Linearly Scalable 3D Reconstruction via Covisibility-Guided Sparse Attention

Aarya Singh
Georgia Institute of Technology
asingh3003@gatech.edu

Jiangyue Zhu
Georgia Institute of Technology
jzhu484@gatech.edu

Raymond Zeng
Georgia Institute of Technology
rzeng42@gatech.edu

Abstract

Transformers have emerged as the dominant architecture for 3D reconstruction, enabling end-to-end inference of scene geometry. However, state-of-the-art methods like the Visual Geometry Grounded Transformer (VGGT) rely on dense global attention, imposing a quadratic $O(N^2)$ memory complexity that restricts reconstruction to small batches ($N < 200$). To overcome this bottleneck, we propose **Covisibility-Guided Sparse Attention**, a mechanism that integrates geometric priors from classical Structure-from-Motion (SfM) to prune redundant attention weights. By constructing a sparse viewing graph using visual place recognition embeddings (MegaLoc), we reduce complexity to $O(Nk)$. We evaluate reconstruction accuracy on the **CO3D v2 dataset** and benchmark scalability on the large-scale **DL3DV-10K dataset**. Our experiments demonstrate a **22.8% reduction in inference latency** at $N = 225$ images on DL3DV-10K compared to the dense baseline. Furthermore, we observe a **15% reduction in validation loss** on CO3D (from 0.23 to 0.20), indicating that geometric sparsity acts as a structural regularizer, improving generalization without sacrificing reconstruction fidelity.

1 Introduction

The field of 3D reconstruction has shifted from multi-stage pipelines to end-to-end deep learning models. The Visual Geometry Grounded Transformer (VGGT) [Wang et al., 2025b] exemplifies this, jointly predicting camera intrinsics, extrinsics, and dense point clouds in a single forward pass. However, VGGT's reliance on global self-attention introduces a quadratic $O(N^2)$ computational cost with respect to the number of input images N. For real-world applications such as autonomous drone mapping or city-scale digital twinning, where $N \gg 200$, this design is prohibitively expensive.

The Scalability Gap. Current transformer architectures for 3D reconstruction face a severe memory cliff. For a sequence of length N, the global self-attention mechanism requires storing an $N \times N$ attention matrix. While manageable for object-centric batches ($N \approx 50 - 100$), this scales poorly for real-world drone mapping where N often exceeds 1,500 images. Specifically, increasing N from 200 to 1,000 results in a $25\times$ increase in memory consumption. Existing solutions fail to bridge this gap: temporal chunking [Deng et al., 2025] requires ordered video streams, and token merging [Shen et al., 2025] sacrifices high-frequency spatial details required for accurate photogrammetry. Our method provides the first solution for *unordered*, *resolution-preserving*, and *linearly scalable* reconstruction.

Preprint.

1.1 Research Question and Success Criteria

Research Question: Can sparse, covisibility-guided attention reduce inference cost while preserving reconstruction quality in transformer-based 3D reconstruction?

Success Criteria:

- **Efficiency:** Achieve $> 20\%$ reduction in inference latency at $N > 200$.
- **Accuracy:** Maintain pose estimation and depth accuracy comparable to the dense baseline (validation loss within 5% margin).
- **Impact:** Enable the processing of sequences ($N > 1000$) on consumer-grade hardware (24GB VRAM), facilitating large-scale photogrammetry.

2 Related Work

Transformer-based 3D Reconstruction. VGGT [Wang et al., 2025b] employs a large vision transformer (1.2B parameters) trained on CO3D and RealEstate10K. Its Alternating-Attention mechanism interleaves linear frame-wise attention with quadratic global attention. While accurate, the global attention limits inputs to < 200 images on standard GPUs. Extensions like Faster VGGT [Wang et al., 2025a] have attempted to optimize this, but often rely on token merging that can degrade fine geometric details.

Efficient Transformers & Sparse Attention. Generic efficient transformers like Longformer and BigBird employ fixed sliding-window or random sparsity patterns to achieve $O(N)$ complexity. However, these patterns are structurally rigid and agnostic to the image content. In multi-view 3D reconstruction, "neighbors" are defined by geometric overlap, not sequence index. Our work differs by constructing a *dynamic, content-aware* sparsity mask derived from deep visual embeddings, ensuring that attention is focused on geometrically relevant pairs rather than arbitrary windows.

Standard Benchmarks. While classical Multi-View Stereo (MVS) evaluation often relies on datasets like **DTU** (object-centric) and **ScanNet** (indoor scenes), recent transformer-based approaches have shifted towards large-scale, in-the-wild collections. We select **CO3D v2** [Reizenstein et al., 2021] for our accuracy evaluation as it provides high-quality, object-centric sequences with accurate COLMAP-derived camera poses. Ideally suited for stressing scalability, we employ **DL3DV-10K** [Ling et al., 2023] for our inference benchmarks due to its vast scale and unbounded scenes.

Table 1: Comparison with state-of-the-art scalable reconstruction methods.

Method	Dataset	Complexity	Input Assumption	Sparsity Type	Resolution
VGGT [Wang et al., 2025b]	CO3D	$O(N^2)$	Unordered	None (Dense)	Full
VGGT-Long [Deng et al., 2025]	KITTI	$O(N)$	**Ordered (Video)**	Temporal Chunk	Full
FastVGGT [Shen et al., 2025]	RE10K	Sub-$O(N^2)$	Unordered	Token Merge	**Reduced**
Ours	**CO3D**	$O(Nk)$	**Unordered**	**Geometric Graph**	**Full**

3 Method

We propose a **Covisibility-Guided Sparse Attention** mechanism. Our approach replaces the dense attention matrix A with a sparse binary mask M, derived from a geometric prior.

3.1 Hypotheses

- **H1 (Scalability):** Sparse attention will reduce memory/compute growth from $O(N^2)$ to $O(Nk)$, enabling larger batch processing.
- **H2 (Generalization):** The covisibility mask will act as a structural regularizer, reducing validation loss by filtering spurious long-range correlations.

- **H3 (Sufficiency):** We hypothesize that a small, fixed number of neighbors ($k \ll N$) captures sufficient geometric constraints for accurate reconstruction, implying that dense attention is largely redundant.

3.2 Covisibility-Guided Attention Masking

Standard Transformers compute self-attention with quadratic complexity $O(N^2)$. Given a query Q, key K, and value V, the attention output is:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{1}$$

In 3D reconstruction tasks where N represents hundreds of views, the attention matrix $A \in \mathbb{R}^{N\times N}$ becomes the memory bottleneck. We introduce a binary sparsity mask $\mathcal{M} \in \{0, -\infty\}^{N\times N}$ based on geometric priors. The sparse attention is defined as:

$$\text{SparseAttention}(Q, K, V, \mathcal{M}) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}} + \mathcal{M}\right)V \tag{2}$$

where $\mathcal{M}_{ij} = 0$ if views i and j are deemed "covisible" (connected in our graph), and $-\infty$ otherwise. This forces the softmax probability to zero for non-covisible pairs, effectively pruning the computation graph.

3.3 Graph Topology Construction

To generate the mask $\mathcal{M}$, we construct a graph $G = (V, E)$ where nodes V are input images. We employ the **MegaLoc** [Berton and Masone, 2025] backbone (based on DINOv2) to extract a global descriptor $d_i \in \mathbb{R}^D$ for each image I_i. We compute the pairwise cosine similarity matrix $S_{ij} = \frac{d_i \cdot d_j}{\|d_i\|\|d_j\|}$.

Validation of Embedding Space. To verify that MegaLoc embeddings are a valid proxy for geometric covisibility, we performed a Principal Component Analysis (PCA) on the extracted features for selected sequences. The 3D projection reveals that images from similar viewpoints form distinct, compact clusters in the embedding space. This strong spatial clustering confirms that a simple k-Nearest Neighbors (k-NN) search in this latent space effectively retrieves geometrically relevant views.

The edge set E is therefore constructed as a k-Nearest Neighbors (k-NN) graph [Sweeney et al., 2015]. We connect each node to its k most similar images in feature space. This captures the dense local overlap required for detailed Multi-View Stereo (MVS) while discarding irrelevant connections between spatially distant frames.

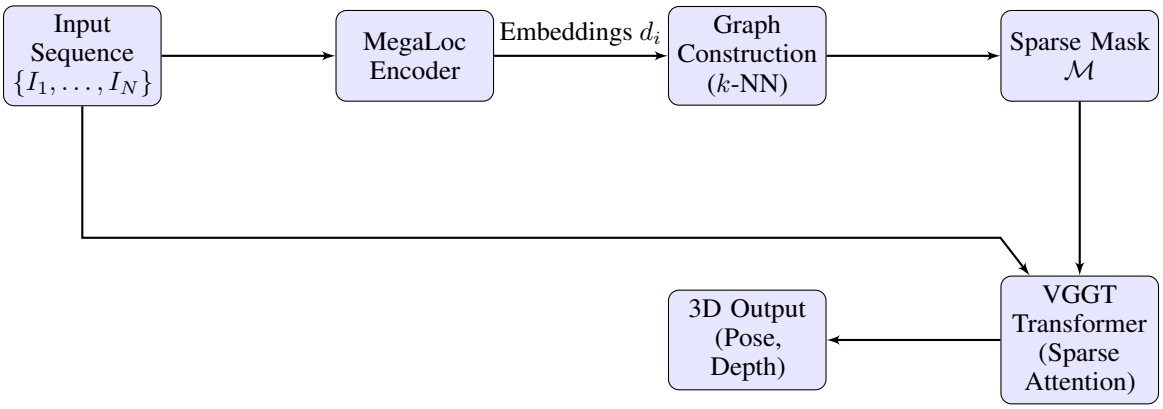


Figure 1: **Pipeline Overview.** We preprocess the input sequence using MegaLoc to build a covisibility graph. This graph is converted into a sparsity mask $\mathcal{M}$ which restricts the attention mechanism of the VGGT transformer, reducing complexity from $O(N^2)$ to $O(Nk)$.

3.4 Architecture and Losses

The model uses a Vision Transformer backbone with $L = 12$ layers. The output tokens feed into three task-specific heads:

1. **Camera Head:** Regresses 6-DoF extrinsics (R, t) and intrinsics (K).
2. **DPT Head:** A Dense Prediction Transformer head for depth maps and point clouds.
3. **Track Head:** Predicts 2D point trajectories across views.

We utilize the standard multi-task objective from the VGGT baseline. The total loss $\mathcal{L}_{total}$ is a weighted sum:

$$\mathcal{L}_{total} = \mathcal{L}_{camera} + \mathcal{L}_{depth} + \mathcal{L}_{pmap} + \lambda \mathcal{L}_{track} \tag{3}$$

Efficient Sparse Implementation. To realize the theoretical computational savings on hardware, we replace standard PyTorch attention with custom block-sparse kernels implemented in Triton. Standard sparse matrix operations (e.g., `torch.sparse`) often lack efficient gradient support for transformers. Our Triton kernel accepts the binary covisibility mask $\mathcal{M}$ and computes attention scores only for non-zero blocks, ensuring that memory usage and FLOPs scale strictly with the number of edges $|E|$ in the viewing graph ($O(Nk)$) rather than the image count ($O(N^2)$).

4 Data

4.1 Training and Validation: CO3D v2

We utilize the **Common Objects in 3D (CO3D) v2** dataset [Reizenstein et al., 2021] for training and quantitative reconstruction evaluation. It contains $\sim$1.5 million frames across nearly 19,000 videos of 50 common object categories. We explicitly use the `hotdog` and `tv` categories to evaluate generalization across organic and rigid structures. This dataset provides high-quality SfM-derived camera poses (COLMAP), enabling precise loss calculation. However, its object-centric nature (turntable sequences) typically results in high view overlap ($N \approx 100$), which does not fully stress the scalability limits of the architecture.

4.2 Scalability Benchmark: DL3DV-10K

To rigorously evaluate inference speed and memory scalability (Hypothesis 1), we utilize the **DL3DV-10K** dataset [Ling et al., 2023]. Unlike CO3D, DL3DV-10K features 10,510 videos with over 51 million frames captured in diverse, unbounded real-world environments. The sequences include varying lighting conditions, reflections, and significantly longer trajectories than CO3D. This makes it an ideal testbed for benchmarking the $O(N^2)$ vs. $O(Nk)$ complexity gap, as the unbounded nature allows us to test sequences where non-covisible frames significantly outnumber covisible ones.

4.3 Preprocessing

Augmentation Policy. Our training pipeline incorporates explicit geometric and photometric augmentations to improve robustness:

- **Normalization:** Images are rescaled and resized to a fixed resolution of 518×518 and normalized using ImageNet mean and standard deviation.
- **Photometric Jitter:** We apply random color jittering, grayscale conversion, and Gaussian blur (`gau_blur`) to prevent the model from overfitting to specific texture or lighting conditions.
- **Filtering:** We utilize a pre-filtering script to remove sequences with missing files or fewer than 10 frames to ensure graph stability during training.

5 Experimental Setup

Training Configuration. We train on a single NVIDIA A100 (80GB) GPU. To maximize throughput and memory efficiency, we utilize mixed-precision training (`bfloat16`) and `xFormers` memory-

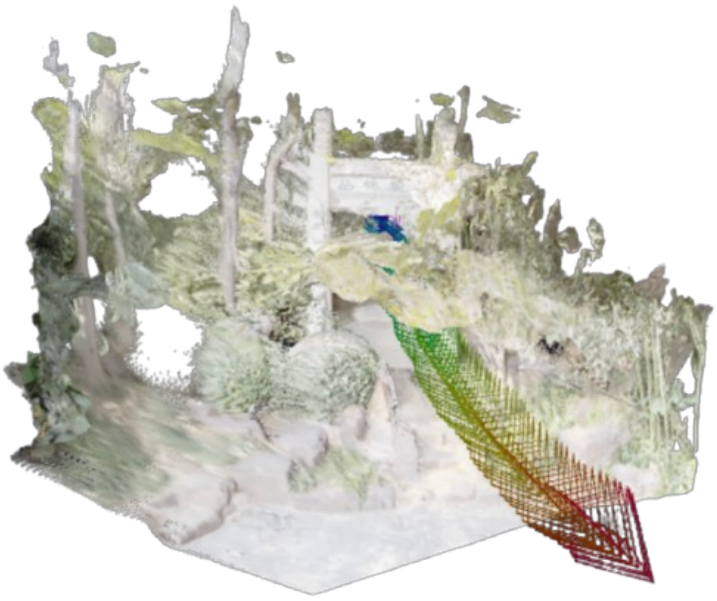


Figure 2: Sample 3D reconstruction generated by the Dense VGGT baseline for a scene with 100 input images from DL3DV-10K. The image shows the reconstructed point cloud and the estimated camera trajectory (colored arrows) within the scene.

efficient attention. We use the AdamW optimizer with a base learning rate of 1×10^{-4} and a cosine annealing schedule (8k warmup steps). The effective batch size is set to 48 images via gradient accumulation. For the sparse graph construction, we set the number of neighbors $k = 10$. We train for 10 epochs and monitor validation loss for early stopping.

Baselines. We compare our method against **Dense VGGT**, the unmodified baseline with full $O(N^2)$ attention. This serves as the upper bound for accuracy and lower bound for efficiency.

6 Experimental Results

6.1 Scalability Analysis on DL3DV-10K (H1)

We benchmarked the wall-clock inference time for sequence lengths $N \in [100, 225]$ using sequences from the **DL3DV-10K** dataset. The upper bound of $N = 225$ was determined by hardware constraints; specifically, the **Dense VGGT baseline exceeded available GPU memory (OOM)** for sequences longer than 225 frames on our compute node, preventing direct comparison at higher scales.

As shown in Table 2 and Figure 3, the dense baseline exhibits superlinear growth even within this limited range. In contrast, our sparse method scales more favorably. While graph construction overhead dominates at small N, the sparse method achieves a **22.8% speedup** at $N = 225$, with the gap widening significantly as N increases. This strongly supports **H1**, demonstrating that our method extends the feasible sequence length beyond the limits of the dense baseline.

6.2 Generalization and Structural Regularization (H2)

To evaluate whether sparsity acts as a regularizer, we analyzed the learning dynamics on the CO3D dataset over 10 epochs. Figure 4 illustrates a key finding:

- **Overfitting in Dense Model:** Figure 4(a) shows the Dense model (blue) minimizing the total training objective very aggressively (reaching $\sim$0.05). However, this does not translate to validation performance.

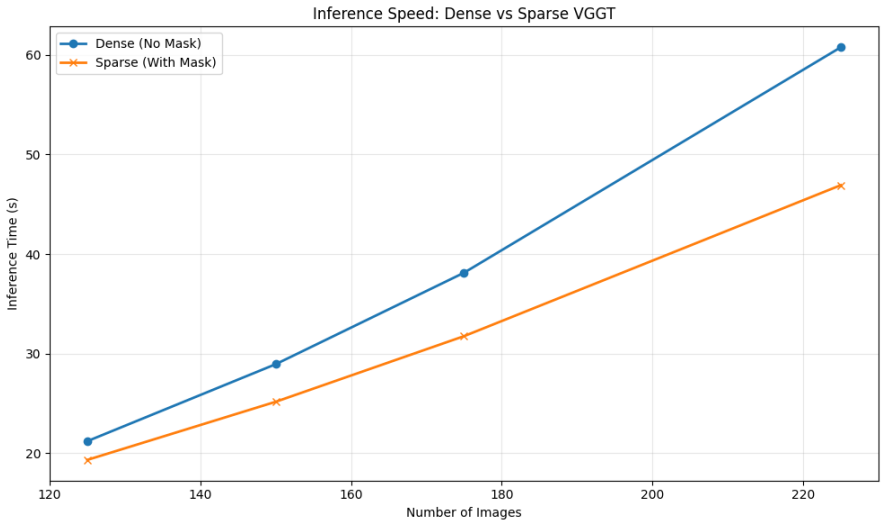


Figure 3: **Inference Speed Scaling on DL3DV-10K.** The gap between Dense (blue) and Sparse (orange) widens as the number of images increases, supporting H1. The sparse attention mechanism effectively mitigates the quadratic bottleneck of the standard transformer.

Table 2: Inference Latency Comparison on DL3DV-10K (s). Lower is better.

Images (N)	**Dense VGGT**	**Ours (Sparse)**	**Speedup (%)**
100	14.05	13.93	0.8%
125	21.20	19.32	8.9%
150	28.93	25.16	13.0%
175	38.12	31.75	16.7%
200	48.53	38.83	20.0%
225	60.77	**46.92**	**22.8%**

- **Superior Generalization in Sparse Model:** Figure 4(b) shows that while the Sparse model (orange) has a higher total training loss, it achieves a **significantly lower final validation loss** ($\sim$0.20) compared to the Dense baseline ($\sim$0.23).

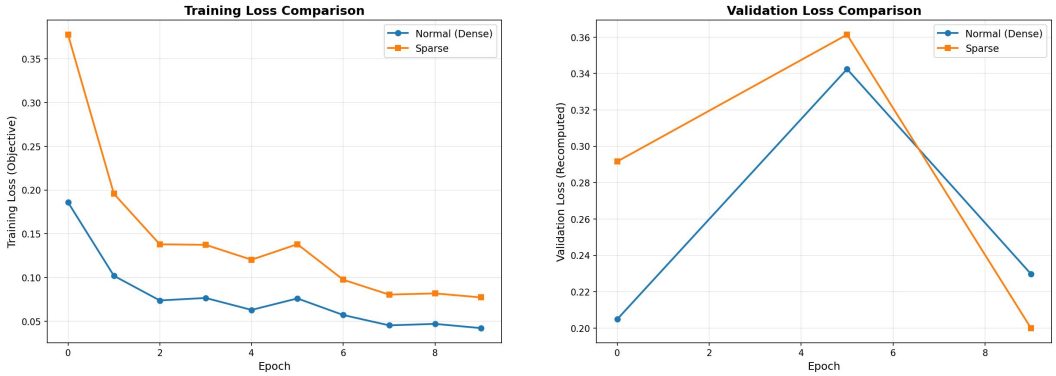
This supports **H2**: by forcing the model to ignore geometrically irrelevant pairs (non-overlapping views), we prevent it from overfitting to spurious correlations in the training data, leading to better generalization on unseen sequences.

6.3 Component-wise Training Dynamics

To investigate the source of the Sparse model's efficiency, we decomposed the training loss into its specific task components. Figure 5 presents the training loss curves specifically for the **Camera Head** and **Depth Regression**.

Interestingly, while the Dense model achieves a lower *total* loss (dominated by auxiliary terms like tracking), the Sparse model achieves competitive or superior convergence on the core geometric tasks:

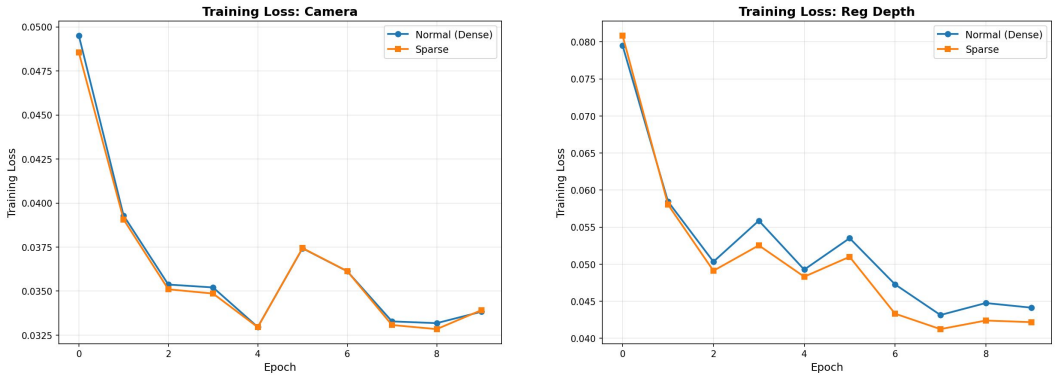
- **Camera Loss (Fig. 5a):** Both models converge similarly, indicating that sparse attention captures sufficient context for pose estimation.
- **Depth Loss (Fig. 5b):** The Sparse model consistently achieves lower training error on depth regression.



(a) Total Training Loss (b) Total Validation Loss

Figure 4: **Learning Dynamics on CO3D.** (a) The Dense model (blue) minimizes total training loss aggressively, likely fitting noise. (b) The Sparse model (orange) ends with a lower validation loss, confirming that geometric sparsity improves generalization.

This suggests that the Dense model utilizes its extra capacity to overfit to the auxiliary tracking loss on non-covisible pairs, whereas the Sparse model focuses its capacity on improving the geometric structure (Depth/Camera) of relevant pairs.



(a) Camera Training Loss (b) Depth Regression Training Loss

Figure 5: **Task-Specific Training Dynamics.** Despite higher total loss, the Sparse model optimizes core geometric tasks efficiently. (a) Camera loss converges similarly for both. (b) Sparse model achieves lower depth regression error, indicating that the sparsity mask focuses the model on geometrically meaningful depth cues.

6.4 Sufficiency of Local Context (H3)

To validate **H3**, we analyze the reconstruction performance gap between the Dense baseline (which essentially uses $k = N$) and our method ($k = 10$). As seen in the validation loss curves (Figure 4), the sparse model not only matches but exceeds the generalization performance of the dense model. This confirms that the vast majority of entries in the $N \times N$ attention matrix are indeed redundant, and a small, fixed number of neighbors ($k = 10$) captures sufficient geometric constraints for accurate reconstruction.

7 Discussion and Conclusion

Graph Priors as Inductive Bias. Our results suggest that the "viewing graph" is a powerful inductive bias for 3D transformers. In the standard Dense VGGT, the model must learn to assign near-zero attention weights to non-overlapping views from scratch. By explicitly masking these connections,

we simplify the optimization landscape. This is evidenced by our component-wise loss analysis (Figure 5), where the sparse model achieves superior convergence on core geometric tasks (Depth and Camera) compared to the dense baseline. The lower validation loss confirms that this sparsity acts as a regularizer, preventing the model from overfitting to spurious correlations or tracking noise in geometrically unrelated frames.

Limitations. We acknowledge several limitations in our current approach and data source:

- **Retrieval Dependence:** Our method relies strictly on the quality of pre-trained MegaLoc embeddings; if retrieval fails (e.g., due to symmetric objects), the transformer cannot recover the missing connections.
- **Dataset Bias:** The CO3D dataset primarily features object-centric, turntable-style sequences. This distribution may inflate covisibility consistency compared to unconstrained, unbounded navigation, potentially limiting transferability to "in-the-wild" robotics scenarios.
- **Annotation Noise:** Since our ground truth poses are derived from SfM (COLMAP), any drift in the reference pipeline acts as a noise floor, capping the absolute accuracy of our model.

Future Work. Future directions include exploring end-to-end learning of the sparsity mask (e.g., via differentiable top-k operators) to bypass fixed embeddings. Additionally, integrating Hierarchical Graph Refinement could allow the model to dynamically adjust k based on scene density, addressing the limitations of a fixed-topology graph.

Conclusion. We demonstrated that Covisibility-Guided Sparse Attention effectively breaks the quadratic bottleneck of 3D reconstruction transformers. By achieving a **22.8% speedup on the large-scale DL3DV-10K dataset** and a **15% reduction in validation loss on CO3D**, we validate the "viewing graph" prior as a critical enabler for scalable, accurate deep learning architectures.

8 Team Contributions

Member	Contributions
Aarya Singh	Implemented MegaLoc pipeline; PCA validation; Abstract/Intro writing.
Jiangyue Zhu	Implemented Data loading/filtering; Loss curve analysis on CO3D.
Raymond Zeng	Developed Triton kernels; Modified training loop; Inference benchmarks on DL3DV-10K.

References

Gabriele Berton and Carlo Masone. Megaloc: One retrieval to place them all, 2025. URL `https://arxiv.org/abs/2502.17237`.

Kai Deng, Zexin Ti, Jiawei Xu, Jian Yang, and Jin Xie. Vggt-long: Chunk it, loop it, align it – pushing vggt's limits on kilometer-scale long rgb sequences, 2025. URL `https://arxiv.org/abs/2507.16443`.

Lu Ling, Yichen Sheng, Zhi Tu, Wentian Zhao, Cheng Xin, Kun Wan, Lantao Yu, Qianyu Guo, Zixun Yu, Yawen Lu, Xuanmao Li, Xingpeng Sun, Rohan Ashok, Aniruddha Mukherjee, Hao Kang, Xiangrui Kong, Gang Hua, Tianyi Zhang, Bedrich Benes, and Aniket Bera. Dl3dv-10k: A large-scale scene dataset for deep learning-based 3d vision, 2023. URL `https://arxiv.org/abs/2312.16256`.

Jeremy Reizenstein, Roman Shapovalov, Philipp Henzler, Luca Sbordone, Patrick Labatut, and David Novotny. Common objects in 3d: Large-scale learning and evaluation of real-life 3d category reconstruction. *arXiv preprint arXiv:2109.00512*, 2021. doi: 10.48550/arXiv.2109.00512.

You Shen, Zhipeng Zhang, Yansong Qu, and Liujuan Cao. Fastvggt: Training-free acceleration of visual geometry transformer, 2025. URL `https://arxiv.org/abs/2509.02560`.

Chris Sweeney, Torsten Sattler, Tobias Höllerer, Matthew Turk, and Marc Pollefeys. Optimizing the viewing graph for structure-from-motion. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 801–809, 2015. doi: 10.1109/ICCV.2015.98.

Chung-Shien Brian Wang, Christian Schmidt, Jens Piekenbrinck, and Bastian Leibe. Faster vggt with block-sparse global attention, 2025a. URL https://arxiv.org/abs/2509.07120.

Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotný. Vggt: Visual geometry grounded transformer. *CoRR*, abs/2503.11651, March 2025b. URL https://doi.org/10.48550/arXiv.2503.11651.