Speed3R: Sparse Feed-forward 3D Reconstruction Models

Weining Ren[1] Xiao Tan[2] Kai Han[1,*]
[1]The University of Hong Kong [2]Baidu Inc.
https://visual-ai.github.io/speed3r/

Abstract

While recent feed-forward 3D reconstruction models accelerate 3D reconstruction by jointly inferring dense geometry and camera poses within a single forward pass, their reliance on dense attention imposes quadratic complexity, creating a computational bottleneck that limits inference speed. To resolve this, we introduce Speed3R, an end-to-end trainable model inspired by the core principle of Structure-from-Motion: that a sparse set of keypoints is sufficient for robust pose estimation. Speed3R features a dual-branch attention mechanism in which a compression branch generates a coarse contextual prior to guide a selection branch, which applies fine-grained attention only to more informative image tokens. This strategy mimics the efficiency of traditional keypoint matching, achieving a remarkable 12.4x inference speed-up on 1000-view sequences, while introducing a minimal, controlled trade-off in geometric accuracy. Validated on standard benchmarks with both VGGT and π^3 backbones, our method delivers high-quality reconstructions at a fraction of computational cost, paving the way for efficient large-scale scene modeling.

1. Introduction

Classical 3D reconstruction methods are fundamentally rooted in the principle of sparsity. They traditionally begin with the detection and matching of a sparse set of salient keypoints. Early methods relied on handcrafted descriptors [2, 10, 21, 27, 28] to establish 2D correspondences across views. Initial geometric relationships are estimated from these matches, followed by a large-scale, iterative optimization Bundle Adjustment [13], which refines both the camera parameters and the sparse 3D structure. This multi-stage process, exemplified by highly successful and robust systems [24, 29, 30], demonstrates a core insight: a sparse, carefully selected set of 2D features is sufficient to establish robust geometric constraints and reconstruct an accurate sparse point cloud.

*Corresponding author.

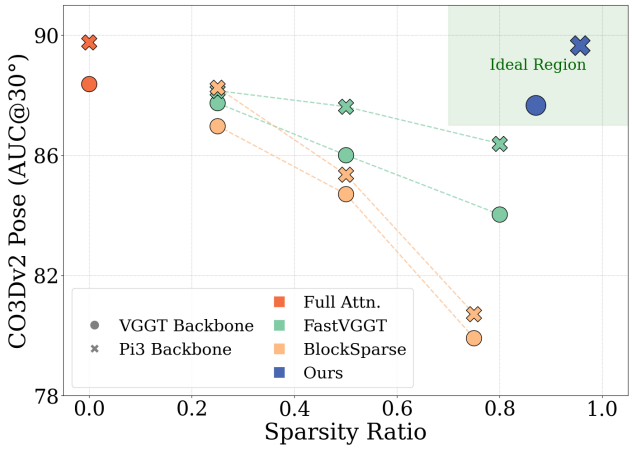


Figure 1. **Performance comparison of different methods**. CO3Dv2 pose estimation accuracy vs. sparsity ratio, highlighting the trade-off between sparsity and accuracy. The Ideal Region marks the desired balance of high accuracy and high sparsity.

Recently, the advent of feed-forward models [39, 42, 43, 47] has revolutionized 3D reconstruction, enabling the joint inference of dense geometry and camera poses from multiple views in a single network pass. These methods, often built upon powerful end-to-end vision transformer architectures [11, 23], bypass the complex, multi-stage pipelines of classical approaches like Structure-from-Motion (SfM) and Multi-View Stereo (MVS). However, this progress comes at a price. The dense global attention along all image tokens imposes a quadratic complexity with respect to the number of input image tokens. This creates a prohibitive computational bottleneck that severely limits inference speed, making the processing of a large number of views or high-resolution images intractable.

To relieve the computational bottleneck, we propose Speed3R, an end-to-end trainable model that accelerates inference while retaining the benefits of feed-forward reconstruction. Our approach integrates two key inspirations: the classical Structure-from-Motion (SfM) principle that robust geometric estimation relies on sparse keypoints rather than dense pixel comparisons, and the success of sparse atten-

tion in Large Language Models [12, 22, 50] and Video Diffusion Models [6, 52]. Speed3R operationalizes these insights through a dual-branch attention mechanism. A compression branch generates a global scene summary, guiding a selection branch that performs fine-grained attention on a small subset of informative tokens. This design emulates traditional keypoint-based methods, concentrating computation where it is most impactful, and achieves significant efficiency gains without sacrificing accuracy.

Through this tailored sparse attention mechanism, Speed3R achieves a remarkable 12.4x inference speedup on 1000-view sequences. Our extensive experiments demonstrate that this substantial acceleration is achieved while introducing only a minimal and controlled trade-off in geometric accuracy, establishing a new Pareto-optimal frontier in the efficiency-fidelity landscape. We validate the generalizability of our method by integrating it with state-of-the-art backbones, including VGGT [39] and π^3 [43], and demonstrate consistently superior performance on standard benchmarks than training-free methods. With zero-shot test-time adaptation, our method can outperform dense models on long sequences. By rethinking the attention mechanism for the global attention layer, Speed3R delivers high-quality reconstructions at a fraction of the computational cost, paving the way for efficient large-scale scene modelling. Overall, we make the following contributions:

- We propose Speed3R, a novel dual-branch feed-forward reconstruction model with a trainable sparse attention mechanism that mimics classical SfM by focusing computation on a small, informative subset of tokens.
- We achieve a new SoTA in the efficiency-accuracy trade-off, demonstrating a 12.4x speedup for a 1000-view sequence with minimal impact on geometric accuracy.
- We validate the generalizability of Speed3R, showing that it integrates with various backbones and outperforms competing training-free methods.

2. Related Work

Optimization-based Multi-view Reconstruction. Traditional 3D reconstruction methods operate on a sparse-to-dense paradigm. The process begins with Structure-from-Motion (SfM) [24, 29], where a sparse set of matched keypoints [2, 21, 27] is used to optimize camera poses and a sparse point cloud via bundle adjustment [13]. This sparse geometric backbone is then densified using Multi-View Stereo (MVS) algorithms [30]. While deep learning has modernized this pipeline with learned feature matchers [18, 28], learned MVS cost volumes [48, 53], and differentiable optimization [34, 38], the core methodology remains dependent on an initial sparse representation and iterative optimization, making it inherently slow and computationally demanding.

Feed-forward 3D Reconstruction. Recent end-to-end 3D reconstruction methods facilitate the simultaneous estimation of camera poses and dense scene geometry within a single neural network forward pass. This paradigm, pioneered by DUSt3R [42] for pairwise input, was quickly refined with dedicated feature heads for improved matching (MASt3R [17]) and extended to handle sequential inputs (CUT3R [40], Spann3R [37]). Architectural innovations, such as the elegant and effective design of VGGT [39] and the permutation-equivariant structure of π^3 [43], have pushed the state of the art. However, the reliance of these advanced models on dense, all-to-all attention for global information exchange creates a significant computational bottleneck, especially for long sequences. To mitigate this, several training-free sparsification approaches have been proposed. For instance, FastVGGT [32] employs a token merge-and-unmerge strategy, while Block Sparse VGGT [36] applies top-k attention. Because these methods are not training-aware, their ability to sparsify the model is limited; aggressive pruning results in a notable degradation of reconstruction accuracy.

Sparse Attention. To mitigate the quadratic complexity of standard attention, various sparse attention methods have been proposed. Early approaches employed fixed, data-agnostic patterns like local windows (StreamingLLM [46]) or dilated windows (Longformer [3]). More advanced methods are dynamic, such as those that prune the KV cache during inference (H2O [54]), although these post-hoc optimizations do not accelerate the training phase. Another class of dynamic methods performs query-aware token selection using techniques like clustering [20] or heuristic scoring [33]; however, these methods often suffer from non-differentiable operations or non-contiguous memory access, hindering end-to-end training. To resolve these issues, recent works NSA [50] and MOBA [22] present breakthroughs with trainable sparse attention, achieving results comparable to full attention. The success of these methods has led to their application in various domains, including video generation [6, 52] and 3D generation [44]. The philosophy of adaptively selecting important tokens is particularly well-suited to the sparsity inherent in 3D reconstruction. Highly inspired by NSA, our work aims to develop an efficient sparse attention mechanism tailored for 3D feed-forward reconstruction.

3. Method

Our primary goal is to develop a feed-forward 3D reconstruction model capable of efficiently processing a large number of views. To address the scalability bottleneck, we introduce a novel attention mechanism, **Global Sparse Attention (GSA)**, designed as a drop-in replacement for the original global attention module.

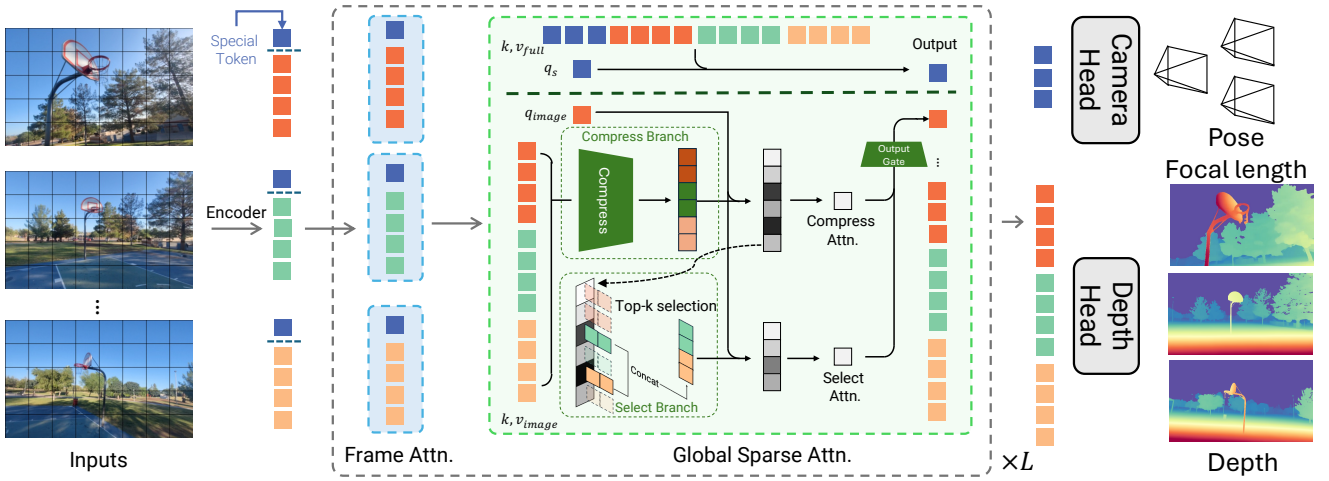


Figure 2. **Architecture.** Our model processes a sequence of input images through a shared feature encoder. The resulting tokens are then processed by a series of transformer blocks that alternate between local **Frame Attention** (within each view) and our proposed **Global Sparse Attention** (across all views). The GSA module efficiently integrates global information by decomposing attention into a **Compression Branch** for coarse context and a **Selection Branch** for fine-grained details, guided by a Top-k selection mechanism. Finally, updated tokens are passed to task-specific heads to predict camera pose and dense depth maps.

3.1. Architecture Overview

Our approach builds upon the architecture of recent feed-forward reconstruction models, as depicted in Figure 2. The model processes a sequence of input images to jointly predict camera parameters (pose and focal length) and dense depth maps. The architecture consists of three main stages:

- **Per-frame Feature Encoder:** A sequence of N images $\{I_i\}_{i=1}^N$ are independently passed through vision encoder (e.g., DINOv2 [23]) to extract patch-based feature tokens.
- **Alternating Attention Transformer:** A series of transformer blocks processes the tokens. These blocks alternate between local Frame Attention, which operates on tokens within a single frame, and our proposed Global Sparse Attention, which efficiently aggregates information across all frames.
- **Task-Specific Prediction Heads:** The refined tokens are used by downstream heads to predict per-view camera parameters $\{\hat{C}_i\}_{i=1}^N$, depth maps $\{\hat{D}_i\}_{i=1}^N$ and their associated uncertainty $\{\hat{\alpha}_i\}_{i=1}^N$.

Our key innovation lies in the second stage, where we replace the computationally intensive global full-attention layer with our GSA module. For all other components, including the feature encoder, frame attention, and prediction heads, we follow the design of the original base models (VGGT [39] or π^3 [43]).

3.2. Global Sparse Attention (GSA)

The GSA module is designed to approximate full attention with significantly reduced computational complexity, making it scalable to a large number of views. Its core principle is to leverage a coarse, low-resolution representation of the scene, computed by the **Compression Branch**, to guide the selection of a sparse subset of high-resolution tokens for the **Selection Branch**. This coarse-to-fine strategy enables the model to efficiently build a global understanding of the scene while focusing its limited computational budget on the most salient, keypoint region details.

Let the input to the GSA block be tokens $X \in \mathbb{R}^{M\times C}$, where M is the total number of tokens and C is the channel dimension. The sequence X is a concatenation of special tokens $X_{\text{spec}} \in \mathbb{R}^{M_{\text{spec}}\times C}$ and image tokens $X_{\text{img}} \in \mathbb{R}^{M_{\text{img}}\times C}$, such that $M = M_{\text{spec}} + M_{\text{img}}$. For brevity, the batch dimension is omitted. We first project the sequence into queries, keys, and values with dimension d using linear transformations W_Q, W_K, W_V. The resulting tensors are then partitioned corresponding to their original token types:

$$Q = \begin{bmatrix} Q_{\text{spec}} \\ Q_{\text{img}} \end{bmatrix}, \quad K = \begin{bmatrix} K_{\text{spec}} \\ K_{\text{img}} \end{bmatrix}, \quad V = \begin{bmatrix} V_{\text{spec}} \\ V_{\text{img}} \end{bmatrix} \tag{1}$$

The attention computation then proceeds differently for the two token types.

Full Attention for Special Tokens. Special tokens serve as global information bottlenecks and are critical for tasks like pose estimation. To ensure they have a comprehensive view of the entire scene, they perform standard, dense self-attention over all other tokens. The output for these tokens, O_{spec}, is computed as:

$$O_{\text{spec}} = \text{Attention}(Q_{\text{spec}}, K, V) = \text{softmax}\left(\frac{Q_{\text{spec}}K^T}{\sqrt{d_k}}\right)V \tag{2}$$

This operation, while quadratic, is inexpensive as the number of special tokens (M_{spec}) is very small.

Sparse Attention for Image Tokens. For the vast number of image tokens, we employ a two-branch strategy that computes a coarse global summary and fine-grained local details in sequence. The outputs from these branches are then dynamically fused using a learned gating mechanism, allowing the model to adaptively balance global context against local specificity for each token.

Compression Branch. This branch provides a coarse but comprehensive summary of the scene in a highly efficient manner. To create a computationally inexpensive proxy for global attention, we spatially downsample QKV tensors (Q_{img}, K_{img}, V_{img}). This is achieved using a non-overlapping average pooling operation with a window size of $s \times s$, yielding compressed tensors Q_{comp}, K_{comp}, and V_{comp}, all of size $\mathbb{R}^{M'_{\text{img}} \times d}$, where $M'_{\text{img}} = M_{\text{img}}/s^2$.

The attention calculation of this branch is then performed entirely within this compressed space, producing a coarse output tensor O'_{comp}. Additionally, to guide the subsequent selection process, a score matrix S_{guide} is computed from the compressed queries and keys. The coarse output O'_{comp} is subsequently upsampled to the original image token resolution using nearest-neighbor interpolation, which assigns the same context vector to all fine-grained tokens that belong to the same spatial window. This yields the final branch output, O_{comp}.

$$O'_{\text{comp}} = \text{Attention}(Q_{\text{comp}}, K_{\text{comp}}, V_{\text{comp}}) \in \mathbb{R}^{M'_{\text{img}} \times d} \quad (3)$$

$$S_{\text{guide}} = Q_{\text{comp}} K^T_{\text{comp}} \in \mathbb{R}^{M'_{\text{img}} \times M'_{\text{img}}} \quad (4)$$

$$O_{\text{comp}} = \text{Upsample}(O'_{\text{comp}}) \in \mathbb{R}^{M_{\text{img}} \times d} \quad (5)$$

Selection Branch. To recover fine-grained attention, this branch performs attention on a small subset of the original, full-resolution key-value pairs. To guide the selection, we use the pre-computed relevance score matrix S_{guide} to identify the most relevant coarse regions for each query. For each query, we use a TopKSelect$(\cdot)$ function on S_{guide} to identify the indices of the most relevant *regions*. Queries belonging to the same compression window share the same set of KV pairs. The original, full-resolution key-value pairs $(K_{\text{img}}, V_{\text{img}})$ corresponding to these top regions are then selected, forming the sparse sets K_{sel} and V_{sel}. The fine-grained output O_{sel} is computed by attending only to this small subset:

$$O_{\text{sel}} = \text{Attention}(Q_{\text{img}}, K_{\text{sel}}, V_{\text{sel}}) \quad (6)$$

This operation is highly efficient as each query only attends to $k \ll M_{\text{img}}$ tokens.

Gated Aggregation. The outputs from the two branches are combined using a learnable gating mechanism that weights their contributions based on the query itself. A gating vector $g \in \mathbb{R}^{M_{\text{img}} \times d}$ is computed from the image queries, and the final output for the image tokens, O_{img}, is a dynamic, weighted sum:

$$g = \sigma(W_g Q_{\text{img}}) \quad (7)$$

$$O_{\text{img}} = g \odot O_{\text{comp}} + (1 - g) \odot O_{\text{sel}} \quad (8)$$

where σ is the sigmoid function, W_g is a learned projection matrix, and $\odot$ denotes element-wise multiplication. This allows the model to decide for each token whether to rely more on the global summary from the Compression Branch or the specific details from the Selection Branch.

Final Output. The final output of the GSA layer, O_{GSA}, is produced by concatenating the outputs from the two pathways in their original order:

$$O_{\text{GSA}} = \text{concat}(O_{\text{spec}}, O_{\text{img}}) \in \mathbb{R}^{M \times d} \quad (9)$$

Efficient Kernel Implementation. A naive implementation of the Compression Branch with TopKSelect$(\cdot)$ is inefficient, bottlenecked by the large memory footprint of the full score matrix S_{guide}. To overcome this, we developed a fused GSA kernel in Triton [35]. Our approach integrates a streaming top-k algorithm directly into the FlashAttention [8] workflow. As our kernel computes score matrix tiles in fast on-chip SRAM, it not only performs the online softmax but simultaneously maintains a running set of the top-k indices and scores for each query. This allows the selection of the most relevant keys and the calculation of compression output to occur in a single, fused pass over the input data. By doing so, we avoid materializing the full score matrix and maximize data locality. Implementation details are provided in the supplementary material.

3.3. Speed3R-VGGT

We instantiate our method on the VGGT architecture, terming this variant **Speed3R-VGGT**. The VGGT model architecture is unique in that it designates the first frame of a sequence as a global reference and utilizes dedicated camera tokens to encode pose information.

To ensure this critical global reference is not lost by the sparse attention mechanism, we adapt the GSA's Selection Branch. For any given query, its attention set $(K_{\text{sel}}, V_{\text{sel}})$ is constructed from the **concatenation** of two groups: (i) A fixed global context set comprising all tokens from the reference frame and frames sampled at 100-frame intervals. (ii) The dynamically selected Top-K image tokens windows from non-reference frames, identified by our standard selection process. This hybrid approach guarantees that while

the model can efficiently focus on salient local details in subsequent frames, it never loses sight of the foundational reference frame and camera parameters.

To transfer the performance of the original dense model to our efficient sparse variant, we employ a knowledge distillation strategy. The student model (Speed3R-VGGT) is trained to replicate the outputs of its pre-trained dense teacher. The teacher's predictions for depth and camera pose serve as the pseudo ground truth for the student. The total training loss is a weighted sum of a depth distillation loss and a camera pose distillation loss:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{depth}} + \lambda \mathcal{L}_{\text{camera}} \tag{10}$$

where the losses are inherited from original VGGT losses.

3.4. Speed3R-π^3

Similarly, we apply our method to the π^3 [43] architecture to create Speed3R-$\boldsymbol{\pi^3}$. Unlike VGGT [39], π^3 [43] does not rely on reference frame or camera tokens, which allows for a more direct application of our GSA module as described in Section 3.2. Additionally, we empirically observe that the register tokens used in the original π^3 can be omitted in our sparse variant without performance drop, further simplifying the model without special tokens. We follow the same distillation strategy, using the original dense π^3 model as the teacher to train our Speed3R-π^3 student with relative pose loss and depth loss. The total loss function is identical in form to that used for original π^3.

3.5. Training Details

We train our model on a mixture of seven datasets: ArkitScene [1], Scannet++ [49], DL3DV [19], CO3D [25], Hypersim [26], WildRGBD [45], and VirtualKitti2 [4]. Some datasets were downsampled for storage efficiency (see supplementary material), and the sparse model's weights were initialized from its pre-trained dense counterpart. The model was trained for 80 epochs (800 steps per epoch) over approximately 7 days on 8 NVIDIA H20 GPUs. We followed the original dense model's setting, but with two adjustments: the learning rate was initialized to 1×10^{-5}, and a gradient accumulation factor of 4 was used to achieve an effective batch size of 32.

4. Experiments

We evaluate our sparse models against their dense counterparts and two training-free baselines: FastVGGT [32] and Block-Sparse VGGT [36]. Variants of these baselines with VGGT/π^3 are referred to as FastVGGT-VGGT/π^3 and Block-Sparse VGGT/π^3, respectively. Unless stated otherwise, we use the following parameters. Our method employs a 4x4 compression window and selects the top-32 blocks for selective attention. For the baselines, we adopt their default configurations: a 0.9 merge ratio for FastVGGT [32] and a 0.75 sparsity ratio for Block-Sparse VGGT/π^3 [36]. All inference times are benchmarked on a single H100 GPU.

Table 1. **Pair-wise pose results on ScanNet-1500 [7, 28].** We report the Area Under the Curve (AUC) of the pose error at different thresholds. Best results per backbone are marked in **bold**.

Methods	ScanNet1500		
	AUC@5 ↑	AUC@10 ↑	AUC@20 ↑
VGGT [39]	37.45	59.24	75.69
Block Sparse-VGGT [36]	33.21	55.11	72.51
FastVGGT-VGGT [32]	33.59	56.21	73.47
Speed3R-VGGT	**37.02**	**59.11**	**75.62**
π^3 [43]	**38.76**	**61.57**	**77.61**
Block Sparse-π^3 [36]	35.13	57.74	74.98
FastVGGT-π^3 [32]	34.87	58.31	75.51
Speed3R-π^3	**36.97**	**59.83**	**76.38**

4.1. Two-view Pose Estimation

We first evaluate our method on the ScanNet relative pose estimation task [7, 28], reporting the Area Under the Curve (AUC) of the pose error. This metric is the area under an accuracy-threshold curve where per-pair accuracy is defined as the minimum of the Relative Rotation Accuracy (RRA) and Relative Translation Accuracy (RTA) at different thresholds. The results in Table 1 show that our method (Speed3R) consistently outperforms competing training-free sparse methods. Given that this benchmark evaluates performance under large viewpoint changes, this result highlights our method's superior robustness to such variations. Notably, Speed3R-VGGT also achieves performance nearly on par with the original dense VGGT model, and this advantage holds for the more powerful π^3 backbone, confirming the general effectiveness of our approach.

4.2. Multi-view Pose Estimation

We further validate our approach on the multi-view RE10k [55] and CO3Dv2 [25] benchmarks, which include 10 frames per scene, with comparisons to SAIL-Recon [9]. As shown in Table 2 and visualized in Figure 1, Speed3R establishes a new Pareto-optimal frontier for accuracy and efficiency. It consistently outperforms all competing sparse and anchor-based methods, often at significantly higher sparsity ratios. This superiority is demonstrated by two key results: first, our Speed3R-VGGT model (84% sparsity) surpasses the dense VGGT baseline on RE10k; second, the Speed3R-π^3 model (94% sparsity) nearly matches the performance of its dense counterpart. These findings demonstrate that Speed3R effectively prunes redundant computations while preserving performance-critical information.

Table 2. **Pose Estimation on RE10k [55] and CO3Dv2 [25].** These datasets contain an average of 10 images per scene.

Method	Sparse Ratio (%)/ Anchor Number	RE10K AUC@30 ↑	CO3Dv2 AUC@30 ↑
VGGT [39]	0	74.17	88.33
Block Sparse-VGGT [36]	25	71.79	86.98
	50	68.25	84.71
	75	63.82	79.92
SAIL-Recon [9]	10 anchor	74.31	87.63
	5 anchor	72.66	84.25
	2 anchor	69.11	80.03
FastVGGT [32]	25	72.97	87.74
	50	71.55	86.01
	82	69.99	84.03
Speed3R-VGGT	84	74.81	87.71
π^3 [43]	0	87.37	89.67
Block Sparse-π^3 [36]	25	85.18	88.25
	50	81.29	85.36
	75	75.39	80.72
FastVGGT-π^3 [32]	25	87.26	88.15
	50	86.67	87.62
	90	86.04	86.39
Speed3R-π^3	94	87.17	89.41

Table 3. **Pose Estimation on Tanks & Temples [16].** This dataset contains an average of 300 images per scene. Best results and second best results are **in bold** and underlined separately.

Method	RRA@5 ↑	RTA@5 ↑	AUC@30 ↑	Time [s] ↓
VGGT [39]	70.29	79.30	77.67	34.51
Block Sparse-VGGT [36]	66.83	71.29	74.15	10.79
SAIL-Recon(20 anchor) [9]	68.34	73.77	74.98	20.35
SAIL-Recon(100 anchor) [9]	**69.72**	75.16	75.70	53.02
FastVGGT [32]	69.28	**77.98**	76.29	15.98
Speed3R-VGGT	69.51	77.81	**76.57**	**6.55**
π^3 [43]	72.14	81.26	79.63	22.32
Block Sparse-π^3 [36]	67.85	78.91	76.64	8.16
FastVGGT-π^3 [32]	69.78	79.51	77.76	11.96
Speed3R-π^3	**70.72**	**80.72**	**79.77**	**4.19**

4.3. Long-sequence Pose Estimation

We further evaluate on the large-scale Tanks & Temples (T&T) benchmark [16] with an average of 300 images per sequence. As shown in Table 3, Speed3R demonstrates a state-of-the-art balance of accuracy and speed. With the VGGT backbone, our method is by far the fastest (6.55s), achieving a 5.2x speedup over the dense baseline while maintaining top-tier accuracy, including the best AUC@30 score among all sparse methods. The same holds for the π^3 backbone. Speed3R-π^3 simultaneously achieves the highest accuracy across all metrics and the lowest runtime (4.19s) among all sparse methods. It nearly matches the performance of the dense π^3 model while being 5.3x faster. These results confirm that Speed3R excels at achieving Pareto-optimal results that are both highly accurate and efficient for large-scale feed-forward pose estimation.

4.4. Pointmap Estimation

Following π^3 [43], we compare our predicted point cloud with baseline methods. We report the mean and median of accuracy, completeness, and normal consistency. As shown in Table 4 and illustrated in Figure 3, our proposed Speed3R method demonstrates a superior trade-off between performance and efficiency for pointmap estimation. When compared against other sparse methods, Speed3R consistently achieves the best results across nearly all metrics on both the DTU and ETH3D datasets, establishing it as the state-of-the-art among efficiency-focused techniques. More importantly, while the dense baseline models (VGGT and π^3) exhibit slightly better accuracy, our Speed3R variants remain highly competitive, incurring only a marginal performance degradation. This indicates that the learned sparse attention patterns effectively preserve the most critical information for high-quality reconstruction.

4.5. Ablation Study

We first conduct ablation studies on the Speed3R-π^3, with a baseline using a 4x4 window and top-32. We train all models with 40 epochs and gradient accumulation factor of 2. As detailed in Table 5, we first analyze our GSA module. Removing the compression branch value (1) impairs long-sequence performance on the T&T [16] dataset by sacrificing global context, while removing the selection branch (2) is uniformly detrimental, causing a significant drop on both datasets. Adding a register token (3) has a negligible effect. In our hyperparameter analysis (4-7), we find that our choice (4x4 window with top-32 indices) strikes the balance between accuracy and efficiency. Finally, removing knowledge distillation (8) substantially degrades performance across both datasets. This result highlights its importance in mitigating the impact of noisy labels from the real-world dataset.

Due to the utilization of reference-frame and camera token of VGGT [39], the design of Speed3R-VGGT differs from that of Speed3R-π^3, we ablate the design choices specific to the *selection branch* of Speed3R-VGGT in Table 6. Removing the reference frame attention (1) impairs performance on both datasets, reflecting the inherent inductive bias of the original VGGT. Following Speed3R-π^3, we also try to remove the register token. However, we find that removing the register tokens (2) also degrades performance. We find these tokens provide crucial stability by acting as an assistant to the camera token; while it is not used for the final pose prediction, its absence clearly impairs model accuracy. Finally, forcing patch tokens to attend to all special tokens (3) hurts long-sequence performance. We hypothesize that for longer sequences, these appended special tokens will dominate over the more crucial top-k patch selection, thereby impairing performance.

Table 4. **Pointmap Estimation on the DTU [15] and ETH3D [31] datasets**. The arrows (↓/↑) indicate whether lower or higher values are better. Best results are highlighted in **bold**.

Method	DTU [15]						ETH3D [31]					
	Acc. ↓		Comp. ↓		N.C. ↑		Acc. ↓		Comp. ↓		N.C. ↑	
	Mean	Med.	Mean	Med.	Mean	Med.	Mean	Med.	Mean	Med.	Mean	Med.
VGGT [39]	1.403	0.802	2.566	1.307	0.658	0.742	0.289	0.192	0.294	0.173	0.847	0.953
Block Sparse-VGGT [36]	1.966	1.052	2.311	1.135	0.647	0.715	0.861	0.754	1.171	0.812	0.681	0.772
FastVGGT-VGGT [32]	1.466	**0.786**	2.385	1.188	0.654	0.736	0.510	0.379	0.580	0.354	0.788	0.913
Speed3R-VGGT	**1.426**	0.827	**2.179**	**1.101**	**0.657**	**0.740**	**0.295**	**0.190**	**0.289**	**0.168**	**0.853**	**0.953**
π^3 [43]	1.151	0.622	1.793	0.629	0.668	0.754	0.194	0.130	0.220	0.135	0.867	0.965
Block Sparse-π^3 [36]	2.434	1.130	2.714	1.004	**0.664**	**0.749**	0.313	0.235	0.439	0.276	0.816	0.951
FastVGGT-π^3 [32]	1.255	0.737	2.250	0.857	0.650	0.730	0.291	0.215	0.291	0.179	0.841	0.961
Speed3R-π^3	**1.175**	**0.710**	**2.037**	**0.731**	0.657	0.739	**0.198**	**0.136**	**0.213**	**0.126**	**0.878**	**0.970**

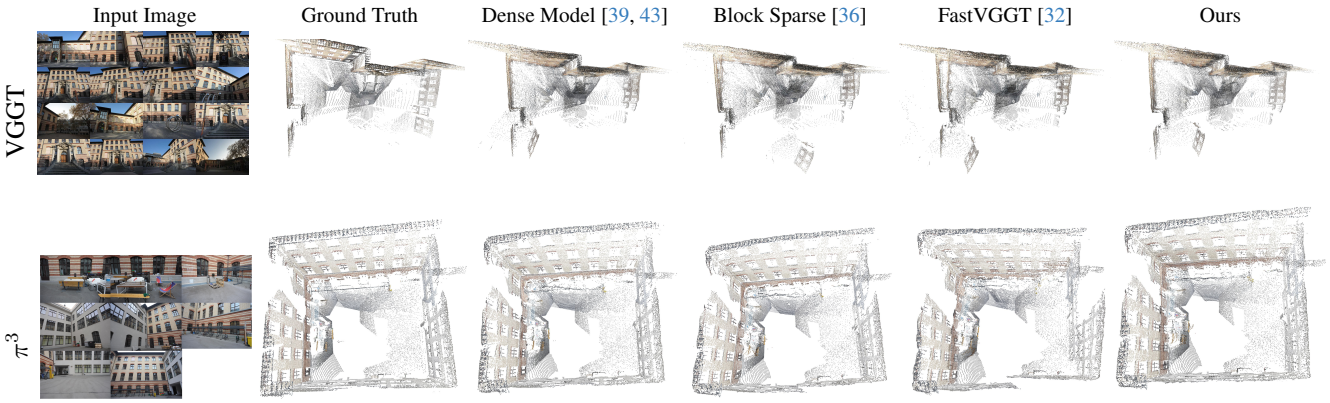


Figure 3. **Qualitative comparison of predicted point clouds from our method and baselines.** Compared with sparse baselines, our method produces visually more accurate reconstructions and reduces the multi-layer wall phenomenon across both architectures.

Table 5. **Ablation study for Speed3R-π^3.** The "Time" column reports the average inference time on the T&T [16] dataset.

Method	RE10K [55] AUC@30 ↑	T&T [16] AUC@30 ↑	Time s↓
Base	86.35	78.69	4.19
(1) w/o Compress. Branch Value	86.29	77.90	3.99
(2) w/o Select. Branch	83.44	76.84	3.56
(3) w/ register	86.39	78.57	4.25
(4) Top-8	85.37	78.17	3.72
(5) Top-16	85.98	78.55	3.92
(6) Top-64	86.42	78.90	4.64
(7) 8x8 window	86.49	78.71	5.27
(8) w/o distillation	85.18	77.81	4.19

Table 6. **Ablation Study of Speed3R-VGGT.** Analysis of the selection attention branch of GSA in Speed3R-VGGT.

Method	RE10K [55] AUC@30 ↑	T&T [16] AUC@30 ↑	Time s↓
Base	74.56	76.57	6.52
(1) w/o reference frame attn.	73.87	75.69	5.91
(2) w/o register token	74.14	76.21	6.50
(3) w/ all special token	74.91	74.77	7.05

4.6. Latency Benchmarking

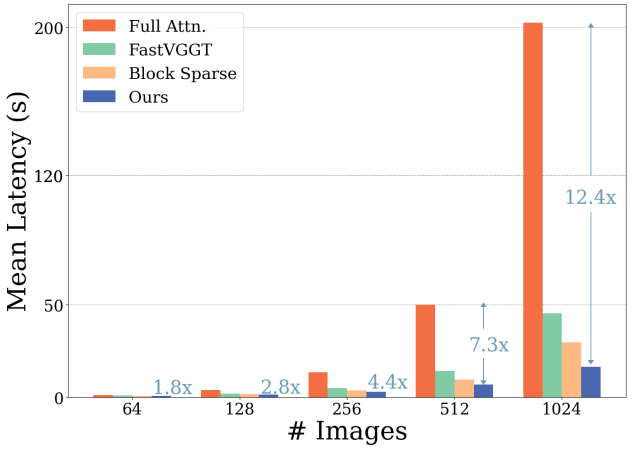
As shown in Figure 4, the Speed3R-π^3 model demonstrates superior inference speed compared to all baselines. This efficiency arises from its computational complexity, which avoids the quadratic $O(n^2)$ of the standard full-attention baseline. The performance advantage becomes increasingly pronounced with sequence length, reaching a 12.4x speedup at an input length of 1024. This result highlights our model's efficacy for high-throughput, long-sequence processing. The benchmarking results for Speed3R-VGGT can be found in the supplementary materials.

5. Discussions

Dense *vs* Sparse Attention. While our method establishes a new efficiency-accuracy *Pareto* frontier, the accuracy of our sparse model on short sequences does not yet

Figure 4. **Inference Time for Different Models.** Mean latency (in seconds) for varying sequence lengths. Our method achieves a 12.4× speedup on sequences of 1024 images.

Seq Length	32	64	128	256	512	1024
Full Attn.(π^3)	0.50	1.31	3.97	13.41	50.01	202.39
Block Sparse [36]	0.46	0.85	1.69	3.77	9.64	29.58
FastVGGT [32]	0.44	0.88	1.96	4.95	14.13	45.49
Ours	0.37	0.71	1.44	3.06	6.83	16.38



match its dense counterparts. We attribute this gap primarily to limitations in data and computational resources. To investigate this, we train both the full-attention and our sparse-attention models under identical conditions (details in Supplementary). As illustrated in Figure 5, our sparse model achieves a comparable training loss to the full-attention model, while completing the training process 1.12x faster. This suggests that the models possess similar learning capacities. Therefore, despite not yet exceeding the dense model's accuracy, Speed3R demonstrates a viable path toward greater efficiency.

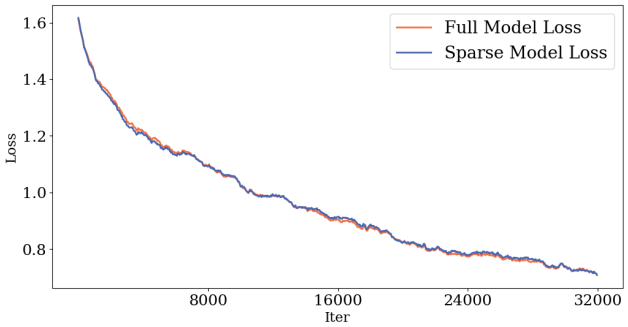
Test-time adaptation. Although our flagship model is trained with top-32, we observe that increasing the top-k value to top-64 and top-128 during inference consistently improves performance on long-sequence datasets (T&T [16]) in Table 7. Notably, this adjustment enables our model to outperform dense models on RTA@5 and AUC@30 during testing, highlighting the robustness of our method and its flexibility in handling long sequences.

Sparsification Challenges. The nature of 3D reconstruction fundamentally differs from domains where sparse methods have traditionally flourished, such as Large Language Models (LLMs) [22, 50] and generation models [44, 52]. These generative domains are inherently probabilistic, and they optimize for perceptual quality and semantic coherence, making them naturally robust to the minor information loss introduced by sparsification. In contrast, 3D reconstruction demands strict numerical precision and is sensitive to minor spatial inaccuracies. Recognizing these challenges, our work presents an initial exploration into developing a sparse attention mechanism for 3D reconstruction, providing a foundational step toward reconciling computational efficiency with geometric precision.

Table 7. **Test-time adaptation on Tanks & Temples. [16].**

Method	RRA@5 ↑	RTA@5 ↑	AUC@30 ↑	Time [s] ↓
π^3 [43]	72.14	81.26	79.63	22.32
Speed3R-π^3(top-8)	69.73	77.60	78.21	3.72
Speed3R-π^3(top-16)	70.26	79.49	79.21	3.92
Speed3R-π^3(top-32)	70.72	80.72	79.77	4.19
Speed3R-π^3(top-64)	71.60	81.54	80.10	4.64
Speed3R-π^3(top-128)	71.89	82.00	80.33	6.07

Figure 5. **Training loss curve of dense attention and our GSA.**



Limitations. The dual-branch architecture of GSA, while enabling sparsity, incurs a 15% memory overhead compared to full attention. In practice, this is manageable, as the model can accommodate up to 1024 images on an 80GB GPU. Looking forward, the strategy proposed in SAIL-Recon [9] provides a path to extend our method to arbitrarily long sequences, removing the memory constraint.

6. Conclusion

In this paper, we introduced Speed3R, a novel sparse attention model designed to mitigate the prohibitive computational cost of feed-forward 3D reconstruction. Inspired by the efficiency of classical SfM and recent advancements in sparse attention, Speed3R employs a trainable dual-branch mechanism to focus computation on a small subset of informative tokens. Our approach establishes a new SoTA in the efficiency-accuracy trade-off, achieving a **12.4x speedup** on 1024-images sequences with minimal impact on geometric accuracy. We validated the robustness of Speed3R by integrating it with two backbones, where it consistently outperformed training-free alternatives, paving the way for practical and scalable large-scale 3D scene modeling.

7. Acknowledgement

This work is supported by Hong Kong Research Grant Council - General Research Fund (Grant No. 17213825) and HKU Seed Fund for PI Research. The authors would like to thank Hongjun Wang for insightful discussions regarding kernel implementation, Fernando Julio Cendra for proofreading, and Zhiheng Wu and Yumeng Zhang for generously sharing GPU resources during the computationally intensive phases of the experiments.

References

[1] Gilad Baruch, Zhuoyuan Chen, Afshin Dehghan, Tal Dimry, Yuri Feigin, Peter Fu, Thomas Gebauer, Brandon Joffe, Daniel Kurz, Arik Schwartz, and Elad Shulman. ARKitscenes - a diverse real-world dataset for 3d indoor scene understanding using mobile RGB-d data. In *NeurIPS*, 2021. 5, 2

[2] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-up robust features (surf). *Computer vision and image understanding*, 2008. 1, 2

[3] Iz Beltagy, Matthew E. Peters, and Arman Cohan. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*, 2020. 2

[4] Yohann Cabon, Naila Murray, and Martin Humenberger. Virtual kitti 2. *arXiv preprint arXiv:2001.10773*, 2020. 5

[5] Yohann Cabon, Naila Murray, and Martin Humenberger. Virtual kitti 2, 2020. 2

[6] Shengqu Cai, Ceyuan Yang, Lvmin Zhang, Yuwei Guo, Junfei Xiao, Ziyan Yang, Yinghao Xu, Zhenheng Yang, Alan Yuille, Leonidas Guibas, Maneesh Agrawala, Lu Jiang, and Gordon Wetzstein. Mixture of contexts for long video generation. *arXiv preprint arXiv:2508.21058*, 2025. 2

[7] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *CVPR*, 2017. 5

[8] Tri Dao. FlashAttention-2: Faster attention with better parallelism and work partitioning. In *ICLR*, 2024. 4, 1

[9] Junyuan Deng, Heng Li, Tao Xie, Weiqiang Ren, Qian Zhang, Ping Tan, and Xiaoyang Guo. Sail-recon: Large sfm by augmenting scene regression with localization. *arXiv preprint arXiv:2508.17972*, 2025. 5, 6, 8

[10] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. In *CVPR*, 2018. 1

[11] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR*, 2021. 1

[12] Yizhao Gao, Zhichen Zeng, Dayou Du, Shijie Cao, Peiyuan Zhou, Jiaxing Qi, Junjie Lai, Hayden Kwok-Hay So, Ting Cao, Fan Yang, et al. Seerattention: Learning intrinsic sparse attention in your llms. *arXiv preprint arXiv:2410.13276*, 2024. 2

[13] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000. 1, 2

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *ICCV*, 2015. 2

[15] Rasmus Jensen, Anders Dahl, George Vogiatzis, Engil Tola, and Henrik Aanæs. Large scale multi-view stereopsis evaluation. In *CVPR*, 2014. 7

[16] Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM TOG*, 2017. 6, 7, 8

[17] Vincent Leroy, Yohann Cabon, and Jerome Revaud. Grounding image matching in 3d with mast3r. In *ECCV*, 2024. 2

[18] Philipp Lindenberger, Paul-Edouard Sarlin, and Marc Pollefeys. Lightglue: Local feature matching at light speed. *arXiv preprint arXiv:2306.13643*, 2023. 2

[19] Lu Ling, Yichen Sheng, Zhi Tu, Wentian Zhao, Cheng Xin, Kun Wan, Lantao Yu, Qianyu Guo, Zixun Yu, Yawen Lu, et al. Dl3dv-10k: A large-scale scene dataset for deep learning-based 3d vision. In *CVPR*, 2024. 5, 2

[20] Guangda Liu, Chengwei Li, Jieru Zhao, Chenqi Zhang, and Minyi Guo. Clusterkv: Manipulating llm kv cache in semantic space for recallable compression. In *2025 62nd ACM/IEEE Design Automation Conference (DAC)*, 2025. 2

[21] David G Lowe. Object recognition from local scale-invariant features. In *ICCV*, 1999. 1, 2

[22] Enzhe Lu, Zhejun Jiang, Jingyuan Liu, Yulun Du, Tao Jiang, Chao Hong, Shaowei Liu, Weiran He, Enming Yuan, Yuzhi Wang, et al. Moba: Mixture of block attention for long-context llms. *arXiv preprint arXiv:2502.13189*, 2025. 2, 8

[23] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023. 1, 3, 2

[24] Linfei Pan, Daniel Barath, Marc Pollefeys, and Johannes Lutz Schönberger. Global Structure-from-Motion Revisited. In *ECCV*, 2024. 1, 2

[25] 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. In *ICCV*, 2021. 5, 6, 2

[26] Mike Roberts, Jason Ramapuram, Anurag Ranjan, Atulit Kumar, Miguel Angel Bautista, Nathan Paczan, Russ Webb, and Joshua M. Susskind. Hypersim: A photorealistic synthetic dataset for holistic indoor scene understanding. In *ICCV*, 2021. 5, 2

[27] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. Orb: An efficient alternative to sift or surf. In *ICCV*, 2011. 1, 2

[28] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks. In *CVPR*, 2020. 1, 2, 5

[29] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *CVPR*, 2016. 1, 2

[30] Johannes Lutz Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. Pixelwise view selection for unstructured multi-view stereo. In *ECCV*, 2016. 1, 2

[31] Thomas Schops, Johannes L Schonberger, Silvano Galliani, Torsten Sattler, Konrad Schindler, Marc Pollefeys, and Andreas Geiger. A multi-view stereo benchmark with high-resolution images and multi-camera videos. In *CVPR*, 2017. 7

[32] You Shen, Zhipeng Zhang, Yansong Qu, and Liujuan Cao. Fastvggt: Training-free acceleration of visual geometry transformer. *arXiv preprint arXiv:2509.02560*, 2025. 2, 5, 6, 7, 8, 1, 3

[33] Jiaming Tang, Yilong Zhao, Kan Zhu, Guangxuan Xiao, Baris Kasikci, and Song Han. Quest: Query-aware sparsity for efficient long-context llm inference. *ICML*, 2024. 2

[34] Zachary Teed and Jia Deng. Droid-slam: Deep visual slam for monocular, stereo, and rgb-d cameras. *NeurIPS*, 2021. 2

[35] Philippe Tillet, Hsiang-Tsung Kung, and David Cox. Triton: an intermediate language and compiler for tiled neural network computations. In *Proceedings of the 3rd ACM SIGPLAN International Workshop on Machine Learning and Programming Languages*, 2019. 4

[36] Chung-Shien Brian Wang, Christian Schmidt, Jens Piekenbrinck, and Bastian Leibe. Faster vggt with block-sparse global attention. *arXiv preprint arXiv:2509.07120*, 2025. 2, 5, 6, 7, 8, 1

[37] Hengyi Wang and Lourdes Agapito. 3d reconstruction with spatial memory. *arXiv preprint arXiv:2408.16061*, 2024. 2

[38] Jianyuan Wang, Nikita Karaev, Christian Rupprecht, and David Novotny. Vggsfm: Visual geometry grounded deep structure from motion. In *CVPR*, 2024. 2

[39] Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *CVPR*, 2025. 1, 2, 3, 5, 6, 7

[40] Qianqian Wang, Yifei Zhang, Aleksander Holynski, Alexei A. Efros, and Angjoo Kanazawa. Continuous 3d perception model with persistent state. In *CVPR*, 2025. 2

[41] Ruicheng Wang, Sicheng Xu, Cassie Dai, Jianfeng Xiang, Yu Deng, Xin Tong, and Jiaolong Yang. Moge: Unlocking accurate monocular geometry estimation for open-domain images with optimal training supervision. *arXiv preprint arXiv:2410.19115*, 2024. 2

[42] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *CVPR*, 2024. 1, 2

[43] Yifan Wang, Jianjun Zhou, Haoyi Zhu, Wenzheng Chang, Yang Zhou, Zizun Li, Junyi Chen, Jiangmiao Pang, Chunhua Shen, and Tong He. π^3: Scalable permutation-equivariant visual geometry learning. In *ICLR*, 2025. 1, 2, 3, 5, 6, 7, 8

[44] Shuang Wu, Youtian Lin, Feihu Zhang, Yifei Zeng, Yikang Yang, Yajie Bao, Jiachen Qian, Siyu Zhu, Philip Torr, Xun Cao, and Yao Yao. Direct3d-s2: Gigascale 3d generation made easy with spatial sparse attention. *NeurIPS*, 2025. 2, 8

[45] Hongchi Xia, Yang Fu, Sifei Liu, and Xiaolong Wang. Rgbd objects in the wild: Scaling real-world 3d object learning from rgb-d videos. In *arXiv preprint arXiv: 2401.12592*, 2024. 5, 2

[46] Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks. *ICLR*, 2024. 2

[47] Jianing Yang, Alexander Sax, Kevin J Liang, Mikael Henaff, Hao Tang, Ang Cao, Joyce Chai, Franziska Meier, and Matt Feiszli. Fast3r: Towards 3d reconstruction of 1000+ images in one forward pass. In *CVPR*, 2025. 1

[48] Yao Yao, Zixin Luo, Shiwei Li, Tian Fang, and Long Quan. Mvsnet: Depth inference for unstructured multi-view stereo. In *ECCV*, 2018. 2

[49] Chandan Yeshwanth, Yueh-Cheng Liu, Matthias Nießner, and Angela Dai. Scannet++: A high-fidelity dataset of 3d indoor scenes. In *ICCV*, 2023. 5, 2

[50] Jingyang Yuan, Huazuo Gao, Damai Dai, Junyu Luo, Liang Zhao, Zhengyan Zhang, Zhenda Xie, Yuxing Wei, Lean Wang, Zhiping Xiao, et al. Native sparse attention: Hardware-aligned and natively trainable sparse attention. In *ACL*, 2025. 2, 8, 1

[51] Jintao Zhang, Chendong Xiang, Haofeng Huang, Jia Wei, Haocheng Xi, Jun Zhu, and Jianfei Chen. Spargeattn: Accurate sparse attention accelerating any model inference. In *ICML*, 2025. 1

[52] Peiyuan Zhang, Yongqi Chen, Haofeng Huang, Will Lin, Zhengzhong Liu, Ion Stoica, Eric Xing, and Hao Zhang. Vsa: Faster video diffusion with trainable sparse attention. *arXiv preprint arXiv:2505.13389*, 2025. 2, 8

[53] Zhe Zhang, Rui Peng, Yuxi Hu, and Ronggang Wang. Geomvsnet: Learning multi-view stereo with geometry perception. In *CVPR*, 2023. 2

[54] Zhenyu Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, Ruisi Cai, Zhao Song, Yuandong Tian, Christopher Ré, Clark Barrett, et al. H2o: Heavy-hitter oracle for efficient generative inference of large language models. *NeurIPS*, 2023. 2

[55] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: learning view synthesis using multiplane images. *ACM TOG*, 2018. 5, 6, 7