TRADE: TRADING CLUSTER DISTORTION FOR EFFICIENT VISUAL GEOMETRY TRANSFORMERS

Anonymous authors
Paper under double-blind review

ABSTRACT

Visual geometry foundation models enable powerful feed-forward 3D reconstruction, but their reliance on global attention leads to rapidly growing computational costs for long input sequences. Existing training-free token reduction methods alleviate this cost, yet typically determine token selection and retention through predefined regular structures, overlooking the heterogeneous redundancy of token representations. We observe that token similarity is highly non-uniform across input sequences and spatial regions and dynamically evolves with network depth. Based on this observation, we propose TraDE, a training-free framework that adaptively determines both which tokens to retain and how many are needed according to their underlying representations. TraDE formulates token reduction as iterative clustering and uses cluster distortion to balance representation fidelity against the number of retained tokens, while stage-wise repartitioning accommodates evolving representations across depth. Extensive experiments across multiple visual geometry foundation models, datasets, and geometric tasks demonstrate favorable performance–efficiency trade-offs, achieving up to $8.0\times$ inference acceleration on long sequences. Code is provided in the Supplementary Material.

1 INTRODUCTION

Recovering the 3D geometry of a scene is a central problem in 3D vision, as it enables visual systems to infer the spatial structure of the physical world from image observations Hartley et al. (2003). Recent advances are driving a transition from conventional iterative, optimization-based geometric pipelines Triggs et al. (1999); Schonberger & Frahm (2016); Mur-Artal et al. (2015) toward unified feed-forward geometry foundation models. Building upon early feed-forward geometry models Wang et al. (2024); Leroy et al. (2024), recent approaches Wang et al. (2025; 2026d); Lin et al. (2025) have further advanced toward unified visual geometry foundation models, enabling comprehensive scene geometry recovery directly from multi-view images within a single forward pass. Despite their strong geometric reasoning capabilities, these models incur rapidly growing computational and memory costs as the number of input images increases, making efficient inference a critical challenge for scaling geometry foundation models to long input sequences.

To establish geometric relationships across multiple frames, visual geometry Transformers typically rely on global attention for cross-frame information exchange, whose computational cost scales quadratically with the number of input frames. A natural way to alleviate this cost is to exploit token redundancy by removing highly similar tokens with redundant representations before global attention. Existing methods Shen et al. (2026); Sun et al. (2026); Wang et al. (2026c) typically rely on regular structure and fixed retention ratio to remove redundant tokens, which implicitly assumes a homogeneous distribution of token similarity. However, token similarity is inherently heterogeneous across different input sequences, spatial regions, and network depths. As illustrated in Figure 1, for different reference tokens, their similar tokens are non-uniformly distributed across different regions and frames, and such distributions dynamically change with network depth. Therefore, such heterogeneity calls for a token reduction strategy that adaptively determines how many tokens to retain and which tokens to preserve according to the underlying token representations.

To characterize the heterogeneity of token similarity, we summarize it from two aspects: *input-wise non-uniformity* and *depth-wise dynamics*. As illustrated in Figure 1, *input-wise non-uniformity* is reflected by the non-uniform distribution of similar tokens across different sequences and image

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.



Figure 1: Token-similarity heterogeneity on 7Scenes. For two reference tokens, tokens with cosine similarity above 0.9 are shown at Layers 0, 11, and 23. Their distributions vary across regions, sequences, and network depth, highlighting both input-wise non-uniformity and depth-wise dynamics.

regions for different reference tokens. Meanwhile, *depth-wise dynamics* indicates that these similarity distributions dynamically change with network depth as token representations are updated throughout the model. Motivated by these observations, we propose trading cluster distortion for efficient visual geometry Transformers (TraDE), a training-free framework that starts from singleton token clusters and iteratively merges similar clusters, retaining one representative token per cluster for global attention. More importantly, TraDE explicitly quantifies the representation discrepancy introduced by representing each cluster with a single representative token through cluster distortion, and combines this distortion with the number of retained clusters in a cluster-merging objective, enabling an adaptive trade-off between representation fidelity and inference efficiency. Specifically, TraDE iteratively searches for candidate clusters based on local spatio-temporal relations, selects the optimal merging pair, and performs cluster merging according to the merging objective, while updating the representative tokens and cluster assignments. Furthermore, motivated by the stage-wise behavior of visual geometry Transformers Sun et al. (2026), we perform token clustering at the first global attention layer of each stage and reuse the partition in the remaining layers, enabling TraDE to accommodate depth-wise dynamics. Since the clustering process predefines neither the number of clusters nor their representative tokens, TraDE can adaptively determine how many tokens to retain and which tokens to retain according to the similarity structure of token representations.

Our contributions are summarized as follows:

- We reveal pronounced heterogeneity in token similarity in visual geometry Transformers, characterized by input-wise non-uniformity and depth-wise dynamics. To account for such heterogeneity, we introduce cluster distortion to quantify representation discrepancy and combine it with the number of retained tokens to formulate the cluster merging objective.
- We propose TraDE, a training-free token reduction framework that iteratively merges token clusters under the cluster-merging objective, thereby adaptively determining which tokens to retain and how many are needed from the underlying token representations, while stage-wise repartitioning updates the partition to accommodate dynamics across depth.
- Extensive experiments across multiple visual geometry foundation models, datasets, and geometric tasks demonstrate that TraDE achieves favorable performance–efficiency trade-offs, with up to $8.0\times$ inference acceleration on long input sequences.

2 Related Work

2.1 Visual Geometry Transformers

Feed-forward visual geometry models replace conventional iterative optimization pipelines Triggs et al. (1999); Schonberger & Frahm (2016); Mur-Artal et al. (2015) by directly predicting scene geometry from image observations in a single forward pass. As a pioneering work in this paradigm, DUSt3R Wang et al. (2024) directly regresses pairwise point maps with a standard Transformer, without requiring camera calibration or pose priors, and subsequent works Yang et al. (2025); Cabon et al. (2025) extend this paradigm to one-shot multi-view reconstruction. Following the paradigm of unified multi-view inference, VGGT Wang et al. (2025) performs feed-forward prediction of

comprehensive 3D scene geometry in a reference coordinate frame, including camera poses, depth maps, and point maps. Depth Anything 3 Lin et al. (2025) further simplifies both the architecture and prediction target by using a single standard ViT together with unified depth and ray prediction. To remove the inductive bias introduced by a designated reference view, π^3 Wang et al. (2026d) adopts a permutation-equivariant formulation that predicts relative camera poses and scale-invariant local point maps. VGGT-Ω Wang et al. (2026b) simplifies VGGT through a single dense prediction head and register tokens that aggregate global scene information. Beyond static multi-view reconstruction, feed-forward geometry models have expanded toward heterogeneous inputs Keetha et al. (2026); Karhade et al. (2026), streaming reconstruction Yuan et al. (2026); Zhuo et al. (2026), and dynamic scene modeling Zhou et al. (2026); Fang et al. (2026), broadening their applicability to general and realistic visual environments. However, visual geometry Transformers rely on token-level global attention for cross-view reasoning, whose quadratic complexity limits scalability to long sequences. This motivates reducing tokens in global attention while preserving geometric fidelity.

2.2 Efficient Inference for Visual Geometry Transformers

Improving the inference efficiency of visual geometry Transformers has been extensively studied through token reduction Shen et al. (2026); Wang et al. (2026c), sparse global attention Wang et al. (2026a); Sun et al. (2026); Deng et al. (2026), and compact scene representations for scalable long-sequence inference Elflein et al. (2026); Jin et al. (2026); Xie et al. (2026). Among these directions, token reduction provides a natural and direct way to accelerate global attention by operating directly on the participating tokens. FastVGGT Shen et al. (2026) introduces training-free token merging with regularly sampled tokens and predefined merging ratios. Co-Me Chen et al. (2026) and LiteVGGT Shu et al. (2026) incorporate confidence and geometric cues into token selection, while HTTM Wang et al. (2026c) performs head-wise token merging by jointly exploiting spatial locality and temporal correspondence. Recent studies have also begun to account for depth-wise variation, with RegimeVGGT You et al. (2026) and Spark3R Tang et al. (2026) exploiting layer-wise characteristics and asymmetric query–key/value redundancy to assign different reduction strategies across network depth. In general vision Transformers, several methods achieve adaptive token reduction through representation clustering Marin et al. (2023), attention from task-specific tokens Lee et al. (2026), or calibration-based redundancy estimation Fayyaz et al. (2022). However, existing visual geometry methods have not comprehensively characterized token heterogeneity across inputs and depth, and still rely on predefined compression ratios and regular sampling structures. Meanwhile, adaptive methods for general vision Transformers often depend on architecture- or input-specific designs, limiting their transfer to long sequence geometry inputs. To this end, we propose TraDE, which uses a cluster distortion objective with a token-count penalty to adaptively determine both token selection and reduction extent directly from the underlying representations.

3 Method

3.1 Heterogeneity of Token Similarity

Reducing similar tokens before global attention is a direct way to alleviate the computational cost of visual geometry Transformers, especially for long sequences with representation redundancy. However, we observe that token similarity is highly heterogeneous rather than uniformly distributed across inputs, image regions, and network depth. As qualitatively illustrated in Figure 1 and quantitatively verified in Figure 2, this heterogeneity can be summarized into two characteristics: *input-wise non-uniformity* and *depth-wise dynamics*.

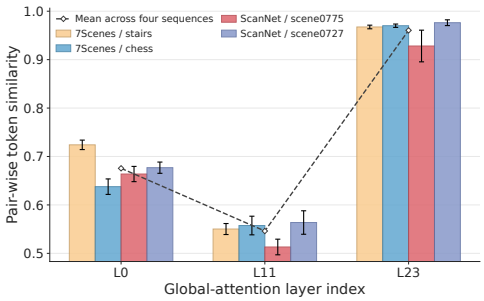


Figure 2: Quantitative analysis of token-similarity across four scene sequences and three layers using 300-frame inputs, including intra-sequence standard deviation.

Input-wise non-uniformity refers to the variation of token similarity across inputs and tokens. As shown in Figure 2, the varying bar heights across sequences reveal sequence-level differences in token similarity, while the error bars indicate considerable variation across tokens within each se-

quence. Regions containing repeated visual content or frames with similar viewpoints generally exhibit higher similarity, whereas newly observed content or larger viewpoint changes produce more distinctive token representations. *Depth-wise dynamics* refers to the dynamic variation of token similarity across network depth. The dashed trend in Figure 2 connects the four-sequence mean similarity at the three measured global-attention layers, showing that the overall similarity level changes as token representations are progressively updated. Predefined regular structures cannot accommodate this heterogeneity, motivating token reduction to adapt directly to token representations.

3.2 Trading Cluster Distortion for Efficiency

Cluster distortion. Let $\{\mathbf{x}_i\}_{i=1}^{N}$ denote the reducible patch-token representations entering global self-attention, and let $\mathbf{z}_i = \mathbf{x}_i/\|\mathbf{x}_i\|_2$ denote their normalized features. We initialize each token as an individual cluster and iteratively merge clusters with similar token representations, yielding a partition $\mathcal{P} = \{G_1, \ldots, G_K\}$. For each cluster G, we select a representative index $r_G \in G$ to replace the remaining cluster members in global self-attention, thereby reducing the original N tokens to K representatives. Given the heterogeneity discussed above, the representation discrepancy introduced by this replacement also varies across token clusters. We therefore explicitly use *cluster distortion* Linde et al. (1980); Banerjee et al. (2005) to quantify the discrepancy between a representative token and the cluster members it represents:

$$\begin{aligned} \mathbf{u}_G &= \sum_{i \in G} \mathbf{z}_i, \qquad w_G = |G|, \\ E(G, r_G) &= \sum_{i \in G} \left(1 - \mathbf{z}_i^\top \mathbf{z}_{r_G}\right) = w_G - \mathbf{u}_G^\top \mathbf{z}_{r_G}, \end{aligned} \tag{1}$$

where $\mathbf{u}_G$ and w_G denote the summed normalized feature and cluster size, respectively. These statistics enable efficient incremental updates of $E(G, r_G)$ without revisiting all cluster members.

Cluster Merging Objective. Minimizing cluster distortion alone yields the trivial solution in which each token forms an individual cluster, preserving the original representations but achieving no token reduction. We therefore balance cluster distortion against the number of retained clusters by introducing the latter as a penalty term:

$$\mathcal{J}(\mathcal{P}) = \frac{\sum_{G \in \mathcal{P}} E(G, r_G)}{2N} + \lambda \frac{K}{N}, \tag{2}$$

where the first term measures cluster distortion, K/N denotes the retention ratio among reducible patch tokens, and λ controls their trade-off. This objective captures how reducing the number of retained tokens affects cluster distortion. As a result, the retention ratio is not predefined, but is determined adaptively by the representations. Clusters can be merged more aggressively when the distortion increase is small, while merging is discouraged when it causes a large discrepancy.

3.3 Iterative Token Clustering

We now present the iterative token clustering procedure of TraDE, which repeatedly performs four steps, as illustrated in Figure 3, until the merge criterion in Equation 6 is no longer satisfied.

Step 1: Candidate Cluster Search. For each cluster, we first search for candidate clusters that may be merged with it, namely those likely to contain similar tokens. Exhaustively searching for the most similar tokens over the entire image sequence would incur substantial computational overhead, especially for long inputs. Inspired by the spatio-temporal redundancy exploited in HTTM Wang et al. (2026c), we restrict the search for similar tokens to local spatio-temporal neighborhoods. Specifically, for a token at frame f and spatial position (u, v), its neighborhood is defined as:

$$|f - f'| \le t, \qquad |u - u'| \le \rho, \qquad |v - v'| \le \rho, \tag{3}$$

where t and ρ denote the temporal and spatial search ranges, respectively. Two clusters are treated as candidate clusters if any pair of their member tokens satisfies Equation 3. As clustering proceeds, a merged cluster inherits the candidate relations of its constituent clusters, allowing the effective search range to expand iteratively beyond the initially specified t and ρ.

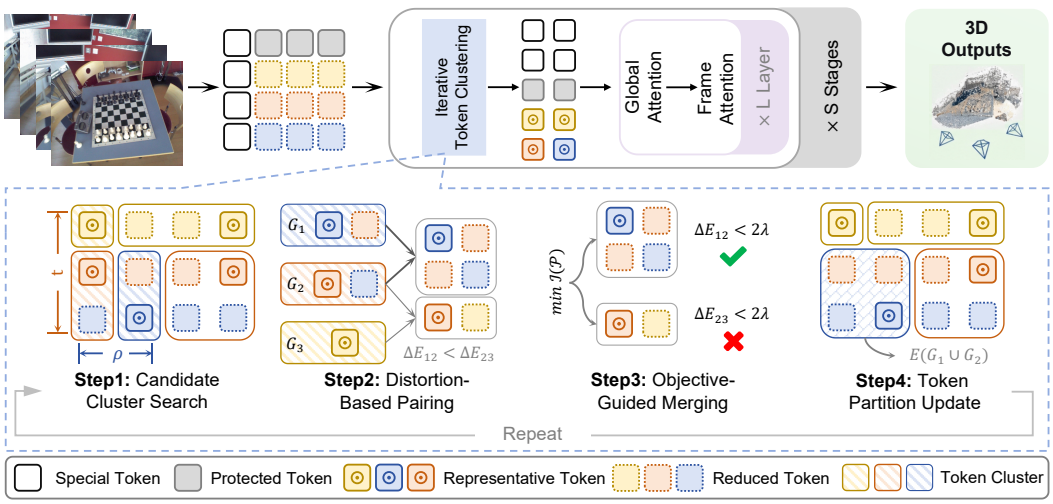


Figure 3: Overview of TraDE. The iterative process is illustrated using three token clusters as an example. TraDE performs candidate cluster search, distortion-based pairing, objective-guided merging, and token partition update. The resulting token partition is reused within each stage and recomputed across stages with L layers per stage and S stages in total.

Step 2: Distortion-based Pairing. For each cluster G_i, we establish a potential merging pair by selecting the candidate cluster that introduces the minimum distortion change. Specifically, for each candidate cluster G_j, we evaluate the additional distortion caused by merging G_i and G_j. Given two clusters G_i and G_j, the merged cluster first updates the normalized feature statistics and cluster size in Equation 1, resulting in $E(G_i \cup G_j, r_{G_i \cup G_j})$, where $r_{G_i \cup G_j}$ denotes the representative token of the merged cluster. To efficiently determine the representative token, we select it from the two existing representatives r_{G_i} and r_{G_j} and compute the corresponding distortion change:

$$\begin{aligned} r_{G_i \cup G_j} &= \arg \min_{r \in \{r_{G_i}, r_{G_j}\}} E(G_i \cup G_j, r), \\ \Delta E(G_i, G_j) &= E(G_i \cup G_j, r_{G_i \cup G_j}) - E(G_i, r_{G_i}) - E(G_j, r_{G_j}), \\ p_i &= \arg \min_{j: G_j \in \mathcal{C}_i} \Delta E(G_i, G_j). \end{aligned} \tag{4}$$

We denote the set of candidate clusters for G_i by $\mathcal{C}_i$. A smaller $\Delta E(G_i, G_j)$ indicates that merging the two clusters introduces less additional representation discrepancy. Moreover, only mutually selected cluster pairs are considered valid merging pairs for the subsequent merging step.

Step 3: Objective-Guided Merging. The previous pairing step identifies a potential merge pair (G_i, G_{p_i}) for each cluster. However, different pairs introduce different amounts of distortion after merging, and therefore not all candidate pairs should be merged. Based on the merging objective defined in Equation 2, we perform merging only when it leads to a reduction of the objective function. Specifically, merging a candidate pair (G_i, G_{p_i}) introduces a distortion change ΔE while reducing the number of clusters by one, resulting in the following change of the objective:

$$\Delta \mathcal{J}(G_i, G_{p_i}) = \frac{\Delta E(G_i, G_{p_i})}{2N} - \frac{\lambda}{N}. \tag{5}$$

Accordingly, the pair is merged only if it decreases the objective:

$$\Delta \mathcal{J}(G_i, G_{p_i}) < 0 \quad \Longleftrightarrow \quad \Delta E(G_i, G_{p_i}) < 2\lambda. \tag{6}$$

Thus, a potential pair is merged only when the additional cluster distortion is sufficiently small to justify reducing one retained token. All valid pairs from the previous pairing step that satisfy Equation 6 are merged in parallel, since these pairs are mutually selected and therefore disjoint. Additionally, the criterion in Equation 6 also serves as the termination condition of the iterative clustering process. When none of the valid candidate pairs satisfies this criterion, further merging would increase the objective function, and the iterative token clustering procedure is therefore terminated.

Step 4: Token Partition Update. After merging the cluster pairs, the token partition, i.e., the assignment of each original token to a cluster, is updated accordingly. For an accepted merge between G_i and its paired clusterG_{p_i}, the corresponding partition update is given by:

$$\mathcal{P}^{(k+1)} = \left(\mathcal{P}^{(k)} \setminus \{G_i, G_{p_i}\}\right) \cup \{G_i \cup G_{p_i}\}, \tag{7}$$

where $\mathcal{P}^{(k)}$ and $\mathcal{P}^{(k+1)}$ denote the cluster partitions before and after the k-th iteration, respectively. The representative token, cluster statistics, and distortion of the newly merged cluster are updated according to Equation 1 and Equation 4. The updated partition is then fed back to Step 1 to reconstruct candidate cluster relations for the next iteration.

Stage-wise Repartitioning. Token similarity evolves with network depth as token representations are progressively updated, making a single token partition insufficient for all global attention layers. However, recomputing the partition at every layer would introduce clustering overhead. Motivated by the stage-wise behavior of global attention in visual geometry Transformers Sun et al. (2026), TraDE performs iterative clustering only at the first global attention layer of each stage and reuses the partition throughout that stage. The partition is then recomputed at the next stage boundary, allowing token reduction to adapt to the changing functions of global attention layers across depth. Following prior work Shen et al. (2026); Shu et al. (2026), camera tokens, register tokens, and reference-frame patch tokens are excluded from clustering, while representative outputs are propagated to their cluster members after global attention to restore the original token layout.

4 Experiments

4.1 Implementation Details

Experiments are conducted on three representative visual geometry foundation models, including VGGT Wang et al. (2025), π^3 Wang et al. (2026d), and VGGT-Ω Wang et al. (2026b), with camera pose estimation and point-cloud reconstruction evaluated on ScanNet, 7Scenes, and NRGBD. For ScanNet, 50 sequences are selected from the test split, and frames are uniformly sampled with a stride of 2 to construct 100-, 300-, 500-, and 1000-frame inputs, while 7Scenes and NRGBD are sampled with a stride of 3. Camera pose estimation is evaluated using AUC, and point-cloud reconstruction using Accuracy (Acc), Completeness (Comp), and Normal Consistency (NC). Inference efficiency is measured by frames per second (FPS), acceleration ratio (Spd.) relative to the corresponding dense backbone, and allocated GPU memory (Mem.). The trade-off coefficient is set to $\lambda = 0.04$, with temporal neighborhood size $t = 4$ and spatial neighborhood radius $\rho = 2$ throughout the experiments. Stage-wise repartitioning is performed at layers $\{0, 9, 21\}$ for VGGT, $\{0, 9\}$ for π^3, and $\{0, 10, 17\}$ for VGGT-Ω. For VGGT, the VRAM-efficient implementation provided by FastVGGT Shen et al. (2026) is adopted for long-sequence inference. All experiments are conducted on a single NVIDIA RTX PRO 6000 GPU with 96 GB of memory.

4.2 Camera Pose Estimation

Table 1 reports camera pose estimation results on ScanNet under different sequence lengths and backbones. As the input length increases from 100 to 1000 frames, the acceleration ratio of TraDE consistently increases across all three backbones. At 1000 frames, TraDE achieves $7.25\times$, $8.05\times$, and $4.47\times$ acceleration on VGGT, π^3, and VGGT-Ω, respectively, compared with $3.95\times$, $4.98\times$, and $4.15\times$ for HTTM, the second-fastest executable method in these settings. Compared with the dense backbones, TraDE decreases AUC@3 by 1.42 on average across all backbone and sequence-length settings, while AUC@30 remains nearly unchanged, with an average change of only -0.03. In terms of memory usage, TraDE maintains memory consumption close to the corresponding dense backbone across all three architectures. Under the 1000-frame VGGT setting, SparseVGGT and Co-Me encounter OOM, whereas TraDE completes inference with 69.44 GiB of GPU memory.

Results on 7Scenes and NRGBD are reported in Table 3. TraDE achieves AUC@30 values of 79.03 and 95.63 with $3.95\times$ and $4.15\times$ acceleration, respectively, outperforming the second-fastest HTTM at $2.92\times$ and $2.93\times$. On 7Scenes, TraDE maintains pose accuracy close to dense VGGT, while on NRGBD it achieves the best results across all three AUC metrics. Overall, TraDE consistently provides the best efficiency while preserving competitive camera pose accuracy.

Table 1: ScanNet comparison using AUC@3, AUC@30, acceleration ratio (Spd.), and allocated GPU memory (Mem.) under 100-, 300-, 500-, and 1000-frame input settings. Spd. is measured relative to the corresponding backbone baseline; Mem. is reported in GiB, where lower is better.

Method	100 Frames				300 Frames				500 Frames				1000 Frames			
	A_3 ↑	A_{30} ↑	Spd.↑	Mem.↓	A_3 ↑	A_{30} ↑	Spd.↑	Mem.↓	A_3 ↑	A_{30} ↑	Spd.↑	Mem.↓	A_3 ↑	A_{30} ↑	Spd.↑	Mem.↓
VGGT CVPR'25	**12.49**	**68.84**	1.00	10.53	15.76	76.20	1.00	**27.27**	13.29	74.25	1.00	**44.00**	12.20	73.68	1.00	69.10
FastVGGT ICLR'26	11.81	68.24	1.64	11.46	**16.03**	**77.10**	2.31	30.03	**14.96**	77.10	2.58	48.61	**15.33**	79.89	2.64	76.46
SparseVGGT CVPR'26	9.03	66.51	1.45	**10.03**	12.92	74.24	1.74	27.78	10.50	72.17	1.84	48.83	-	-	-	OOM
Co-Me CVPR'26	9.19	61.73	1.28	13.07	9.51	69.42	1.94	31.17	8.00	66.49	2.23	49.37	-	-	-	OOM
HTTM CVPR'26	11.25	66.96	1.72	12.01	12.01	70.54	3.26	31.56	10.81	70.17	4.19	51.07	8.92	69.19	3.95	**60.91**
TraDE	11.22	67.93	**2.41**	10.55	15.60	77.03	**4.76**	27.41	14.63	**77.50**	**6.43**	44.27	14.94	**79.91**	**7.25**	69.44
π^3 ICLR'26	13.85	**73.19**	1.00	10.08	**21.97**	**81.57**	1.00	19.52	**23.53**	**83.84**	1.00	**28.96**	**23.57**	**85.41**	1.00	58.25
FastVGGT ICLR'26	**13.87**	72.45	1.56	10.08	19.34	80.49	2.34	19.52	20.11	82.62	2.74	**28.96**	19.64	84.07	3.18	58.26
SparseVGGT CVPR'26	13.05	71.76	1.31	10.08	18.75	80.40	1.63	19.71	18.96	82.19	1.76	31.42	18.81	83.94	1.89	69.87
HTTM CVPR'26	13.17	71.05	1.49	**10.07**	19.04	80.12	2.80	**19.51**	19.71	82.22	3.67	**28.96**	19.06	83.25	4.98	**52.57**
TraDE	13.44	71.89	**1.96**	10.08	17.96	79.60	**3.85**	19.52	18.22	81.80	**5.34**	28.97	17.89	83.06	**8.05**	58.27
VGGT-Ω CVPR'26	**13.64**	**70.65**	1.00	11.59	**18.10**	**78.80**	1.00	**22.91**	**17.96**	**80.15**	1.00	**34.24**	**16.77**	**81.55**	1.00	64.15
FastVGGT ICLR'26	13.16	70.07	1.36	12.15	17.01	78.30	1.99	24.53	16.96	79.78	2.34	36.91	15.71	81.27	3.07	69.47
SparseVGGT CVPR'26	13.05	69.99	1.23	12.51	16.83	78.40	1.56	26.48	16.92	79.14	1.67	41.59	15.44	80.99	1.83	84.31
HTTM CVPR'26	12.18	68.94	1.30	**11.58**	15.60	77.06	2.30	22.98	15.54	78.49	3.02	34.39	13.35	78.62	4.15	**62.87**
TraDE	13.08	69.96	**1.41**	12.04	16.81	78.17	**2.45**	24.23	16.77	79.78	**3.22**	36.47	15.56	81.17	**4.47**	67.27

4.3 Point-Cloud Reconstruction

Table 2 reports point-cloud reconstruction results on ScanNet under different sequence lengths and backbones. Across all backbone and sequence-length settings, TraDE improves Acc by 0.001 on average, while Comp remains nearly unchanged and NC decreases by only 0.003, indicating that the substantially reduced inference cost introduces only minor changes in reconstruction quality. The reconstruction behavior varies across backbones. On VGGT, TraDE generally improves Acc and Comp over dense inference, particularly for long sequences; at 1000 frames, Acc/Comp/NC change from $0.042/0.030/0.842$ to $0.032/0.027/0.870$. On π^3 and VGGT-Ω, TraDE introduces moderate degradation in reconstruction quality, with NC decreasing from 0.927 to 0.919 and from 0.888 to 0.880 at 1000 frames, respectively. These changes are accompanied by substantially higher acceleration ratios, reaching $7.25\times$, $8.05\times$, and $4.47\times$ on the three backbones. Compared with HTTM, TraDE also achieves higher acceleration and generally better reconstruction quality in the long-sequence settings, especially under the most demanding 1000-frame inputs.

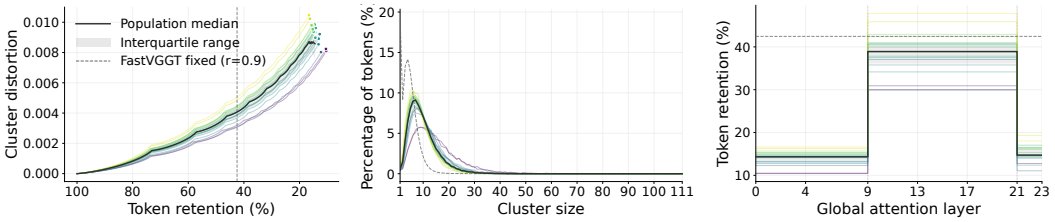
Table 3 further reports reconstruction results on 7Scenes and NRGBD. On 7Scenes, TraDE obtains Acc/Comp/F1/NC of $0.029/0.027/0.895/0.817$, remaining close to the strongest baselines. On NRGBD, TraDE achieves $0.0238/0.0255/0.9366/0.8820$, with the best Acc and F1 among all methods. Overall, TraDE preserves competitive reconstruction quality while substantially improving inference efficiency across both datasets, despite its notably higher acceleration ratios.

4.4 Analysis of Adaptivity

Figure 4 further analyzes the adaptive behavior of TraDE from the sequence, cluster, and layer perspectives. As shown in Figure 4(a), cluster distortion increases progressively as the token-retention ratio decreases. More importantly, different sequences exhibit distinct distortion–retention curves and terminate at different retention ratios under the same merging criterion, indicating substantial

Table 2: Point-cloud reconstruction results on ScanNet with 100, 300, 500, and 1000 input frames. Acc, Comp, and NC denote mean accuracy, completeness, and normal consistency, respectively. Spd. is the acceleration ratio relative to the corresponding backbone baseline.

Method	100 Frames				300 Frames				500 Frames				1000 Frames			
	Acc↓	Comp↓	NC↑	Spd.↑	Acc↓	Comp↓	NC↑	Spd.↑	Acc↓	Comp↓	NC↑	Spd.↑	Acc↓	Comp↓	NC↑	Spd.↑
VGGT CVPR'25	**0.018**	**0.019**	**0.888**	1.00	0.024	0.022	0.896	1.00	0.036	0.029	0.877	1.00	0.042	0.030	0.842	1.00
FastVGGT ICLR'26	**0.018**	**0.019**	0.885	1.64	**0.021**	**0.021**	**0.897**	2.31	0.032	0.027	**0.889**	2.58	**0.032**	**0.027**	**0.875**	2.64
SparseVGGT CVPR'26	0.020	**0.019**	0.853	1.45	0.027	0.022	0.853	1.74	0.044	0.037	0.838	1.84	-	-	-	-
Co-Me CVPR'26	0.027	0.024	0.823	1.28	0.037	0.028	0.827	1.94	0.050	0.040	0.812	2.23	-	-	-	-
HTTM CVPR'26	0.019	**0.019**	0.871	1.72	0.035	0.028	0.851	3.26	0.040	0.029	0.839	4.19	0.052	0.056	0.813	3.95
TraDE	**0.018**	**0.019**	0.879	**2.41**	**0.021**	**0.021**	0.893	**4.76**	**0.030**	**0.025**	0.886	**6.43**	**0.032**	**0.027**	0.870	**7.25**
π^3 ICLR'26	**0.015**	**0.019**	**0.914**	1.00	**0.018**	**0.021**	**0.925**	1.00	**0.020**	**0.023**	**0.929**	1.00	**0.023**	**0.025**	**0.927**	1.00
FastVGGT ICLR'26	0.016	**0.019**	0.911	1.56	0.019	0.022	0.920	2.34	0.021	**0.023**	0.924	2.74	0.024	**0.025**	0.921	3.18
SparseVGGT CVPR'26	0.021	0.020	0.846	1.31	0.025	0.023	0.862	1.63	0.028	0.025	0.870	1.76	0.030	0.026	0.880	1.89
HTTM CVPR'26	0.018	0.020	0.903	1.49	0.019	0.022	0.917	2.80	0.021	**0.023**	0.920	3.67	0.025	0.026	0.916	4.98
TraDE	0.016	**0.019**	0.908	**1.96**	0.019	0.022	0.918	**3.85**	0.022	**0.023**	0.921	**5.34**	0.025	0.026	0.919	**8.05**
VGGT-Ω CVPR'26	**0.020**	**0.021**	**0.890**	1.00	**0.024**	**0.024**	**0.897**	1.00	**0.028**	**0.026**	**0.899**	1.00	**0.032**	**0.029**	**0.888**	1.00
FastVGGT ICLR'26	**0.020**	**0.021**	0.888	1.36	**0.024**	**0.024**	0.893	1.99	**0.028**	**0.026**	0.895	2.34	**0.032**	**0.029**	0.884	3.07
SparseVGGT CVPR'26	**0.020**	**0.021**	0.886	1.23	**0.024**	**0.024**	0.892	1.56	0.031	0.029	0.895	1.67	0.033	**0.029**	0.884	1.83
HTTM CVPR'26	0.021	**0.021**	0.881	1.30	0.025	0.025	0.881	2.30	0.029	0.027	0.874	3.02	0.037	0.031	0.857	4.15
TraDE	**0.020**	**0.021**	0.888	**1.41**	**0.024**	**0.024**	0.888	**2.45**	**0.028**	0.027	0.886	**3.22**	0.033	0.030	0.880	**4.47**



(a) Sequence distortion **(b)** Cluster-size distribution **(c)** Layer-wise retention

Figure 4: Adaptive behavior of TraDE on the 18 test sequences of 7Scenes. (a) Sequence-wise distortion–retention trade-offs, (b) cluster-size distributions, and (c) layer-wise token retention ratios. Dashed lines indicate the fixed FastVGGT setting ($r = 0.9$).

variation in the amount of removable redundancy across inputs. Figure 4(b) shows that the resulting cluster sizes are highly non-uniform: most clusters remain relatively small, while a smaller portion grows to much larger sizes, and their distributions vary across sequences. Figure 4(c) further reveals pronounced depth-wise variation in token retention. TraDE retains relatively few tokens in the early layers, substantially more tokens in the intermediate stage, and again reduces the retention ratio in later layers, with noticeable differences across input sequences.

These observations directly reflect how TraDE adapts token reduction to the underlying representation structure. At the sequence level, inputs with higher redundancy can tolerate more merging before the distortion criterion is violated, whereas less redundant inputs naturally retain more tokens. At the cluster level, the non-uniform cluster-size distributions show that redundancy is not evenly distributed over the input: highly redundant tokens can be represented by larger clusters, while distinctive tokens remain in smaller clusters. At the depth level, the substantial variation in retention ratios indicates that token redundancy evolves as representations are progressively updated by the

Table 3: Camera pose estimation and point-cloud reconstruction results on 7Scenes and NRGBD.

Dataset	Method	Camera Pose Estimation			Point-cloud Reconstruction				Efficiency
		AUC@3↑	AUC@15↑	AUC@30↑	Acc↓	Comp↓	F1↑	NC↑	Spd.↑
7Scenes	VGGT CVPR'25	**15.35**	**63.64**	**79.15**	0.030	0.028	0.889	**0.821**	1.00
	FastVGGT ICLR'26	14.98	63.86	79.12	**0.028**	**0.026**	**0.900**	0.818	1.87
	SparseVGGT CVPR'26	10.70	59.29	76.43	0.031	0.028	0.881	0.784	2.10
	Co-Me CVPR'26	10.76	57.02	73.90	0.032	0.035	0.851	0.758	1.81
	HTTM CVPR'26	12.92	60.03	76.37	0.032	0.027	0.877	0.787	2.92
	TraDE	15.28	63.81	79.03	0.029	0.027	0.895	0.817	**3.95**
NRGBD	VGGT CVPR'25	68.04	90.27	94.80	0.0249	**0.0247**	0.9271	0.8850	1.00
	FastVGGT ICLR'26	68.36	90.84	95.18	0.024	0.025	0.932	**0.886**	2.36
	SparseVGGT CVPR'26	38.04	81.55	90.20	0.030	0.030	0.878	0.852	2.28
	Co-Me CVPR'26	38.43	79.00	88.37	0.041	0.037	0.816	0.866	1.82
	HTTM CVPR'26	55.60	78.03	82.98	0.034	0.075	0.853	0.853	2.93
	TraDE	**70.72**	**91.66**	**95.63**	**0.0238**	0.0255	**0.9366**	0.8820	**4.15**

Table 4: Ablation study of the key components in TraDE on 7Scenes with VGGT backbone. AUC@3, AUC@15, and AUC@30 denote camera pose estimation accuracy. Acc, Comp, and NC denote point-cloud reconstruction accuracy, completeness, and normal consistency, respectively. Ret. denotes the token retention rate, and FPS measures inference throughput.

Method	ΔE	Ada.	Repart.	Camera Pose			Point Cloud			Efficiency	
				AUC@3↑	AUC@15↑	AUC@30↑	Acc↓	Comp↓	NC↑	Ret.↓	FPS↑
Dense	–	–	–	**15.35**	63.64	**79.15**	0.030	0.028	**0.821**	100.00%	3.91
Cosine-Fixed	–	–	–	13.88	62.67	78.36	**0.029**	0.027	0.810	26.69%	18.40
ΔE-Fixed	✓	–	–	14.28	63.21	78.70	**0.029**	0.027	0.810	26.69%	18.41
ΔE-Adaptive	✓	✓	–	12.39	61.50	77.66	**0.029**	**0.026**	0.784	**14.20%**	**21.96**
TraDE-full	✓	✓	✓	15.28	**63.81**	79.03	**0.029**	0.027	0.817	26.69%	15.46

network, making a single fixed partition or compression ratio insufficient for all layers. In contrast to predefined reduction schemes, TraDE allows both the retained-token ratio and the resulting cluster structure to emerge from the current token representations. This adaptive behavior enables stronger reduction where redundancy is abundant while preserving more tokens where aggressive merging would introduce excessive cluster distortion.

4.5 Ablation Studies

Table 4 evaluates TraDE components. Cosine-Fixed uses representative-token cosine similarity with fixed retention. Under the same 26.69% token retention, replacing pairwise cosine similarity with ΔE improves AUC@3/15/30 from 13.88/62.67/78.36 to 14.28/63.21/78.70, showing the benefit of cluster-level distortion. Building on ΔE, the adaptive criterion reduces the retention ratio to 14.20%, achieving the highest FPS of 21.96 but degrading accuracy due to aggressive reduction. This indicates that a single partition from shallow representations is insufficient as representations evolve with depth. Stage-wise repartitioning instead recomputes the partition from updated representations and retains more tokens in later stages, substantially recovering geometric performance. The full TraDE operates at 15.46 FPS, demonstrating the complementary effects of adaptive reduction and stage-wise repartitioning for balancing geometric accuracy and inference efficiency.

5 Conclusion

We present TraDE, a training-free framework that adapts token reduction to heterogeneous token representations. TraDE balances cluster distortion against the number of retained tokens, adaptively determining both which tokens to retain and how many are needed. Stage-wise repartitioning further accommodates representation changes across network depth with limited overhead. Experiments across multiple backbones, datasets, and geometric tasks show that TraDE substantially improves inference efficiency while preserving competitive geometric performance, demonstrating the effectiveness of representation-adaptive token reduction for long-sequence inference.

AI USE STATEMENT

In this work, we used generative AI tools solely to assist with language editing and polishing, including improving grammar, clarity, readability, and presentation of the manuscript. We did not use generative AI tools to develop research ideas or hypotheses, design the methodology or experiments, or generate data. All AI-assisted text was carefully reviewed and revised by the authors. The technical content, experimental results, analyses, and conclusions of this work were produced and verified by the authors. We take full responsibility for the final content of this work, including any text produced with the aid of generative AI.

REFERENCES

Arindam Banerjee, Srujana Merugu, Inderjit S Dhillon, and Joydeep Ghosh. Clustering with bregman divergences. *Journal of machine learning research*, 6(Oct):1705–1749, 2005.

Yohann Cabon, Lucas Stoffl, Leonid Antsfeld, Gabriela Csurka, Boris Chidlovskii, Jerome Revaud, and Vincent Leroy. Must3r: Multi-view network for stereo 3d reconstruction. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1050–1060. IEEE, 2025.

Yutian Chen, Yuheng Qiu, Ruogu Li, Jay Patrikar, and Sebastian Scherer. Co-me: Confidence guided token merging for visual geometric transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14590–14599, 2026.

Jianing Deng, Yuanzhe Li, Jialu Wang, Song Wang, Tianlong Chen, Huanrui Yang, and Jingtong Hu. Saf3r: Dynamic sparse attention for feed-forward 3d reconstruction transformers. *arXiv preprint arXiv:2607.03612*, 2026.

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. *arXiv preprint arXiv:2602.23361*, 2026.

Juntong Fang, Zequn Chen, Weiqi Zhang, Donglin Di, Xuancheng Zhang, Chengmin Yang, and Yu-Shen Liu. More: Motion-aware feed-forward 4d reconstruction transformer. *arXiv preprint arXiv:2603.05078*, 2026.

Mohsen Fayyaz, Soroush Abbasi Koohpayegani, Farnoush Rezaei Jafari, Sunando Sengupta, Hamid Reza Vaezi Joze, Eric Sommerlade, Hamed Pirsiavash, and Jürgen Gall. Adaptive token sampling for efficient vision transformers. In *European conference on computer vision*, pp. 396–414. Springer, 2022.

Richard Hartley, Andrew Zisserman, et al. *Multiple view geometry in computer vision*, volume 2. Cambridge university press Cambridge, 2003.

Haian Jin, Rundi Wu, Tianyuan Zhang, Ruiqi Gao, Jonathan T Barron, Noah Snavely, and Aleksander Hołyński. Zipmap: Linear-time stateful 3d reconstruction via test-time training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21748–21759, 2026.

Jay Karhade, Nikhil Keetha, Yuchen Zhang, Tanisha Gupta, Akash Sharma, Sebastian Scherer, and Deva Ramanan. Any4d: Unified feed-forward metric 4d reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14578–14589, 2026.

Nikhil Keetha, Norman Müller, Johannes Schönberger, Lorenzo Porzi, Yuchen Zhang, Tobias Fischer, Arno Knapitsch, Duncan Zauss, Ethan Weber, Nelson Antunes, et al. Mapanything: Universal feed-forward metric 3d reconstruction. In *2026 International Conference on 3D Vision (3DV)*, pp. 499–509. IEEE, 2026.

Semi Lee, Hyejin Go, and Hyesong Choi. Adamerge: Salience-aware adaptive token merging for training-free acceleration of vision transformers. *arXiv preprint arXiv:2605.27465*, 2026.

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

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

Yoseph Linde, Andres Buzo, and Robert Gray. An algorithm for vector quantizer design. *IEEE Transactions on communications*, 28(1):84–95, 1980.

Dmitrii Marin, Jen-Hao Rick Chang, Anurag Ranjan, Anish Prabhu, Mohammad Rastegari, and Oncel Tuzel. Token pooling in vision transformers for image classification. In *2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 12–21. IEEE, 2023.

Raul Mur-Artal, Jose Maria Martinez Montiel, and Juan D Tardos. Orb-slam: A versatile and accurate monocular slam system. *IEEE transactions on robotics*, 31(5):1147–1163, 2015.

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

You Shen, Zhipeng Zhang, Yansong Qu, Xiawu Zheng, Jiayi Ji, Shengchuan Zhang, and Liujuan Cao. FastVGGT: Fast visual geometry transformer. In *The Fourteenth International Conference on Learning Representations*, 2026.

Zhijian Shu, Cheng Lin, Tao Xie, Wei Yin, Ben Li, Zhiyuan Pu, Weize Li, Yao Yao, Xun Cao, Xiaoyang Guo, et al. Litevggt: Boosting vanilla vggt via geometry-aware cached token merging. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 36422–36432, 2026.

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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 251–260, 2026.

Zecheng Tang, Jiaye Fu, Qiankun Gao, Haijie Li, Yanmin Wu, Jiaqi Zhang, Siwei Ma, and Jian Zhang. Spark3r: Asymmetric token reduction makes fast feed-forward 3d reconstruction. *arXiv preprint arXiv:2605.06270*, 2026.

Bill Triggs, Philip F McLauchlan, Richard I Hartley, and Andrew W Fitzgibbon. Bundle adjustment—a modern synthesis. In *International workshop on vision algorithms*, pp. 298–372. Springer, 1999.

Chung-Shien Brian Wang, Christian Schmidt, Jens Piekenbrinck, and Bastian Leibe. Block-sparse global attention for efficient multi-view geometry transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14546–14555, 2026a.

Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. VGGT: Visual geometry grounded transformer. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5294–5306. IEEE, 2025.

Jianyuan Wang, Minghao Chen, Shangzhan Zhang, Nikita Karaev, Johannes Schönberger, Patrick Labatut, Piotr Bojanowski, David Novotny, Andrea Vedaldi, and Christian Rupprecht. VGGT-Ω. *arXiv preprint arXiv:2605.15195*, 2026b.

Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20697–20709. IEEE, 2024.

Weitian Wang, Lukas Meiner, Rai Shubham, Cecilia De La Parra, and Akash Kumar. Httm: Head-wise temporal token merging for faster vggt. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 26379–26388, 2026c.

Yifan Wang, Jianjun Zhou, Haoyi Zhu, Wenzheng Chang, Yang Zhou, Zizun Li, Junyi Chen, Jiangmiao Pang, Chunhua Shen, and Tong He. π^3: Permutation-equivariant visual geometry learning. In *International Conference on Learning Representations*, volume 2026, pp. 10481–10497, 2026d.

Tao Xie, Peishan Yang, Yudong Jin, Yingfeng Cai, Wei Yin, Weiqiang Ren, Qian Zhang, Wei Hua, Sida Peng, Xiaoyang Guo, et al. Scal3r: Scalable test-time training for large-scale 3d reconstruction. *arXiv preprint arXiv:2604.08542*, 2026.

Jianing Yang, Alexander Sax, Kevin J Liang, Mikael Henaff, Hao Tang, Ang Cao, Joyce Chai, Franziska Meier, and Matt Feiszli. Fast3r: Towards 3d reconstruction of 1000+ images in one forward pass. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21924–21935. IEEE, 2025.

Jinhao You, Shuo Lyu, Zhuohang Lyu, Tanxuan Li, Zibo Zhao, Jiaxiang Hu, Kai Tang, and Yichen Guo. Regimevggt: Layer-wise spatially preserving redundancy removal for visual geometry grounded transformer. *arXiv preprint arXiv:2606.18439*, 2026.

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.

Kaichen Zhou, Yuhan Wang, Grace Chen, Gaspard Beaudouin, Fangneng Zhan, Paul Liang, and Mengyu Wang. Page-4d: Disentangled pose and geometry estimation for vggt-4d perception. In *International Conference on Learning Representations*, volume 2026, pp. 36401–36414, 2026.

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

A Implementation Details

Backbones and integration. TraDE is training-free and requires no additional optimization of the pretrained backbone. We evaluate it on three representative visual geometry foundation models, including VGGT Wang et al. (2025), π^3 Wang et al. (2026d), and VGGT-Ω Wang et al. (2026b). TraDE is inserted before global self-attention to reduce the number of patch tokens participating in cross-frame interaction, while the original backbone architecture and prediction heads remain unchanged. For VGGT, we adopt the VRAM-efficient implementation provided by FastVGGT Shen et al. (2026) to support inference on long input sequences. At the beginning of each clustering stage, the reducible patch tokens are initialized as individual clusters and iteratively merged according to the procedure described in Sec. 3. Only the representative token of each resulting cluster participates in global self-attention. After global attention, the output of each representative token is propagated to the remaining members of its cluster to restore the original dense token layout. Therefore, subsequent frame-wise attention and prediction modules operate at their original resolution without further modification.

Evaluation metrics. For camera pose estimation, we report AUC@3, AUC@15, and AUC@30, corresponding to the area under the pose-error accuracy curve at thresholds of 3°, 15°, and 30°, respectively. For point-cloud reconstruction, we report Accuracy (Acc), Completeness (Comp), F1@5, and Normal Consistency (NC). Acc and Comp measure the mean nearest-neighbor distances from the reconstructed point cloud to the ground truth and vice versa, respectively, while F1@5 denotes the F1 score computed with a 5 cm distance threshold. NC measures the consistency between the normals of corresponding reconstructed and ground-truth points. Lower Acc and Comp indicate better reconstruction, whereas higher AUC, F1, and NC are better.

Runtime and comparison protocol. All models and inputs use BF16 precision, and forward inference is performed under BF16 autocast. Runtime is measured as the wall-clock latency of the complete model forward pass after the input has already been transferred to the GPU. To ensure accurate CUDA timing, we call `torch.cuda.synchronize()` immediately before starting the timer and again after the forward pass before recording the elapsed time. For the VGGT efficiency comparison, Dense VGGT, FastVGGT, and TraDE are constructed from the same VGGT implementation and the same pretrained checkpoint. Under the common evaluation protocol, all three variants retain only the camera and depth prediction heads, and differ only in their global-attention and token-merging paths. This ensures that the measured runtime differences originate from the corresponding token-reduction and global-attention mechanisms rather than differences in backbone implementation or prediction heads. Since runtime measurements depend strongly on the underlying hardware, we re-evaluate all compared methods on our GPU under the same environment instead of directly using runtime numbers reported in the original papers. The HTTM results are obtained from our reproduction of the method, while the remaining methods are evaluated using their corresponding implementations under the same protocol.

Token-similarity measurement for Figure 2. We measure pair-wise token similarity at the outputs of global-attention blocks 0, 11, and 23 of VGGT, denoted by L0, L11, and L23, respectively. For a patch token t in frame f at layer l, its candidate set $\mathcal{N}_l(f,t)$ contains the other tokens in a 10×10 spatial window centered at t in the current frame, together with the tokens in 10×10 windows around the corresponding spatial location in the preceding and following ten frames. Let $\mathbf{z}_{l,f,t}$ be the representation of token t. Its local pair-wise similarity is the mean cosine similarity to the candidate tokens:

$$s_{l,f,t} = \frac{1}{|\mathcal{N}_l(f,t)|} \sum_{(f',j)\in\mathcal{N}_l(f,t)} \frac{\mathbf{z}_{l,f,t}^\top \mathbf{z}_{l,f',j}}{\|\mathbf{z}_{l,f,t}\|_2 \|\mathbf{z}_{l,f',j}\|_2}. \tag{8}$$

For sequence s, we first average these values over its $N_{s,f}$ patch tokens in each frame and then over all $T = 300$ frames:

$$\mu_{s,l,f} = \frac{1}{N_{s,f}} \sum_{t=1}^{N_{s,f}} s_{l,f,t}, \qquad S_{s,l} = \frac{1}{T} \sum_{f=1}^{T} \mu_{s,l,f}. \tag{9}$$

The bar height is $S_{s,l}$. The black error bar reports the standard deviation of the token-level similarities $s_{l,f,t}$ over all patch tokens in the T-frame sequence, and therefore indicates variation across

individual tokens. The dark-gray dashed line is not a fitted curve: at each of the three layers, its ordinate is the arithmetic mean across the four sequences,

$$\bar{S}_l = \frac{1}{4}\sum_{s=1}^{4} S_{s,l}, \tag{10}$$

and the three points $(\mathrm{L0}, \bar{S}_0)$, $(\mathrm{L11}, \bar{S}_{11})$, and $(\mathrm{L23}, \bar{S}_{23})$ are connected in order.

Hyperparameters. We use the same hyperparameters across all datasets and backbones without dataset-specific tuning. The trade-off coefficient in the cluster-merging objective is fixed to $\lambda = 0.04$. Candidate clusters are constructed within a local spatio-temporal neighborhood with temporal range $t = 4$ and spatial radius $\rho = 2$. Thus, candidate relations are initially restricted to nearby tokens in space and time, while the inherited candidate relations of merged clusters allow the effective search range to expand during iterative clustering. These settings are kept fixed throughout all main experiments.

Stage-wise repartitioning. Recomputing the token partition at every global-attention layer would introduce unnecessary clustering overhead. We therefore perform the complete iterative clustering procedure only at predefined stage boundaries and reuse the resulting partition within each stage. To further illustrate the depth-wise evolution that motivates this design, we visualize a portion of the global-attention matrix on a 7Scenes sequence in Fig. 7. Specifically, we use a 10-frame input sampled with a stride of 10 and visualize the attention between the first frame as queries and the second frame as keys across global-attention layers. The visualization shows how cross-frame attention patterns evolve with network depth, providing qualitative motivation for updating the token partition at selected stages rather than at every layer. Specifically, the repartitioning layers are $\{0, 9, 21\}$ for VGGT, $\{0, 9\}$ for π^3, and $\{0, 10, 17\}$ for VGGT-Ω. For VGGT and VGGT-Ω, the stage boundaries are motivated by three distinct attention patterns. In the early stage, attention from different queries tends to concentrate on a similar subset of keys. In the intermediate stage, the attention patterns of different queries become more differentiated and are increasingly concentrated on their corresponding keys. In the later stage, the attention becomes more dispersed again. For π^3, which contains fewer layers, only the first two stages are observed. At each stage boundary, the token partition is recomputed from the updated token representations, allowing the reduction pattern to adapt to the depth-wise evolution of token similarity. The resulting partition is then reused by the remaining global-attention layers in that stage.

Token retention ratio. For a given token partition, we define the retention ratio as the fraction of tokens actually participating in global attention:

$$\eta_s = \frac{N_{\text{active}}^{(s)}}{FP}, \tag{11}$$

where s denotes the stage index, F is the number of input frames, and $P = 5 + HW$ for VGGT, including one camera token, four register tokens, and $H \times W$ patch tokens per frame. Since each partition is reused by multiple global-attention layers within its stage, the reported retention ratio is computed as a layer-weighted average:

$$\bar{\eta} = \frac{\sum_s L_s \eta_s}{\sum_s L_s}, \tag{12}$$

where L_s denotes the number of global-attention layers using the partition obtained at stage s. This weighting reflects the actual proportion of active tokens across all global-attention layers. Dense VGGT has $\bar{\eta} = 1$. For FastVGGT, N_{active} includes unmerged, destination, and protected tokens, and thus its retention ratio is not necessarily equal to $1 - \texttt{merge_ratio}$.

B Parameter Analysis

B.1 Effect of the Coefficient λ

The trade-off coefficient λ controls the balance between cluster distortion and token reduction in the merging objective. Importantly, λ does not specify a target token retention ratio. Instead, it

Table 5: Complete parameter analysis of the trade-off coefficient λ on 7Scenes. The temporal neighborhood size and spatial neighborhood radius are fixed to $t = 4$ and $\rho = 2$. Ret. denotes the token retention ratio across the three global-attention stages, and Spd. denotes the forward-pass acceleration relative to DenseVGGT. The default setting is highlighted in gray.

λ	Camera Pose Estimation		Point-cloud Reconstruction			Efficiency	
	AUC@3↑	AUC@30↑	Acc. (m)↓	Comp. (m)↓	NC↑	Ret.↓	Spd.↑
DENSE	15.35	79.15	0.0304	0.0275	0.8212	100.00%	1.00×
0.01	15.39	79.15	0.0307	0.0277	0.8205	70.21%	1.58×
0.02	15.48	79.15	0.0302	0.0277	0.8214	47.49%	2.42×
0.03	15.41	79.08	0.0297	0.0274	0.8203	34.64%	3.25×
0.04	15.28	79.03	0.0290	0.0270	0.8170	26.69%	3.95×
0.05	14.94	78.88	0.0290	0.0269	0.8121	21.81%	4.44×
0.06	14.48	78.70	0.0288	0.0267	0.8058	18.42%	4.88×
0.07	14.13	78.46	0.0288	0.0264	0.7993	16.00%	5.01×
0.08	13.77	78.29	0.0288	0.0264	0.7929	14.21%	5.30×
0.09	13.43	78.10	0.0288	0.0260	0.7868	12.83%	5.37×
0.10	12.97	77.89	0.0291	0.0261	0.7808	11.73%	5.59×

Table 6: Parameter analysis of the spatio-temporal neighborhood. When varying the temporal neighborhood size t, we fix $\rho = 2$; when varying the spatial neighborhood radius ρ, we fix $t = 4$. The trade-off coefficient is fixed to $\lambda = 0.04$. Grouping denotes the synchronized GPU runtime for token grouping and repartitioning, reported as a percentage of model forward time. The default settings are highlighted in gray.

Parameter	Value	Camera Pose Estimation		Point-cloud Reconstruction			Efficiency		
		AUC@3↑	AUC@30↑	Acc. (m)↓	Comp. (m)↓	NC↑	Ret.↓	Grouping (%)↓	Spd.↑
t	1	15.39	79.02	0.0295	0.0273	0.8177	30.14%	8.73%	3.88×
	2	15.32	78.99	0.0293	0.0271	0.8173	27.91%	9.77%	4.05×
	4	15.28	79.03	0.0290	0.0270	0.8170	26.69%	13.90%	3.95×
	6	15.18	78.96	0.0293	0.0271	0.8164	26.54%	16.45%	3.94×
	8	15.12	78.95	0.0292	0.0270	0.8162	26.43%	19.63%	3.73×
ρ	1	15.36	79.03	0.0295	0.0273	0.8180	29.31%	8.08%	3.98×
	2	15.28	79.03	0.0290	0.0270	0.8170	26.69%	13.90%	3.95×
	3	15.25	78.99	0.0292	0.0271	0.8156	26.28%	19.78%	3.82×
	4	15.15	78.94	0.0293	0.0270	0.8145	26.11%	25.15%	3.56×

determines how much additional cluster distortion can be tolerated for reducing one retained token according to the merging criterion in Equation 6. Therefore, the resulting retention ratio is still determined adaptively by the underlying token representations of each input.

As shown in Table 5, increasing λ permits merges with larger distortion changes and consequently leads to more aggressive token reduction. As λ increases from 0.01 to 0.10, the average token retention ratio decreases from 70.21% to 11.73%, while the inference speedup increases from $1.58\times$ to $5.59\times$. Meanwhile, geometric performance remains stable under moderate values of λ, but gradually degrades when overly aggressive merging is allowed, particularly in AUC@3 and NC. These results demonstrate that λ provides a controllable accuracy–efficiency trade-off without prescribing the number or proportion of tokens to retain. We therefore set $\lambda = 0.04$ throughout the main experiments to achieve a balanced operating point between representation fidelity and inference efficiency.

B.2 Effect of the Spatio-temporal Neighborhood

We analyze the temporal range t and spatial radius ρ used for candidate cluster search. These parameters only determine the initial search neighborhood for constructing candidate relations, rather than specifying the token retention ratio; the actual retained tokens are still determined by the distortion-based merging objective.

Table 7: Analysis of token repartitioning strategies on the VGGT backbone. Single partitioning computes the partition only at layer 0, uniform repartitioning uses fixed intervals at layers $\{0, 8, 16\}$, and the proposed stage-wise strategy recomputes the partition at layers $\{0, 9, 21\}$. "All" denotes repartitioning at every global-attention layer. Ret. denotes the token retention rate, and Spd. is measured relative to Dense VGGT. All reduced variants use $\lambda = 0.04$, $t = 4$, and $\rho = 2$. The default stage-wise setting is highlighted in gray.

Method	Camera Pose Estimation			Point-cloud Reconstruction				Efficiency		
	AUC@3↑	AUC@15↑	AUC@30↑	Acc.↓	Comp.↓	F1↑	NC↑	Ret.↓	FPS↑	Spd.↑
Dense VGGT	15.35	63.64	79.15	0.0304	0.0275	0.8893	0.8212	100.00%	3.911	1.00×
Single (0)	12.39	61.50	77.66	0.0293	0.0263	0.8938	0.7836	14.20%	21.962	5.62×
Uniform ($\{0, 8, 16\}$)	14.42	63.06	78.62	0.0294	0.0274	0.8938	0.8133	27.03%	15.442	3.95×
Stage-wise ($\{0, 9, 21\}$)	15.28	63.81	79.03	0.0290	0.0270	0.8948	0.8170	26.69%	15.460	3.95×
All	12.64	61.06	77.36	0.0307	0.0270	0.8893	0.7826	24.67%	8.412	2.15×

As shown in Table 6, enlarging either the temporal or spatial neighborhood generally enables more candidate clusters to be explored, leading to moderately more aggressive token reduction. When t increases from 1 to 8, the retention ratio decreases from 30.14% to 26.43%, while increasing ρ from 1 to 4 reduces it from 29.31% to 26.11%. Meanwhile, geometric performance remains relatively stable across different neighborhood sizes. For example, AUC@30 varies within only 78.94–79.03, and the reconstruction metrics exhibit similarly small variations.

However, a larger search neighborhood also introduces substantially higher grouping overhead. For the temporal range, the grouping cost increases from 8.73% at $t = 1$ to 19.63% at $t = 8$, while the retention ratio improves only marginally beyond $t = 4$. A similar trend is more pronounced for the spatial radius: increasing ρ from 2 to 4 reduces the retention ratio by only 0.58 percentage points, but increases the grouping overhead from 13.90% to 25.15%, causing the overall speedup to decrease from $3.95\times$ to $3.56\times$. These results indicate diminishing returns from excessively enlarging the candidate search neighborhood. We therefore adopt $t = 4$ and $\rho = 2$ as the default setting, which provides sufficient spatio-temporal coverage while avoiding excessive grouping overhead.

B.3 Effect of Stage-wise Repartitioning

We further investigate how the frequency and placement of token repartitioning affect the performance of TraDE. As shown in Table 7, using a single partition computed at the first global-attention layer achieves the highest FPS of 21.962 and the largest acceleration of $5.62\times$, but substantially degrades camera pose estimation and normal consistency. This indicates that a partition derived from shallow-layer representations cannot adequately accommodate the evolution of token similarity throughout the network. Repartitioning the tokens at intermediate layers substantially recovers the geometric performance. Compared with uniform repartitioning at layers $\{0, 8, 16\}$, the proposed stage-wise setting at $\{0, 9, 21\}$ improves AUC@3 from 14.42 to 15.28 and AUC@30 from 78.62 to 79.03, while also increasing NC from 0.8133 to 0.8170. Notably, these improvements are achieved with a slightly lower token retention ratio (26.69% vs. 27.03%) and the same $3.95\times$ acceleration. This suggests that aligning repartitioning with the stage-wise evolution of token representations is more effective than updating the partition at uniformly spaced layers.

Recomputing the partition at every global-attention layer might appear to provide the strongest adaptation to evolving token representations, since TraDE can independently determine the retention ratio and token assignment at each layer without changing its hyperparameters. However, the results show the opposite trend. The All strategy reduces AUC@3/AUC@30/NC to $12.64/77.36/0.7826$ and achieves only $2.15\times$ acceleration, substantially worse than stage-wise repartitioning. Repartitioning at every layer also repeatedly performs candidate search, cluster pairing, and partition updates, introducing additional computation that reduces the overall inference efficiency. This suggests that more frequent repartitioning is not necessarily beneficial. First, TraDE determines token partitions by optimizing cluster distortion as a proxy for preserving the current-layer representations, rather than directly optimizing the final geometric objective. Consequently, a partition that is locally optimal at every layer does not necessarily correspond to the optimal final task performance, and the results indicate that a gap remains between these two objectives. Second, not every representation

change across consecutive global-attention layers necessarily warrants repartitioning. Within the same stage, representations typically evolve in a relatively consistent manner, and some inter-layer variations may reflect local perturbations or transient representation fluctuations rather than meaningful changes in redundancy structure. Repartitioning in response to such variations can therefore introduce unnecessary adjustments. In contrast, updating the partition only when representations undergo more substantial stage-level changes avoids overreacting to short-term fluctuations while better adapting to meaningful depth-wise representation dynamics.

C Algorithmic Procedure

Algorithm 1 summarizes the inference procedure of TraDE. The token partition is computed from the representations at the first global-attention layer of each stage and reused by the remaining global-attention layers in that stage. At the next stage boundary, the partition is recomputed from the updated representations.

Algorithm 1 Inference procedure of TraDE

Require: Token features X; temporal range t; spatial radius ρ; coefficient λ
Ensure: Token features with the original dense layout

for each stage s **do**
  if at the first global-attention layer of stage s **then**
    Preserve task-specific tokens and let $\mathcal{R}$ be the reducible patch tokens
    Normalize $z_i \leftarrow x_i / \|x_i\|_2,\ \forall i \in \mathcal{R}$
    Initialize $\mathcal{P} \leftarrow \{\{i\} : i \in \mathcal{R}\}$, $u_{\{i\}} \leftarrow z_i$, $w_{\{i\}} \leftarrow 1$, $r_{\{i\}} \leftarrow i$
    repeat
      Step 1: Construct candidate sets $\mathcal{C}_i$ according to Equation 3
      Step 2: Distortion-based Pairing. Compute $r_{G_i \cup G_j}$, $\Delta E(G_i, G_j)$, and p_i according to Equation 4
      Step 3: $\mathcal{M} \leftarrow \{(i, p_i) : p_{p_i} = i,\ \Delta E(G_i, G_{p_i}) < 2\lambda\}$
      if $\mathcal{M} \neq \emptyset$ **then**
        Step 4: Merge all pairs in $\mathcal{M}$ in parallel and update $\mathcal{P}$, u_G, w_G, and r_G
      end if
    until $\mathcal{M} = \emptyset$
  end if
  for each global-attention layer in stage s **do**
    Apply global attention to the protected tokens and $\{x_{r_G} : G \in \mathcal{P}\}$
    Restore the dense layout by $y_i \leftarrow y_{r_G}$ for $\forall i \in G,\ G \in \mathcal{P}$
    Apply the subsequent frame-wise attention and backbone operations
  end for
end for

D Additional Experimental Results

D.1 Evaluation under Full-Sequence Uniform Sampling

The construction of ScanNet inputs varies across existing studies. In our main experiments, we select 50 sequences from the ScanNet test split, requiring each sequence to contain sufficient frames for the maximum 1000-frame setting, and construct inputs using fixed-stride sampling. In contrast, the ScanNet50 setting commonly adopted by FastVGGT Shen et al. (2026) contains sequences with different lengths, including sequences shorter than 1000 frames, and constructs each input by sampling the required number of frames over the full temporal span of the sequence: the first and last frames are retained, while the intermediate frames are uniformly sampled. To examine whether the effectiveness of TraDE depends on the input construction used in our main experiments, we additionally evaluate TraDE following this sampling protocol and report the same pose and reconstruction metrics used by FastVGGT. The results are shown in Table 8.

TraDE maintains comparable geometric accuracy while providing increasingly pronounced efficiency gains as the input sequence becomes longer. For 100-frame inputs, TraDE achieves $1.663\times$

Table 8: Camera pose estimation, point-cloud reconstruction, and inference efficiency on ScanNet50 under full-sequence uniform sampling. For each target input length, frames are uniformly sampled over the full temporal span of each sequence, with the first and last frames retained.

Frames	Method	ATE↓	ARE↓	RPE-rot↓	RPE-trans↓	CD↓	Spd.↑
100	VGGT	0.092	3.948	1.179	0.056	0.441	1.00
	FastVGGT	0.094	3.830	1.243	0.061	0.436	1.72
	TraDE	0.092	3.882	1.189	0.057	0.445	1.66
300	VGGT	0.096	4.187	0.780	0.040	0.450	1.00
	FastVGGT	0.093	3.931	0.746	0.036	0.448	2.66
	TraDE	0.091	3.928	0.731	0.037	0.459	3.52
500	VGGT	0.124	4.818	0.935	0.042	0.465	1.00
	FastVGGT	0.093	4.055	0.677	0.031	0.464	3.06
	TraDE	0.098	4.115	0.653	0.034	0.468	5.38
1000	VGGT	0.128	5.562	1.069	0.039	0.467	1.00
	FastVGGT	0.111	4.177	0.628	0.029	0.469	3.39
	TraDE	0.107	4.253	0.611	0.029	0.473	8.33

acceleration, slightly below the $1.724\times$ of FastVGGT, while maintaining similar pose and reconstruction accuracy. As the sequence length increases, however, the computational advantage of TraDE becomes substantially larger. At 300, 500, and 1000 frames, TraDE reaches $3.520\times$, $5.379\times$, and $8.325\times$ acceleration, respectively, compared with $2.661\times$, $3.057\times$, and $3.386\times$ for FastVGGT. Meanwhile, the geometric errors remain close to those of FastVGGT across different sequence lengths. At 300 frames, TraDE reduces ATE from 0.093 to 0.091 and RPE-rot from 0.746 to 0.731, while maintaining nearly identical ARE. At 1000 frames, TraDE further achieves lower ATE (0.107 vs. 0.111) and RPE-rot (0.611 vs. 0.628), with the same RPE-trans of 0.029. Although small differences are observed in ARE and CD, the overall geometric quality remains comparable while the acceleration ratio increases to $8.325\times$. These results demonstrate that the efficiency advantage of TraDE is preserved under a different sequence construction and frame-sampling protocol, and becomes increasingly evident for long input sequences.

D.2 Adaptation to Input Redundancy.

We further examine whether TraDE can adapt its reduction extent to different levels of input redundancy by varying the temporal sampling stride on 7Scenes while fixing the input length to 100 frames. As shown in Table 9, increasing the sampling stride from 1 to 10 monotonically raises the token retention ratio from 18.08% to 36.01%, while the acceleration ratio correspondingly decreases from $2.57\times$ to $1.68\times$. This trend is consistent with the reduction in temporal overlap: densely sampled inputs contain stronger cross-frame redundancy and therefore permit more aggressive token reduction, whereas more widely spaced frames retain more distinctive information and naturally require more tokens. Geometric performance does not vary monotonically with the sampling stride. Stride 2 achieves the highest pose accuracy, with AUC@30 reaching 82.28, while larger strides generally improve Comp and NC, reaching 0.0236 and 0.827 at stride 10. This suggests that increasing the temporal spacing not only reduces redundancy but also introduces more diverse viewpoints, which can benefit geometric reconstruction. Overall, the monotonic change in token retention and efficiency demonstrates that TraDE adapts its reduction extent to the redundancy structure of the input without relying on a predefined compression ratio.

D.3 Evaluation on TUM-Dynamics

Table 10 further evaluates TraDE on TUM-Dynamics to examine its generalization to dynamic scenes. TraDE achieves the highest inference efficiency, reaching $3.36\times$ acceleration, compared with $2.77\times$ for HTTM and $2.13\times$ for SparseVGGT. Despite the more challenging dynamic content, TraDE maintains substantially better camera-pose accuracy than the other highly accelerated methods. In particular, compared with HTTM, TraDE improves AUC@3/15/30 from $16.70/49.27/64.48$ to $20.61/55.78/70.52$, while providing higher acceleration. The reconstruction metrics show a more pronounced accuracy–efficiency trade-off. TraDE obtains Acc/Comp/F1/NC

Table 9: Adaptation of TraDE to input redundancy on 7Scenes with 100-frame inputs. Increasing the sampling stride reduces temporal overlap and input redundancy. Ret. denotes the token-retention ratio, and Spd. denotes the acceleration ratio relative to dense VGGT.

Stride	Camera Pose Estimation			Point-cloud Reconstruction			Efficiency	
	AUC@3↑	AUC@15↑	AUC@30↑	Acc.↓	Comp.↓	NC↑	Ret.↓	Spd.↑
1	14.64	60.41	77.13	0.0284	0.0260	0.815	18.08%	2.57×
2	20.84	68.18	82.28	0.0311	0.0264	0.807	21.49%	2.10×
4	16.12	64.08	78.79	0.0283	0.0240	0.816	26.98%	1.92×
6	15.99	63.66	78.48	0.0279	0.0230	0.819	30.27%	1.84×
8	15.60	63.85	79.03	0.0284	0.0237	0.823	33.83%	1.80×
10	15.15	64.06	79.39	0.0291	0.0236	0.827	36.01%	1.68×

Table 10: Camera pose estimation and point-cloud reconstruction results on TUM-Dynamics using the VGGT backbone. Frames are sampled with a stride of 3, with at most 300 frames retained for each sequence. AUC@3, AUC@15, and AUC@30 denote pose accuracy. Acc, Comp, F1, and NC denote reconstruction accuracy, completeness, F1 score, and normal consistency, respectively. Spd. is the acceleration ratio relative to dense VGGT.

Dataset	Method	Camera Pose Estimation			Point-cloud Reconstruction				Efficiency
		AUC@3↑	AUC@15↑	AUC@30↑	Acc↓	Comp↓	F1↑	NC↑	Spd.↑
TUM-Dynamics	Dense	**25.67**	**60.14**	**74.08**	**0.053**	0.058	**0.784**	**0.677**	1.00
	FastVGGT	21.55	57.67	72.19	0.055	0.058	0.774	0.668	2.11
	SparseVGGT	14.80	48.83	65.17	0.054	**0.054**	0.778	0.654	2.13
	Co-Me	13.69	42.62	57.64	0.139	0.095	0.584	0.637	1.78
	HTTM	16.70	49.27	64.48	0.061	0.059	0.768	0.655	2.77
	TraDE	20.61	55.78	70.52	0.062	0.062	0.749	0.655	**3.36**

of $0.062/0.062/0.749/0.655$, introducing moderate degradation relative to dense VGGT and FastVGGT. This is expected because dynamic objects reduce the temporal redundancy that can be safely exploited for token reduction and make cross-frame geometric correspondence more sensitive to aggressive compression. Nevertheless, TraDE avoids the severe pose degradation observed in SparseVGGT and Co-Me while achieving the highest overall acceleration. These results indicate that TraDE remains effective under dynamic-scene inputs, although the attainable efficiency gain comes with a larger reconstruction trade-off than in static benchmarks.

D.4 ETH3D Results

To further evaluate the generalization of TraDE beyond the indoor RGB-D benchmarks used in the main experiments, we conduct additional experiments on ETH3D, which contains diverse indoor and outdoor multi-view scenes with high-quality geometric ground truth. As shown in Table 11, TraDE preserves camera pose estimation accuracy while consistently improving point-cloud reconstruction. Compared with dense VGGT, TraDE achieves comparable AUC@3 (26.60 vs. 27.00), while improving AUC@15 from 67.93 to 68.55 and AUC@30 from 80.55 to 81.40. More notably, TraDE achieves the best reconstruction performance across all reported metrics, reducing Acc and Comp from 0.309/0.821 to 0.292/0.803, while improving F1 and NC from 0.627/0.823 to 0.655/0.828. Meanwhile, TraDE provides a $1.20\times$ inference speedup. Although SparseVGGT achieves higher pose accuracy, it is slower than dense VGGT ($0.90\times$), while FastVGGT provides a comparable $1.21\times$ speedup but with noticeably degraded reconstruction quality. HTTM also exhibits substantial degradation in both pose and reconstruction accuracy while running slower than dense inference ($0.75\times$). These results demonstrate that TraDE maintains a favorable accuracy–efficiency trade-off on the diverse ETH3D scenes, further validating its robustness across different input domains.

D.5 Additional 7Scenes and NRGBD Results

The main paper reports results on 7Scenes and NRGBD using the VGGT backbone. For completeness, Table 12 reports additional results with the π^3 and VGGT-Ω backbones. On 7Scenes,

Table 11: Camera pose estimation and point-cloud reconstruction results on ETH3D using the VGGT backbone. AUC@3, AUC@15, and AUC@30 denote pose accuracy. Acc, Comp, F1, and NC denote reconstruction accuracy, completeness, F1 score, and normal consistency, respectively. Spd. is the acceleration ratio relative to dense VGGT.

Dataset	Method	Camera Pose Estimation			Point-cloud Reconstruction				Efficiency
		AUC@3↑	AUC@15↑	AUC@30↑	Acc↓	Comp↓	F1↑	NC↑	Spd.↑
ETH3D	VGGT	27.00	67.93	80.55	0.309	0.821	0.627	0.823	1.00
	FastVGGT	21.84	63.17	77.72	0.359	0.861	0.551	0.806	1.21
	SparseVGGT	**29.38**	**70.34**	**81.92**	0.319	0.827	0.623	0.826	0.90
	Co-Me	6.08	34.28	49.40	1.416	2.072	0.336	0.692	0.66
	HTTM	10.06	46.15	63.56	0.472	1.048	0.419	0.758	0.75
	TraDE	26.60	68.55	81.40	**0.292**	**0.803**	**0.655**	**0.828**	1.20

Table 12: Additional camera pose estimation, point-cloud reconstruction, and inference-efficiency results on 7Scenes and NRGBD using the π^3 and VGGT-Ω backbones, with keyframes sampled at stride 3. Spd. denotes the acceleration ratio relative to the dense backbone, and FPS denotes frames per second. The best and second-best values are in bold and underlined, respectively.

Dataset	Backbone	Method	Camera Pose Estimation			Point-cloud Reconstruction				Efficiency	
			AUC@3↑	AUC@15↑	AUC@30↑	Acc↓	Comp↓	F1↑	NC↑	Spd.↑	FPS↑
7Scenes	π^3	Dense	17.98	67.98	82.05	0.020	0.022	0.930	0.860	1.00	6.10
		FastVGGT	19.17	68.74	82.26	0.020	**0.020**	**0.937**	0.860	2.70	16.46
		SparseVGGT	13.38	64.49	80.15	0.025	0.023	0.903	0.806	2.23	13.57
		HTTM	18.01	68.05	81.89	**0.019**	**0.020**	**0.937**	0.859	3.24	19.76
		TraDE	**19.52**	**69.02**	**82.53**	**0.019**	**0.020**	**0.937**	**0.862**	**4.24**	**25.88**
	VGGT-Ω	Dense	30.12	**77.27**	**87.56**	0.021	**0.020**	0.932	0.831	1.00	8.48
		FastVGGT	29.63	76.71	87.21	0.021	**0.020**	0.933	0.828	2.31	19.55
		SparseVGGT	29.85	76.84	87.31	0.021	**0.020**	0.932	0.826	2.08	17.63
		HTTM	27.22	75.40	86.47	0.021	**0.020**	0.932	0.814	2.69	22.83
		TraDE	**30.81**	77.16	87.45	**0.020**	**0.020**	**0.936**	**0.833**	**2.96**	**25.09**
NRGBD	π^3	Dense	**83.60**	**95.83**	**97.86**	0.025	0.031	0.906	0.912	1.00	4.83
		FastVGGT	71.30	92.43	96.08	**0.024**	**0.029**	**0.922**	0.911	2.82	13.62
		SparseVGGT	59.46	89.57	94.63	0.030	**0.029**	0.885	0.879	2.36	11.39
		HTTM	73.53	92.97	96.37	**0.024**	**0.029**	0.921	0.912	3.82	18.45
		TraDE	67.80	91.59	95.66	0.025	**0.029**	0.920	0.909	**5.96**	**28.79**
	VGGT-Ω	Dense	**86.44**	**96.63**	**98.27**	**0.018**	**0.016**	**0.974**	**0.921**	1.00	6.85
		FastVGGT	85.09	96.16	98.02	**0.018**	**0.016**	**0.974**	0.916	2.43	16.66
		SparseVGGT	81.93	95.51	97.70	0.021	0.018	0.959	0.896	2.17	14.85
		HTTM	80.43	94.99	97.42	0.020	0.017	0.967	0.905	3.22	22.05
		TraDE	84.41	95.88	97.84	0.019	**0.016**	0.971	0.917	**3.95**	**27.10**

TraDE achieves the highest efficiency on both backbones, reaching $4.24\times$ and $2.96\times$ acceleration on π^3 and VGGT-Ω, respectively, while maintaining competitive or best geometric performance. On NRGBD, TraDE reaches $5.96\times$ acceleration with π^3, substantially exceeding HTTM ($3.82\times$) and FastVGGT ($2.82\times$). Although camera-pose accuracy degrades in this setting, reconstruction quality remains competitive. This trade-off is largely governed by the coefficient λ. Under the unified $\lambda = 0.04$ setting, the distortion criterion permits particularly aggressive token reduction for π^3 on NRGBD, leading to higher efficiency but also a larger loss in pose accuracy. This effect may be further amplified by the shallower π^3 architecture, which provides fewer opportunities for stage-wise repartitioning. With VGGT-Ω, TraDE achieves a more balanced trade-off, reaching $3.95\times$ acceleration while largely preserving both pose and reconstruction accuracy.

E QUALITATIVE VISUALIZATION

Figure 5 provides qualitative comparisons of camera trajectories on six ScanNet sequences. Across different trajectory shapes and camera-motion patterns, the trajectories estimated by TraDE generally follow the ground truth closely and remain visually comparable to those produced by dense VGGT. In particular, the global trajectory structures are well preserved without introducing noticeable systematic drift, indicating that the reduced token representations retain the cross-view information required for camera pose estimation.

Figure 6 further compares the reconstructed point clouds on the six sequences. TraDE preserves the overall scene layouts and major geometric structures of dense VGGT, including room boundaries, furniture, and other salient scene components. Despite substantially reducing the number of tokens involved in global attention, no obvious structural degradation is observed in these examples. These qualitative results are consistent with the quantitative evaluation, further showing that TraDE can reduce redundant token computation while preserving the geometric information required for both camera pose estimation and 3D reconstruction.

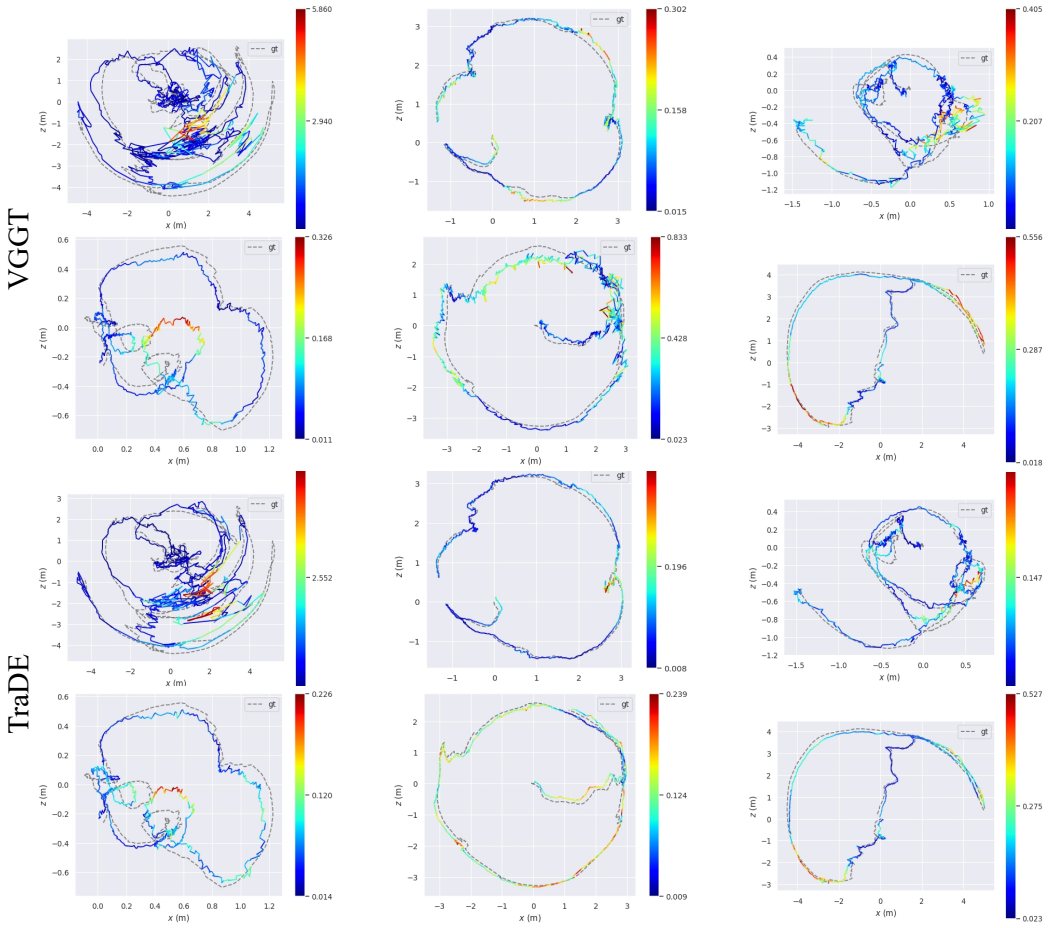
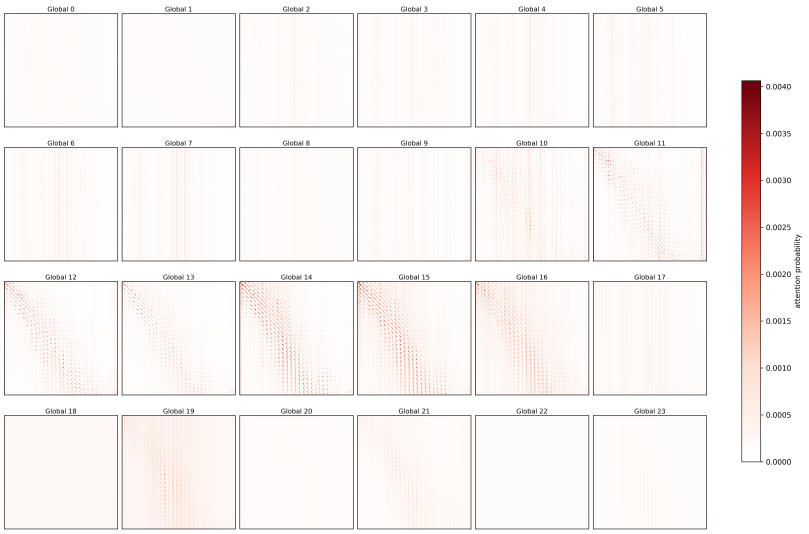


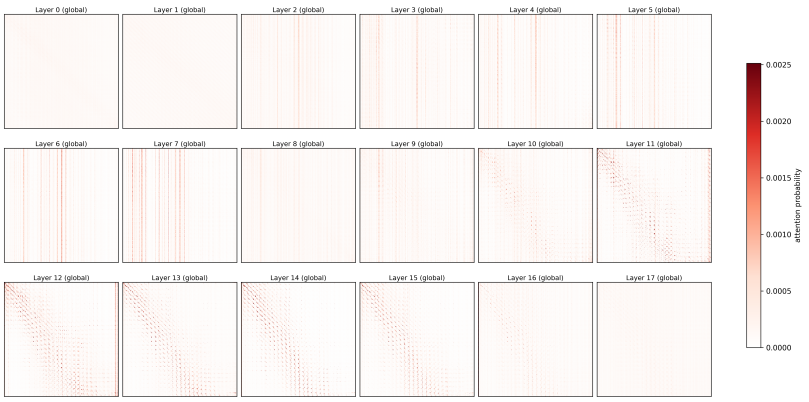
Figure 5: Qualitative camera trajectory comparison between dense VGGT and TraDE on six ScanNet sequences. The dashed gray curves denote the ground-truth trajectories, while the colored curves show the estimated trajectories.



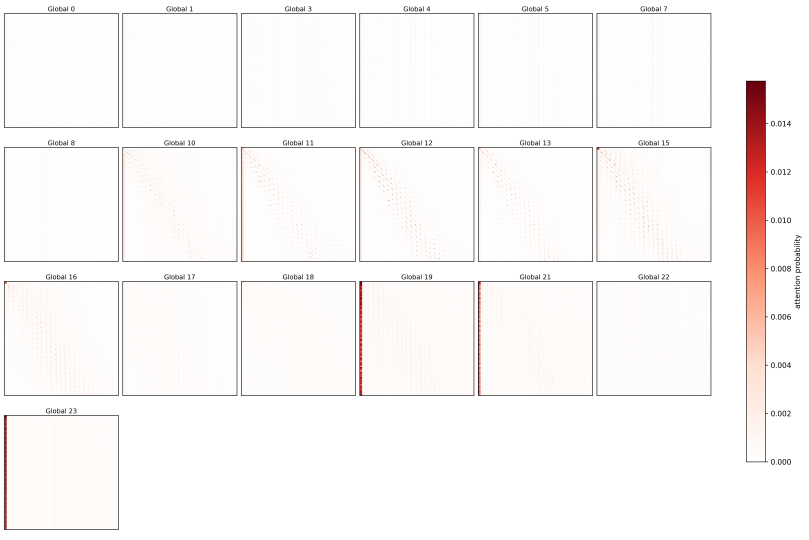
Figure 6: Qualitative point-cloud reconstruction comparison between dense VGGT and TraDE on six ScanNet sequences. TraDE preserves the overall scene geometry and structural details while substantially reducing the computation of global attention.



(a) VGGT



(b) π^3



(c) VGGT-Ω

Figure 7: Visualization of attention probabilities across the global-attention layers of three backbones: (a) VGGT, (b) π^3, and (c) VGGT-Ω. Each panel corresponds to one global-attention layer, and color intensity indicates attention probability.