Diversity-aware View Partitioning for Scalable VGGT

Jinsoo Park[1] Donggyu Choi[2] Ahyun Seo[3] Minsu Cho[1,4] Jeany Son[1⋆]

[1]POSTECH [2]GIST [3]KAIST [4]RLWRLD
https://jspark1213.github.io/DA-VGGT

Abstract. Geometry transformers such as VGGT achieve strong performance by jointly reasoning over multiple views with global attention. However, scaling them to large view collections remains challenging due to the quadratic cost of attention. Moreover, our empirical analysis reveals that the reconstruction quality in VGGT is sensitive to the distribution of viewpoints. Simply increasing the number of views without sufficient viewpoint diversity can even degrade performance, as redundant views introduce highly similar tokens that dilute informative geometric signals in the attention mechanism. Motivated by this observation, we propose a training-free and plug-and-play VGGT inference framework that organizes views into diversity-aware balanced chunks. The chunks are constructed through combinatorial graph partitioning over visual dissimilarity and spatial dispersion. This view organization allows the transformer to focus attention on geometrically informative views while reducing redundant attention interactions. To estimate spatial dispersion without full pose estimation, we approximate spatial relationships via a soft pose propagation strategy based on visual similarity from a small set of seed frames. Extensive experiments demonstrate improved performance in camera pose estimation, multi-view depth prediction, and 3D reconstruction while reducing memory usage and inference latency. Our framework also complements existing VGGT variants, enabling scalable multi-view reconstruction without sacrificing geometric fidelity.

Keywords: Multi-view Geometry · Geometry Transformers · Scalable Multi-view Reconstruction

1 Introduction

Multi-view geometry transformers such as VGGT [35] have recently emerged as a powerful paradigm for 3D scene reconstruction. By jointly reasoning over multiple images through global attention, these models infer camera poses, multi-view depths, and pointmaps within a unified transformer architecture. Despite this capability, VGGT remains fundamentally limited in scalability. The core challenge arises from global attention: as the number of input frames grows, the token set grows proportionally, causing quadratic increases in computation and memory.

⋆ Corresponding author.

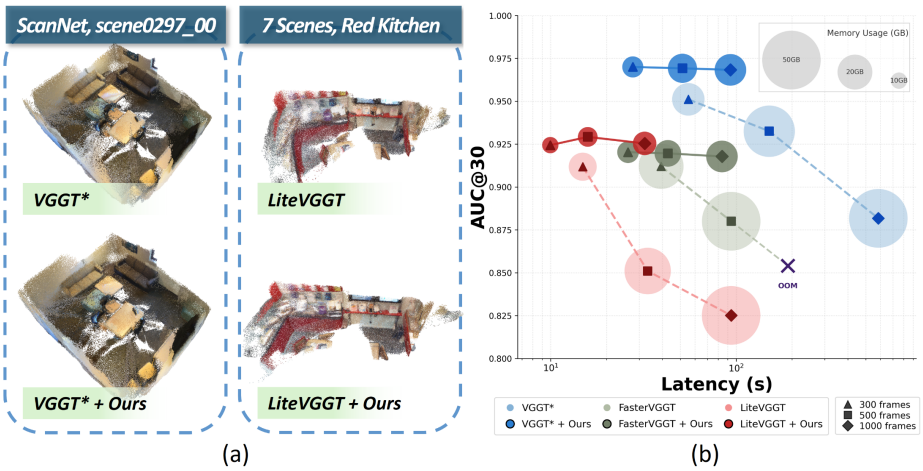

Fig. 1: Effectiveness of diversity-aware view partitioning, (a) Our method reconstructs more geometrically consistent structures while preserving finer scene details compared to existing efficient VGGT variants, such as LiteVGGT [28]. (b) Unlike other methods [28, 35], whose performance tends to degrade as the number of input frames increases, our simple diversity-aware view partitioning mitigates this effect and achieves a better efficiency–accuracy trade-off.

This quickly becomes prohibitive for long sequences, limiting the deployment of VGGT on thousands of images. Recent work addresses this through various efficiency-oriented strategies [7, 12, 14, 28, 40, 43], reducing computation and memory overhead via token reduction [2, 26, 28, 38], efficient attention [12, 32, 33], and hierarchical processing [9]. However, these strategies alone remain insufficient for very long sequences. Another line of work partitions long sequences into temporally ordered chunks processed sequentially [7, 14], enabling reconstruction from larger collections while keeping memory manageable. However, these pipelines rely on temporally ordered inputs and loop-closure constraints for global consistency, limiting their applicability when views are unordered.

Beyond these computational bottlenecks, we identify a more fundamental empirical phenomenon in VGGT: increasing the number of input views can even degrade reconstruction quality (Fig. 1). Our observations suggest that this behavior stems from viewpoint redundancy and uneven coverage within the input view set. In classical SfM, it is well known that near-duplicate views lead to small triangulation baselines and numerical instability. However, this issue manifests differently in VGGT, which performs multi-view reasoning through attention rather than explicit geometric triangulation. In VGGT, redundant views reduce geometric diversity and dilute attention by introducing many non-informative tokens that overwhelm critical geometric cues such as parallax and occlusion, required for reliable pose and depth estimation. These findings suggest that the key challenge in scaling VGGT lies not merely in the number of input views, but in the distribution of viewpoints provided to the model.

Motivated by this observation, we revisit VGGT from the perspective of view organization. Instead of asking how to process more views efficiently, we ask how views should be organized to maximize geometric reasoning under limited attention capacity. To this end, we propose a training-free, plug-and-play framework that improves both reconstruction quality and computational efficiency through diversity-driven graph partitioning. Rather than discarding or merging redundant views, our method reorganizes the entire collection into a set of geometrically diverse partitions. By promoting viewpoint diversity within each subset, the model can focus on complementary views and extract stronger geometric cues under limited attention capacity. This leads to improved accuracy and scalability even for long, unordered sequences.

To construct these chunks, we formulate view partitioning as a combinatorial graph optimization problem that maximizes visual dissimilarity and spatial dispersion. Because spatial dispersion requires camera poses that are unavailable at initialization, we introduce a soft pose propagation strategy that estimates poses for a small set of seed frames and propagates them to the remaining frames based on visual similarity. This enables an efficient approximation of spatial diversity without additional forward passes or ground-truth data.

Extensive experiments show that our method improves camera pose estimation, multi-view depth estimation, and 3D reconstruction while reducing GPU memory consumption and inference latency compared to vanilla VGGT[1]. Our method with vanilla VGGT also outperforms existing efficiency-oriented VGGT variants [28, 33] in overall geometric performance. Unlike prior approaches that trade accuracy for efficiency, our method improves performance while reducing computational cost. When combined with these variants, it further improves efficiency while maintaining competitive accuracy. Moreover, performance remains stable as the number of input frames increases, enabling scalable multi-view reconstruction in scenarios that previously resulted in out-of-memory failures.

Our contributions are summarized as follows:

- We reveal that viewpoint distribution plays a critical role in multi-view geometry transformers, showing that simply increasing the number of views without sufficient viewpoint diversity can degrade reconstruction performance.
- We propose a training-free, plug-and-play framework for VGGT that reorganizes input views into diversity-aware balanced partitions via combinatorial graph optimization over visual dissimilarity and spatial dispersion.
- To enable spatially informed partitioning without requiring camera poses, we introduce a soft pose propagation strategy that approximates spatial relationships from a small set of seed frames.
- Our method improves camera pose estimation, multi-view depth prediction, and 3D reconstruction accuracy while reducing memory usage and inference latency, and further complements existing VGGT variants to enhance scalability.

[1] In our experiments, we use the memory-efficient VGGT implementation from [26], denoted as VGGT*.

2 Related Work

Feed-Forward Multi-View Models. Classical structure-from-motion (SfM) pipelines [21, 24, 36] rely on feature matching [16, 17, 23, 31, 41], geometric verification, and bundle adjustment to recover camera poses and 3D structure from unordered image collections. While robust, these pipelines are computationally expensive and difficult to parallelize. Recent work has moved toward feed-forward approaches that directly regress 3D quantities from image sets using learned transformers. DUSt3R [37], MASt3R [15] and Splatt3R [29] predict dense pointmaps from image pairs, requiring global alignment to fuse pairwise predictions into a coherent reconstruction. VGGT [35] extends this paradigm to the multi-view setting: given N input frames, a single forward pass through an alternating self- and cross-attention transformer jointly predicts camera poses, dense depth, and 3D point maps for all frames simultaneously. While this joint reasoning enables strong performance, it incurs quadratic memory and compute scaling in attention, limiting the number of frames that can be processed in a single batch.

Efficient VGGT Variants. Several concurrent works address the scalability bottleneck of VGGT through architectural modifications. LiteVGGT [28] and FastVGGT [26] reduce token count via progressive token merging, lowering per-layer cost at the expense of spatial resolution. FasterVGGT [33] and AVGGT [32] replace dense global attention with sparse or approximate attention mechanisms that attend to only a subset of the most relevant tokens. VGGT-Long [7] takes a system-level approach, partitioning temporally ordered sequences into overlapping chunks and stitching them via sequential alignment with loop closure, targeting kilometer-scale outdoor reconstruction. In contrast, our approach operates at the input partitioning level and does not assume temporal ordering among frames.

View Selection and Frame Partitioning. The problem of selecting informative views has a long history in active vision and SfM [21,24], typically formulated as next-best-view selection [4,5,25,42] or keyframe sampling [8,10,18]. In the context of feed-forward models, the question shifts from which frames to acquire to how to *partition* a fixed set of frames into chunks that each support high-quality reconstruction. Existing chunking strategies for VGGT [7, 14, 40] use temporal stride or sequential windowing, which does not account for the visual content of the frames. We instead formulate frame partitioning as a combinatorial optimization problem over a pairwise visual distance graph derived from frozen DINOv2 [19] features. Our approach draws on the Kernighan–Lin (KL) local search [11] for balanced graph partitioning, adapting the classical 2-opt swap procedure to maximize within-chunk viewpoint diversity while maintaining balanced coverage across chunks. To our knowledge, this is the first work to treat frame-to-chunk assignment as an explicit optimization problem for feed-forward multi-view models.

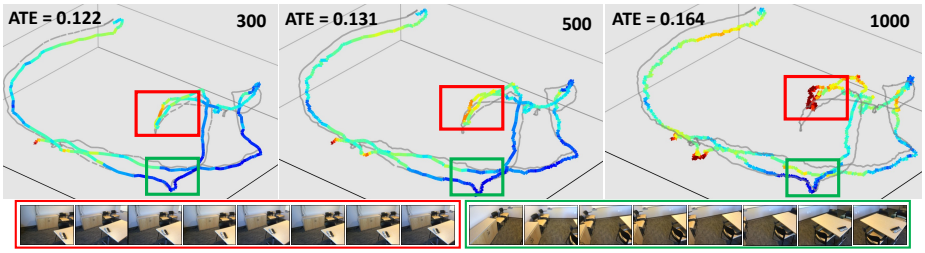

Fig. 2: Impact of frame count on VGGT performance. Red boxes indicate high-error regions where VGGT struggles with many frames, while green boxes indicate low-error regions. Frames sampled from each region show that failures occur in visually redundant views, whereas successful regions exhibit greater viewpoint diversity.

3 Motivation

In practical multi-view capture scenarios, frames are often densely sampled along camera trajectories, resulting in long sequences containing many visually similar views. While such dense sampling increases the number of observations, it also introduces substantial redundancy that can hinder effective multi-view reasoning. This is reflected in practice: as shown in Fig. 1, increasing the number of input frames often leads to not only significantly higher latency but also degraded reconstruction accuracy for existing VGGT variants [28, 33, 35].

Motivated by this phenomenon, we analyze the behavior of VGGT under densely sampled sequences (Fig. 2). By examining sequences of the same scene captured at different frame rates, we observe that reconstruction errors tend to increase as the frame rate grows. To investigate this, we uniformly sample frames from regions with different reconstruction errors (red boxes indicate high-error regions, green boxes indicate low-error regions). We find that high-error regions often contain many visually redundant views, whereas low-error regions tend to exhibit more diverse viewpoints.

To better examine this behavior, we measure frame-level attention entropy under different frame counts and sampling strategies. From the same sequences, we construct subsets of 300 and 500 frames, each sampled either redundantly (visually similar frames) or sparsely (diverse viewpoints). As shown in Fig. 3, the normalized attention entropy increases as the number of frames grows. Moreover, for the same frame count, redundantly sampled chunks exhibit higher entropy than sparsely sampled ones. This trend is consistent with attention dilution, where attention becomes more evenly distributed across many similar views rather than concentrating on geometrically informative ones.

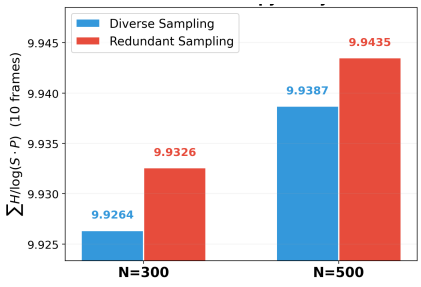

Fig. 3: Attention entropy analysis.

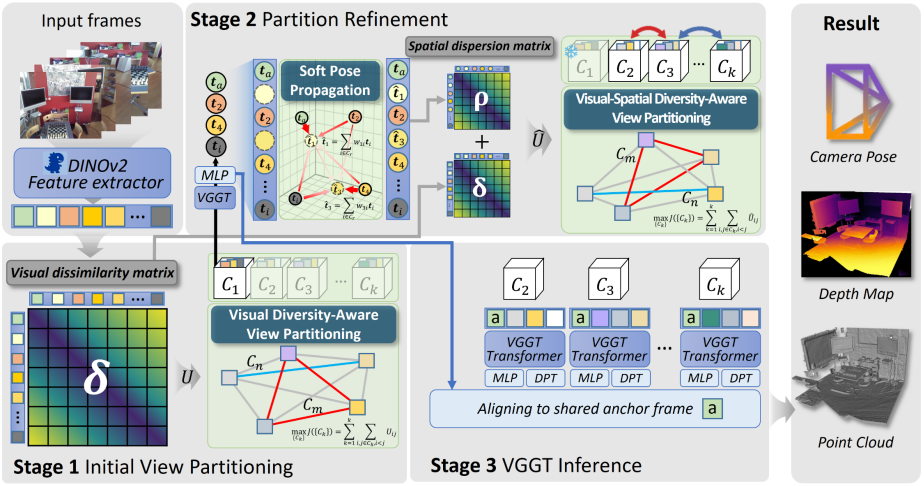

Fig. 4: Overview of the proposed framework. Input frames are first partitioned using visual dissimilarity. A selected reference chunk is processed by VGGT to obtain pseudo-spatial cues, which are used to estimate pseudo-poses for the remaining frames. The frames are then reorganized into balanced chunks that maximize both visual and spatial diversity, and each chunk is processed independently by VGGT.

These findings suggest that the main challenge in long sequences lies not in the number of frames but in how viewpoints are distributed: diversity is what matters most. Motivated by this insight, we reorganize the input views into balanced chunks that promote viewpoint diversity across frames, enabling the transformer to focus on complementary views while mitigating attention dilution and improving both efficiency and reconstruction accuracy.

4 Method

Building on this insight, our goal is to reorganize the input views into balanced chunks that promoting viewpoint diversity, enabling efficient and reliable multi-view reasoning in geometry transformers. Rather than processing all views jointly, we partition them into chunks with complementary visual and spatial characteristics.

We formulate this as a diversity-aware balanced partitioning problem. We first build an initial partition from visual dissimilarity, grouping N frames into K balanced chunks with high visual diversity. Since camera poses are unavailable at initialization, we approximate spatial relationships via a lightweight soft pose propagation strategy based on visual similarity from a few seed frames. The partition is then refined by a local search that jointly improves visual and spatial diversity while preserving balanced chunk sizes. Finally, each chunk is processed independently by VGGT, and the predictions are aligned through a shared anchor frame for global consistency, as illustrated in Fig. 4.

4.1 VGGT Preliminaries

Feature Backbone. Given a sequence of N RGB images $\{I_i\}_{i=1}^{N}$, VGGT [35] is a feed-forward transformer that jointly predicts per-frame 3D scene attributes in a single pass. Each image is tokenized into patch embeddings by a frozen DINOv2 [19] encoder, then processed by L=24 layers of alternating frame-wise and global self-attention, followed by task-specific prediction heads. All outputs are expressed in the coordinate frame of the first (reference) view I_1.

Prediction Heads. The *camera head* takes the per-frame camera token and decodes it into a 9-dimensional pose encoding $\mathbf{g}_i \in \mathbb{R}^9$ comprising rotation and translation parameters, from which the full camera extrinsic $T_i \in \mathrm{SE}(3)$ and intrinsic $K_i \in \mathbb{R}^{3\times3}$ are recovered. The *depth head* and *point head*, both based on DPT [22], decode the patch tokens into dense depth maps in $\mathbb{R}^{H\times W}$ and point maps in $\mathbb{R}^{H\times W\times3}$, respectively.

4.2 Visual Diversity-Aware Initial View Partitioning

Since camera poses are unavailable at initialization, this stage partitions frames using *visual diversity* alone, deferring spatial information to Sec. 4.3. We estimate pairwise visual dissimilarity between frames and use it to drive a balanced, diversity-maximizing partition.

Visual Dissimilarity. To measure visual dissimilarity, we reuse the frozen feature extractor—the encoder built into VGGT [35]—to obtain per-frame embeddings, and mean-pool the M patch tokens of each frame into a single descriptor:

$$\mathbf{z}_i = \frac{1}{M}\sum_{m=1}^{M}\mathbf{f}_{i,m}, \quad \mathbf{z}_i \in \mathbb{R}^d, \tag{1}$$

where $\mathbf{f}_{i,m}$ denotes the m-th patch token of frame i produced by the feature extractor.

We compute the visual dissimilarity δ_{ij} between frames i and j as:

$$\delta_{ij} = 1 - s_{ij}, \quad \text{where} \quad s_{ij} = \frac{\mathbf{z}_i^\top \mathbf{z}_j}{\|\mathbf{z}_i\|\,\|\mathbf{z}_j\|}. \tag{2}$$

Intuitively, frames with larger visual dissimilarity are more likely to correspond to different viewpoints or scene appearances, providing complementary information for multi-view reasoning.

Balanced View Partitioning via Local Swap Optimization. Given N input frames, we partition them into K balanced chunks so that each chunk contains views with diverse visual and spatial characteristics.

Let $\mathcal{C}_k$ denote chunk k, i.e., the set of frames assigned to it, where $k \in \{1,\ldots,K\}$, subject to $|\mathcal{C}_k| \le c$. We denote by U_{ij} the diversity score between

Algorithm 1 K-way Balanced Graph Partitioning via Local Swap

Require: Diversity score matrix $\mathbf{U} \in \mathbb{R}^{N \times N}$, chunks K, capacity c, iterations T
Ensure: Partition $\{\mathcal{C}_1, \ldots, \mathcal{C}_K\}$
Initialize $\{\mathcal{C}_k\}_{k=1}^{K}$ with a random balanced split
for $t = 1, \ldots, T$ **do**
  improved $\leftarrow$ `false`
  for each chunk pair (k_1, k_2) with $k_1 < k_2$ **do**
    $(i^*, j^*) \leftarrow \arg\max_{i \in \mathcal{C}_{k_1},\, j \in \mathcal{C}_{k_2}} \Delta_{ij}$ {Eq. (4)}
    if $\Delta_{i^* j^*} > 0$ **then** swap $i^* \leftrightarrow j^*$; improved $\leftarrow$ `true`; **end if**
  end for
  if $\neg$ improved **then break**; **end if**
end for
return $\{\mathcal{C}_1, \ldots, \mathcal{C}_K\}$

frames i and j. We formulate view partitioning as maximizing the total intra-chunk diversity score $\mathcal{J}$:

$$\max_{\{\mathcal{C}_k\}} \mathcal{J}(\{\mathcal{C}_k\}) = \sum_{k=1}^{K} \sum_{i,j \in \mathcal{C}_k,\, i<j} U_{ij}, \tag{3}$$
$$\text{s.t. } \{\mathcal{C}_k\} \le c, \quad \mathcal{C}_k \cap \mathcal{C}_{k'} = \emptyset,\ k \neq k', \quad \bigcup_{k=1}^{K} \mathcal{C}_k = \{1, \ldots, N\}.$$

Here, a score matrix $\mathbf{U} \in \mathbb{R}^{N \times N}$ is initialized with visual dissimilarity ($\mathbf{U} = \boldsymbol{\delta}$).

We call a partition *balanced* when intra-chunk diversity is comparable across all chunks, so that no chunk is left with disproportionately redundant views. While Eq. (3) maximizes intra-chunk diversity, it does not by itself equalize this diversity across chunks; rather than adding a penalty term, we restrict the feasible set to balanced partitions and enforce balance structurally through the optimizer. Alg. 1 starts from a split satisfying the capacity bound $|\mathcal{C}_k| \le \lceil N/K \rceil \le c$ (with $K = \lceil N/c \rceil$) and updates only through size-preserving swaps, so this bound holds throughout. Balance arises from the same swaps: since each accepted swap has positive gain, it tends to move frames from lower- to higher-diversity chunks, equalizing intra-chunk diversity as the optimization converges.

Since solving this combinatorial problem exactly is expensive, we adopt a greedy local swap strategy inspired by the Kernighan–Lin algorithm [11]. Starting from a random balanced partition, we iteratively swap frame pairs across chunks to increase $\mathcal{J}$ while maintaining balanced sizes. For chunks $(\mathcal{C}_{k_1}, \mathcal{C}_{k_2})$, the gain of swapping $i \in \mathcal{C}_{k_1}$ and $j \in \mathcal{C}_{k_2}$ is

$$\Delta_{ij} = \big(u_{k_2}(i) - u_{k_1}(i)\big) + \big(u_{k_1}(j) - u_{k_2}(j)\big) - 2\,U_{ij}, \tag{4}$$

where $u_k(i) = \sum_{m \in \mathcal{C}_k} U_{im}$. We repeatedly apply swaps with positive gain until convergence. The full procedure is summarized in Alg. 1.

4.3 Spatial Diversity-Aware Partition Refinement

Once approximate poses become available, this stage refines the partition along the *spatial diversity* axis. Since running VGGT on every frame would be costly, we estimate poses for a single reference chunk and propagate them to the remaining frames via visual similarity, then use the resulting spatial dispersion to refine the visual-only partition.

Seed Chunk Pose Estimation. To approximate spatial relationships between views, we first estimate camera poses from a small reference chunk of frames. Specifically, after the initial visual diversity-based partitioning, we simply select first chunk $\mathcal{C}_1$ as reference chunk ($\mathcal{C}_r$) and perform a single forward pass of VGGT to estimate their camera poses. Let $\mathbf{p}_i = (\mathbf{R}_i, \mathbf{t}_i)$ denote the predicted pose of frame i in this reference chunk $\mathcal{C}_r$. These estimated poses serve as geometric anchors for the remaining frames.

Soft Pose Propagation. While camera poses can be estimated reliably within the reference chunk, the remaining frames do not have direct pose estimates. To approximate their spatial relationships without additional forward passes, we propagate geometric information from the reference chunk to the remaining frames using visual similarity. For a frame $l \notin \mathcal{C}_r$, we approximate its pose by propagating pose information from the reference chunk using visual similarity:

$$\hat{\mathbf{t}}_l = \sum_{i \in \mathcal{C}_r} w_{li} \mathbf{t}_i, \quad \text{where} \quad w_{li} = \frac{\exp(s_{li}/\gamma)}{\sum_{j \in \mathcal{C}_r} \exp(s_{lj}/\gamma)}. \tag{5}$$

Here s_{li} denotes the cosine similarity between frames l and i in the feature embedding space (Eq. (2)), and γ controls the sharpness of the soft assignment. Although these estimates are not intended to be accurate camera poses, they provide sufficient geometric cues to approximate spatial dispersion across views. Given pseudo poses $\hat{\mathbf{t}}_i$ for all N frames, we then compute a pairwise spatial dispersion matrix $\boldsymbol{\rho}$:

$$\rho_{ij} = 1 - \exp\left(-\frac{\|\hat{\mathbf{t}}_i - \hat{\mathbf{t}}_j\|}{\tau}\right) \in [0, 1], \tag{6}$$

where τ is set to the median pairwise distance to ensure scale invariance. This term assigns larger scores to pairs of frames that are spatially distant.

Visual–spatial Diversity-aware View Partitioning. We incorporate this spatial dispersion into the diversity score matrix as

$$\hat{U}_{ij} = \delta_{ij} + \epsilon \cdot \rho_{ij}, \tag{7}$$

where ϵ controls the relative contribution of visual and spatial diversity. Using the updated score matrix $\hat{\mathbf{U}}$, Eq. (3) and Alg. 1 are applied to further refine the remaining $K-1$ chunks while keeping the reference chunk fixed. This refinement yields visual–spatial diversity-aware view partitions that jointly improve visual diversity and spatial dispersion with minimal additional overhead.

4.4 VGGT Inference on Balanced View Partitions

Given the updated partition $\{\mathcal{C}_k\}_{k=2}^{K}/\mathcal{C}_r$, we perform VGGT inference independently on each chunk, where the anchor frame a is placed at the first position to serve as the reference view. After first partitioning, each chunk is augmented with a shared *anchor frame* (frame 0 by default) inserted at position 0. This ensures that every chunk contains a common reference, enabling post-hoc alignment without any optimization. After obtaining per-chunk predictions, we align all chunks to the reference chunk C_r using the predicted pose of the shared anchor frame. We treat each pose $\mathbf{p}$ as an element of $SE(3)$, where $\cdot$ denotes composition and $(\cdot)^{-1}$ the inverse transform:

$$T_{\text{align}}^{(k)} = \hat{\mathbf{p}}_a^{(r)} \cdot (\hat{\mathbf{p}}_a^{(k)})^{-1} \in SE(3), \tag{8}$$

All poses predicted in chunk k are then transformed as $\tilde{\mathbf{p}}_i^{(k)} = T_{\text{align}}^{(k)} \cdot \hat{\mathbf{p}}_i^{(k)}$. Because the anchor frame is shared across all chunks, this alignment provides a consistent global coordinate frame while avoiding additional global optimization.

5 Experiments

5.1 Implementation Details

Datasets. We evaluate across six datasets spanning four tasks. We assess camera pose estimation on 7Scenes [27] and NRGBD [1], multi-view depth on Bonn [20], and 3D reconstruction on 7Scenes, NRGBD, and ScanNet-50 [6]. We further evaluate long-sequence SLAM on TUM-RGBD [30] and outdoor reconstruction on Tanks&Temples (T&T) [13].

Metrics. We report AUC of $\max(\text{RRA}, \text{RTA})$ at $(3°, 15°, 30°)$ for camera pose estimation, Abs Rel and $\delta < 1.25$ for multi-view depth, and Chamfer distance (CD), accuracy (Acc.), completeness (Comp.), and normal consistency (NC) for 3D reconstruction. For long-sequence and outdoor evaluation, we additionally report Absolute Trajectory Error (ATE), F1 at threshold τ, and throughput (FPS).

Experimental Setup. We build on VGGT [35] and run all experiments on a single NVIDIA A100 GPU (80 GB). Following FastVGGT [26], we retain activations only at layers $\{4, 11, 17, 23\}$ to reduce peak memory. Since the original VGGT cannot exceed ∼300 frames on an 80 GB GPU, we use VGGT* [26], a memory-efficient re-implementation with comparable accuracy for longer inputs. Operating only on input partitioning, our method applies without modification to LiteVGGT [28] (token merging), FasterVGGT [33] (sparse attention), and the permutation-equivariant transformer π^3 [39], on which we further assess generalization against its efficient variants, FastVGGT [26] and FasterVGGT.

Table 1: Camera pose estimation on 7Scenes [27].

Frames	Model	AUC@3↑	AUC@15↑	AUC@30↑	VRAM(GB)↓	Latency(s)↓
300	VGGT* [35]	0.2412	0.6980	0.8269	18.76	54.16
	VGGT* [35] + Ours	**0.2481**	**0.7020**	**0.8292**	**12.68**	**27.65**
	FasterVGGT [33]	0.1662	0.6504	0.7992	25.18	38.87
	FasterVGGT [33] + Ours	**0.1760**	**0.6559**	**0.8024**	**12.98**	**25.99**
	LiteVGGT [28]	**0.2234**	**0.6816**	**0.8165**	16.43	15.35
	LiteVGGT [28] + Ours	0.2068	0.6698	0.8086	**10.04**	**9.84**
500	VGGT* [35]	0.2359	0.6942	0.8244	28.41	149.68
	VGGT* [35] + Ours	**0.2486**	**0.7013**	**0.8286**	**16.99**	**49.09**
	FasterVGGT [33]	0.1628	0.6462	0.7960	40.90	93.94
	FasterVGGT [33] + Ours	**0.1758**	**0.6546**	**0.8009**	**15.65**	**42.60**
	LiteVGGT [28]	0.2089	0.6691	0.8071	25.94	33.51
	LiteVGGT [28] + Ours	**0.2117**	**0.6736**	**0.8108**	**11.91**	**15.97**
1000	VGGT* [35]	0.2201	0.6853	0.8183	69.56	584.72
	VGGT* [35] + Ours	**0.2467**	**0.6998**	**0.8274**	**18.32**	**87.32**
	FasterVGGT [33]	*OOM*	*OOM*	*OOM*	*OOM*	*OOM*
	FasterVGGT [33] + Ours	**0.1750**	**0.6534**	**0.7998**	**18.31**	**83.84**
	LiteVGGT [28]	0.1877	0.6198	0.7610	48.62	95.11
	LiteVGGT [28] + Ours	**0.2083**	**0.6707**	**0.8087**	**13.98**	**31.68**

Table 2: Multi-view Depth Estimation on Bonn [20].

Frames	Model	Abs Rel↓	δ_1↑	VRAM(GB)↓	Latency(s)↓
500	VGGT* [35]	**0.048**	**0.961**	28.53	156.47
	VGGT* [35] + Ours	**0.048**	0.954	**15.78**	**65.84**
	FasterVGGT [33]	**0.046**	**0.959**	41.02	102.6
	FasterVGGT [33] + Ours	0.047	0.956	**15.78**	**49.92**
	LiteVGGT [28]	0.057	**0.964**	26.01	35.55
	LiteVGGT [28] + Ours	**0.055**	0.956	**11.98**	**14.69**

5.2 Quantitative Results

Camera Pose Estimation. As shown in Tab. 1, our method reduces VRAM and latency across all VGGT variants while maintaining or improving pose accuracy. On VGGT [35], it surpasses VGGT* at all sequence lengths, with up to 6.34× speedup and 3.8× lower VRAM, and higher AUC. On FasterVGGT [33] and LiteVGGT [28], accuracy is preserved or improved on long sequences, with up to 3.5× VRAM reduction and 3.0× lower latency on long sequences. Overall, our approach scales to long inputs: accuracy remains stable, while baselines degrade or run out of memory. At 1000 frames, FasterVGGT fails on an A100 (80 GB), while our method enables all three variants to complete inference successfully.

Multi-view Depth Estimation. Tab. 2 reports multi-view depth estimation on Bonn [20] at 500 frames. Our method reduces VRAM by up to 2.6× and latency by up to 2.4× across all variants, while depth accuracy remains virtually

Table 3: Multi-view 3D reconstruction on ScanNet-50 [6].

Frames	Model	Acc.↓	Comp.↓	NC↑	VRAM(GB)↓	Latency(s)↓
500	VGGT* [35]	0.029	0.024	**0.732**	28.53	155.97
	VGGT* [35] + Ours	**0.025**	**0.019**	0.724	**19.13**	**59.73**
	FasterVGGT [33]	**0.032**	0.024	**0.703**	41.02	99.87
	FasterVGGT [33] + Ours	**0.032**	**0.021**	0.682	**15.78**	**51.94**
	LiteVGGT [28]	0.039	0.047	**0.716**	26.01	36.03
	LiteVGGT [28] + Ours	**0.029**	**0.018**	0.700	**11.98**	**17.28**

Table 4: Multi-view 3D reconstruction on 7Scenes [27] and NRGBD [1]

Frames	Model	7Scenes			NRGBD		
		CD↓	NC↑	L(s)↓	CD↓	NC↑	L(s)↓
500	VGGT* [35]	0.045	0.648	155.38	0.037	0.810	155.95
	VGGT* [35] + Ours	**0.042**	**0.649**	**59.04**	**0.028**	**0.851**	**58.32**
	FasterVGGT [33]	0.046	**0.624**	99.57	0.048	0.742	103.5
	FasterVGGT [33] + Ours	**0.043**	0.613	**51.69**	**0.041**	**0.743**	**49.81**
	LiteVGGT [28]	0.049	**0.626**	34.48	0.069	0.743	40.18
	LiteVGGT [28] + Ours	**0.045**	0.624	**17.16**	**0.049**	**0.745**	**16.84**

unchanged—Abs Rel differences stay within 0.002 and δ_1 within 0.008. This confirms that the efficiency gains from our partitioning transfer to dense prediction tasks without compromising depth quality.

Multi-view 3D Reconstruction. Tab. 3 and Tab. 4 evaluate 3D reconstruction on ScanNet-50 [6], 7Scenes [27], and NRGBD [1] at 500 frames. On ScanNet-50, our method improves completeness for all baselines and maintains or improves accuracy with less VRAM and lower latency. On VGGT [35], it improves Chamfer distance (CD) on both datasets while reducing latency by 2.6×. For FasterVGGT [33] and LiteVGGT [28], it reaches competitive quality with up to 2.4× speedups, sometimes surpassing the baseline in normal consistency (NC). These results confirm that our diversity-aware chunking generalizes across tasks, improving dense 3D reconstruction alongside pose and depth estimation.

Comparison with Memory-based and Chunking Baselines. We evaluate long-sequence SLAM (TUM-RGBD [30]) and outdoor reconstruction (Tanks&Temples [13]) against memory-based Spann3R [34] and MUSt3R [3], and chunking-based VGGT-Long (VL) [7]. As shown in Tab. 5, our method achieves the best accuracy on both benchmarks, reducing ATE by up to 3.4× over VGGT-Long at competitive throughput, extending to outdoor and long-sequence regimes with a strong accuracy–efficiency trade-off.

Generalization to Other Architectures. Since our framework operates only on input partitioning, it is not tied to VGGT [35]. Applying it to π^3 [39] (Tab. 6) preserves pose accuracy while cutting VRAM up to 3.6× and latency up to

Table 5: Pose estimation on TUM-RGBD [30] and 3D reconstruction on T&T [13].

Dataset	Metric	Spann3R [34]	MUSt3R-C [3]	VL [7]	VGGT [35]+Ours
TUM-RGBD	ATE [cm] ↓	38.74	<u>6.43</u>	13.44	**3.94**
	FPS ↑	**15.35**	<u>10.25</u>	1.32	6.32
T&T	F1@$\tau\times10$ ↑	0.37	0.56	<u>0.63</u>	**0.74**
	FPS ↑	**9.79**	5.47	1.36	<u>7.61</u>

Table 6: Generalization to π^3 [39] on camera pose estimation on NRGBD [1]

Model	AUC@30↑	VRAM(GB)↓	L(s)↓
π^3 [39]	**0.9745**	<u>28.96</u>	177.97
+ FasterVGGT [33]	0.9423	37.54	103.04
+ FastVGGT [26]	0.9614	30.50	<u>72.56</u>
+ Ours	<u>0.9717</u>	**7.95**	**43.63**

4.1×. In contrast, methods that modify model internals, such as sparse attention (FasterVGGT [33]) and token merging (FastVGGT [26]), degrade π^3 accuracy with smaller gains, being tailored to a specific attention or tokenization scheme. This shows that organizing input views, rather than altering the network, generalizes beyond VGGT and its variants. Note that we use FastVGGT [26] instead of LiteVGGT [28] due to compatibility.

5.3 Qualitative Results

Pseudo-Pose Estimation. Fig. 5 shows the soft-assigned pseudo-positions: blue points are ground-truth camera centers, green points are VGGT predictions from the reference chunk, and red points are pseudo-positions for the remaining frames from Eq. (5). Our soft pose propagation closely follows the ground-truth trajectory, recovering the scene layout from a single chunk inference. Although only approximate, these pseudo-positions provide useful geometric cues for partitioning when combined with appearance scores.

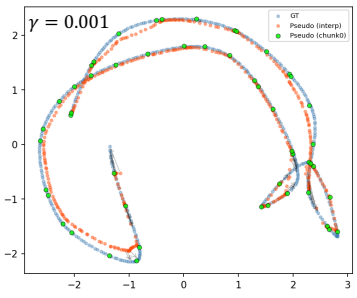

Fig. 5: Visualization of Pseudo-pose on NRGBD [1].

5.4 Ablation Studies

Partitioning Strategy. Tab. 7 compares different sequence partitioning strategies. Sequential partitioning collapses, as temporally adjacent frames are visually redundant and yield low-diversity chunks; K-means acts similarly, since it groups visually similar views together and thus *minimizes* rather than maximizes within-chunk diversity. Random partitioning recovers much of the performance by breaking local redundancy, but still lacks explicit view organization. Our

Table 7: Partitioning strategy comparison on NRGBD [1].

Methods	AUC@3↑	AUC@15↑	AUC@30↑
Random	0.6819	0.8955	0.9412
Sequential	0.3524	0.6559	0.7640
K-means	0.3557	0.6644	0.7733
Ours	**0.7844**	**0.9411**	**0.9691**

Table 8: Diversity-cue ablation on NRGBD [1].

Methods	AUC@3↑	(+Δ)
Random Init.	68.19	
w/ Spat. Disp. ($\boldsymbol{\rho}$)	77.87	+ 9.68
w/ Vis. dissim. ($\boldsymbol{\delta}$)	78.21	+ 10.02
w/ Both ($\boldsymbol{\delta}$+$\epsilon\boldsymbol{\rho}$) (Ours)	**78.44**	**+ 10.25**

diversity-aware partitioning, which directly maximizes within-chunk dissimilarity, performs best across all AUC thresholds, confirming that view organization—not view count—is the source of the gains.

Diversity cue. We adopt the KL algorithm [11], applying the best-improving swap at each step until no further improvement is found within a partition pair. Despite this approximation, the objective $\mathcal{J}$ consistently converges within 5–10 outer iterations (See Sec. A.2 in supplementary). The soft pose assignment of Eq. (5) provides a further gain, as shown in Tab. 8, helping most in scenes where visually diverse viewpoints are spatially close, which appearance alone cannot distinguish.

Soft pose cue. We further analyze the pose cue in our soft pose assignment by comparing its rotation and translation terms, as shown in Tab. 9. Since rotation captures viewing-direction similarity rather than camera-center dispersion, we ablate it to isolate the spatial signal: translation alone (T) outperforms both rotation alone (R) and their combination (T+R), confirming that camera-center translation is the most effective spatial cue for partitioning.

Table 9: Soft pose cue ablation on 7Scenes [27].

Metric	T	T+R	R
AUC@3	**0.2475**	0.2361	0.2252

Chunk Size Analysis. The chunk capacity c controls the trade-off between reconstruction quality and computational cost, and—since our partitioning promotes within-chunk diversity—also governs the diversity–sparsity balance. Smaller chunks lower per-chunk complexity and improve scalability but sample each chunk more sparsely, increasing spatial sparsity; larger chunks provide denser coverage at higher memory and compute cost. Sparsity is thus not adjusted by the method itself but exposed through the chunk-size choice, while the balanced-partition constraint applies it uniformly across chunks. We evaluate this on 7Scenes [27] in Tab. 10 and further analyze the trade-off between chunk diversity and sparsity in the supplementary (Tab. 18).

Component-wise Inference Time Analysis. We measure the latency of VGGT [35] + our framework on Redkitchen scene from the 7Scenes [27] pose estimation task. As shown in Tab. 11, similarity computation, partitioning, and

Table 10: Chunk capacity ablation on (VGGT* [35] + Ours, 7Scenes [27]).

Frames	Size	AUC@3↑	AUC@15↑	AUC@30↑	VRAM(GB)↓	Latency(s)↓
1000	25	0.2399	0.6951	0.8241	18.32	**80.13**
	50	0.2467	0.6998	0.8273	18.32	87.32
	100	**0.2478**	**0.7008**	**0.8281**	18.32	102.0

Table 11: Component-wise latency (s) on the RedKitchen scene from the 7Scenes [27]

Frames	Sim. Matrix	Partitioning	Alignment	Transformer	Total
300	0.0036	0.0475	0.0017	27.685	27.753
500	0.0071	0.0938	0.0158	49.776	49.893
1000	0.0098	0.3426	0.0286	91.735	92.116

alignment introduce only a small overhead compared to the VGGT inference time. Even for 1000 frames, these additional steps take less than a second in total, while the runtime is dominated by the transformer inference. This indicates that our framework improves view organization with negligible additional computational cost.

6 Conclusion

In this paper, we analyze the impact of view redundancy in geometry transformers and show that increasing the number of views without sufficient viewpoint diversity can degrade reconstruction performance by diffusing attention across similar observations. Motivated by this insight, we propose a training-free, plug-and-play framework that reorganizes input views into diverse chunks for more effective multi-view reasoning. By combining diversity-aware partitioning with soft pose propagation, our method improves camera pose estimation, multi-view depth prediction, and 3D reconstruction accuracy while reducing memory usage and inference latency. The proposed framework can be seamlessly integrated with existing VGGT variants, providing a practical solution for scalable long-sequence multi-view reconstruction.

Acknowledgement

This work was supported by the AI Graduate School Program at POSTECH (RS-2019-II191906 (5%)), the NRF grants (RS-2025-24535146(10%), RS-2026-25491789), and the IITP grants (RS-2022-II220926 (30%), RS-2024-00457882 (20%), RS-2026-25518317 (35%) funded by MSIT, Korea.

References

1. Azinović, D., Martin-Brualla, R., Goldman, D.B., Nießner, M., Thies, J.: Neural rgb-d surface reconstruction. In: CVPR (2022) 10, 12, 13, 14
2. Bolya, D., Fu, C.Y., Dai, X., Zhang, P., Feichtenhofer, C., Hoffman, J.: Token merging: Your vit but faster. In: ICLR (2023) 2
3. Cabon, Y., Stoffl, L., Antsfeld, L., Csurka, G., Chidlovskii, B., Revaud, J., Leroy, V.: Must3r: Multi-view network for stereo 3d reconstruction. In: CVPR. pp. 1050–1060 (2025) 12, 13
4. Chen, X., Li, Q., Wang, T., Xue, T., Pang, J.: Gennbv: Generalizable next-best-view policy for active 3d reconstruction. In: CVPR. pp. 16436–16445 (2024) 4
5. Connolly, C.: The determination of next best views. In: ICRA (1985) 4
6. Dai, A., Chang, A.X., Savva, M., Halber, M., Funkhouser, T., Nießner, M.: Scannet: Richly-annotated 3d reconstructions of indoor scenes. In: CVPR (2017) 10, 12
7. Deng, K., Ti, Z., Xu, J., Yang, J., Xie, J.: Vggt-long: Chunk it, loop it, align it – pushing vggt's limits on kilometer-scale long rgb sequences. In: ICRA (2026) 2, 4, 12, 13
8. Jha, R., Zhou, Y., Loianno, G.: Adaptive keyframe selection for scalable 3d scene reconstruction in dynamic environments. arXiv preprint arXiv:2510.23928 (2025) 4
9. Kang, G., Yang, S., Nam, S., Lee, Y., Kim, J., Park, E.: Multi-view pyramid transformer: Look coarser to see broader. In: CVPR (2026) 2
10. Keetha, N., Karhade, J., Jatavallabhula, K.M., Yang, G., Scherer, S., Ramanan, D., Luiten, J.: Splatam: Splat track & map 3d gaussians for dense rgb-d slam. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 21357–21366 (2024) 4
11. Kernighan, B.W., Lin, S.: An efficient heuristic procedure for partitioning graphs. The Bell System Technical Journal **49**(2), 291–307 (1970) 4, 8, 14
12. Kim, Y., Song, W., Lew, J., Hwangbo, H., Lee, J., Yoon, S.: Hess: Head sensitivity score for sparsity redistribution in vggt. In: CVPR. pp. 36509–36517 (2026) 2
13. Knapitsch, A., Park, J., Zhou, Q.Y., Koltun, V.: Tanks and temples: Benchmarking large-scale scene reconstruction. ACM Transactions on Graphics **36**(4) (2017) 10, 12, 13
14. Lee, J., Lee, M., Yang, S., Kang, M., Lee, S.: Swiftvggt: A scalable visual geometry grounded transformer for large-scale scenes. arXiv preprint arXiv:2511.18290 (2025) 2, 4
15. Leroy, V., Cabon, Y., Revaud, J.: Grounding image matching in 3d with mast3r. In: ECCV (2024) 4
16. Li, X., Rao, T., Pan, C.: Edm: Efficient deep feature matching. In: ICCV. pp. 26198–26208 (2025) 4
17. Lindenberger, P., Sarlin, P.E., Pollefeys, M.: LightGlue: Local Feature Matching at Light Speed. In: ICCV (2023) 4
18. Mur-Artal, R., Montiel, J.M.M., Tardos, J.D.: Orb-slam: A versatile and accurate monocular slam system. IEEE transactions on robotics **31**(5), 1147–1163 (2015) 4
19. Oquab, M., Darcet, T., Moutakanni, T., Vo, H., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., et al.: Dinov2: Learning robust visual features without supervision. TMLR (2024) 4, 7
20. Palazzolo, E., Behley, J., Lottes, P., Giguere, P., Stachniss, C.: Refusion: 3d reconstruction in dynamic environments for rgb-d cameras exploiting residuals. In: IROS (2019) 10, 11

21. Pan, L., Barath, D., Pollefeys, M., Schönberger, J.L.: Global Structure-from-Motion Revisited. In: ECCV (2024) 4
22. Ranftl, R., Bochkovskiy, A., Koltun, V.: Vision transformers for dense prediction. In: ICCV (2021) 7
23. Sarlin, P.E., DeTone, D., Malisiewicz, T., Rabinovich, A.: Superglue: Learning feature matching with graph neural networks. In: CVPR. pp. 4938–4947 (2020) 4
24. Schönberger, J.L., Frahm, J.M.: Structure-from-motion revisited. In: CVPR (2016) 4
25. Scott, W.R., Roth, G., Rivest, J.F.: View planning for automated three-dimensional object reconstruction and inspection. ACM Computing Surveys **35**(1), 64–96 (2003) 4
26. Shen, Y., Zhang, Z., Qu, Y., Zheng, X., Ji, J., Zhang, S., Cao, L.: FastVGGT: Fast visual geometry transformer. In: ICLR (2026), https://openreview.net/forum?id=asl8NJlIMe 2, 3, 4, 10, 13
27. Shotton, J., Glocker, B., Zach, C., Izadi, S., Criminisi, A., Fitzgibbon, A.: Scene coordinate regression forests for camera relocalization in rgb-d images. In: CVPR (2013) 10, 11, 12, 14, 15
28. Shu, Z., Lin, C., Xie, T., Yin, W., Li, B., Pu, Z., Li, W., Yao, Y., Cao, X., Guo, X., Long, X.X.: Litevggt: Boosting vanilla vggt via geometry-aware cached token merging. In: CVPR. pp. 36422–36432 (June 2026) 2, 3, 4, 5, 10, 11, 12, 13
29. Smart, B., Zheng, C., Laina, I., Prisacariu, V.A.: Splatt3r: Zero-shot gaussian splatting from uncalibrated image pairs. arXiv preprint arXiv:2408.13912 (2024) 4
30. Sturm, J., Engelhard, N., Endres, F., Burgard, W., Cremers, D.: A benchmark for the evaluation of rgb-d slam systems. In: IROS (Oct 2012) 10, 12, 13
31. Sun, J., Shen, Z., Wang, Y., Bao, H., Zhou, X.: Loftr: Detector-free local feature matching with transformers. In: CVPR. pp. 8922–8931 (June 2021) 4
32. Sun, X., Zhu, Z., Lou, Z., Yang, B., Tang, J., Zhang, L., Wang, H., Zhang, J.: Avggt: Rethinking global attention for accelerating vggt. In: CVPR. pp. 251–260 (June 2026) 2, 4
33. Wang, C.S.B., Schmidt, C., Piekenbrinck, J., Leibe, B.: Faster vggt with block-sparse global attention. arXiv preprint arXiv:2509.07120 (2025) 2, 3, 4, 5, 10, 11, 12, 13
34. Wang, H., Agapito, L.: 3d reconstruction with spatial memory. In: 3DV. pp. 78–89. IEEE (2025) 12, 13
35. Wang, J., Chen, M., Karaev, N., Vedaldi, A., Rupprecht, C., Novotny, D.: Vggt: Visual geometry grounded transformer. In: CVPR (2025) 1, 2, 4, 5, 7, 10, 11, 12, 13, 14, 15
36. Wang, J., Karaev, N., Rupprecht, C., Novotny, D.: Vggsfm: Visual geometry grounded deep structure from motion. In: CVPR (2024) 4
37. Wang, S., Leroy, V., Cabon, Y., Chidlovskii, B., Revaud, J.: Dust3r: Geometric 3d vision made easy. In: CVPR (2024) 4
38. Wang, W., Meiner, L., Shubham, R., De La Parra, C., Kumar, A.: Httm: Head-wise temporal token merging for faster vggt. In: CVPR. pp. 26379–26388 (June 2026) 2
39. Wang, Y., Zhou, J., Zhu, H., Chang, W., Zhou, Y., Li, Z., Chen, J., Pang, J., Shen, C., He, T.: π^3: Permutation-equivariant visual geometry learning. In: ICLR (2026) 10, 12, 13
40. Wang, Z., Xu, D.: Flashvggt: Efficient and scalable visual geometry transformers with compressed descriptor attention. In: CVPR. pp. 21826–21835 (June 2026) 2, 4

41. Weinzaepfel, P., Leroy, V., Lucas, T., BRÉGIER, R., Cabon, Y., ARORA, V., Antsfeld, L., Chidlovskii, B., Csurka, G., Revaud, J.: Croco: Self-supervised pre-training for 3d vision tasks by cross-view completion. In: NeurIPS (2022) 4
42. Wilson, J., Almeida, M., Mahajan, S., Labrie, M., Ghaffari, M., Ghasemalizadeh, O., Sun, M., Kuo, C.H., Sen, A.: Pop-gs: Next best view in 3d-gaussian splatting with p-optimality. In: CVPR. pp. 3646–3655 (2025) 4
43. Yang, J., Sax, A., Liang, K.J., Henaff, M., Tang, H., Cao, A., Chai, J., Meier, F., Feiszli, M.: Fast3r: Towards 3d reconstruction of 1000+ images in one forward pass. In: CVPR (2025) 2