Abs Rel measures the absolute error normalized by the ground-truth depth. It emphasizes relative accuracy, assigning larger penalties to errors at closer depths where

precision is typically more important:

$$\text{Abs Rel} = \frac{1}{N}\sum_i \frac{|d_i - \hat{d}_i|}{d_i}. \tag{H}$$

Squared Relative Error (Sq Rel). Sq Rel penalizes large errors more heavily due to the squared term. This makes it more sensitive to outliers and particularly useful for evaluating robustness when large depth errors occur:

$$\text{Sq Rel} = \frac{1}{N}\sum_i \frac{(d_i - \hat{d}_i)^2}{d_i}. \tag{I}$$

Root Mean Square Error (RMSE). RMSE measures the absolute discrepancy in depth values and strongly penalizes large deviations. It reflects overall reconstruction fidelity in the original depth scale, but can be dominated by errors in distant regions where depth values are larger:

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_i (d_i - \hat{d}_i)^2}. \tag{J}$$

Log RMSE. By operating in log space, Log RMSE reduces sensitivity to absolute scale and instead emphasizes relative differences. It is particularly suitable when the depth range is large, as it balances errors across near and far regions and better reflects perceptual quality:

$$\text{Log RMSE} = \sqrt{\frac{1}{N}\sum_i (\log d_i - \log \hat{d}_i)^2}. \tag{K}$$

Threshold Accuracy ($\delta < 1.25$). This accuracy metric is the percentage of pixels satisfying $\delta_i < 1.25$, where

$$\delta_i = \max\left(\frac{d_i}{\hat{d}_i}, \frac{\hat{d}_i}{d_i}\right). \tag{L}$$

This metric evaluates the fraction of predictions that fall within a multiplicative error bound of the ground truth. It provides an intuitive measure of reliability, indicating how many predictions are sufficiently accurate rather than averaging errors.

G Licenses for existing assets

The following list contains licenses for data and model used in the paper:

- VGGT [85]: View the link for details
  https://github.com/facebookresearch/vggt
- π^3 [89]: BSD-3-Clause License
  https://github.com/yyfz/Pi3
- 7-Scenes [73]: View the link for details
  https://www.microsoft.com/en-us/research/project/rgb-d-dataset-7-scenes/
- Neural RGB-D [3]: View the link for details
  https://github.com/dazinovic/neural-rgbd-surface-reconstruction
- Bonn [63]: Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported License
  https://www.ipb.uni-bonn.de/data/rgbd-dynamic-dataset/index.html
- TUM-Dynamics [75]: CC-BY-4.0 License
  https://cvg.cit.tum.de/data/datasets/rgbd-dataset

The license names can also be found in the above links. As some datasets contain different licenses for different scenes, we do not list them here one by one to save space.

H Additional discussions

Recently, concurrent works [23, 37, 81, 97, 110] have explored replacing the quadratic-time attention with test-time training (TTT) [79] layers to achieve linear-time scaling in visual geometry transformers. While these approaches substantially improve inference efficiency, they typically incur a performance compromise compared to standard attention-based architectures and require considerable computational resources to train the TTT layers. In contrast, our method is fully *training-free* and can be readily applied as a plug-in to a wide range of existing models. Therefore, our approach is complementary to these methods and can potentially be integrated with them for further efficiency improvements, and we leave this as an interesting future work.

I Societal impact

We expect our work to have an overall positive societal impact. By accelerating visual geometry transformers, our method improves the accessibility of advanced 3D reconstruction technologies, enabling individual researchers and smaller organizations with limited computational resources to utilize these visual geometry transformers more effectively. In this sense, improving efficiency can contribute to a more equitable and democratized AI ecosystem [46], where high-quality 3D reconstruction systems are not exclusively accessible to institutions with extensive computational infrastructure.

More efficient visual geometry transformers may also benefit a broad range of downstream applications with positive societal value, such as digital cultural heritage preservation. In particular, reducing inference cost facilitates deployment on resource-constrained platforms and edge devices, enabling efficient deployment that were previously impractical due to computational limitations.

Potential negative societal impact. We do not identify obvious negative societal impact from our work. Potential risks may include misuse for malicious purposes such as unlawful surveillance that harms individual privacy.