

TurboVGGT: Fast Visual Geometry Reconstruction with Adaptive Alternating Attention

David Huang^{1,2,†}, Guile Wu^{1,†}, Chengjie Huang¹, Bingbing Liu³, and Dongfeng Bai¹

¹Huawei Noah's Ark Lab ²University of Toronto ³Foundation Model Department, Huawei

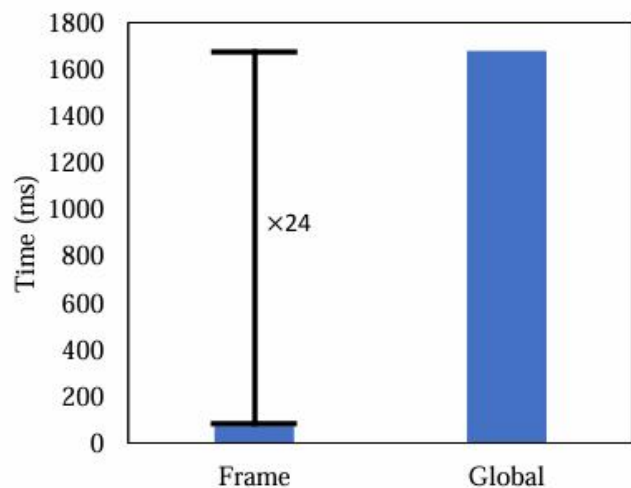
[†]Equal contribution *Project page: <https://turbovggt.github.io/>

NeurIPS 2026

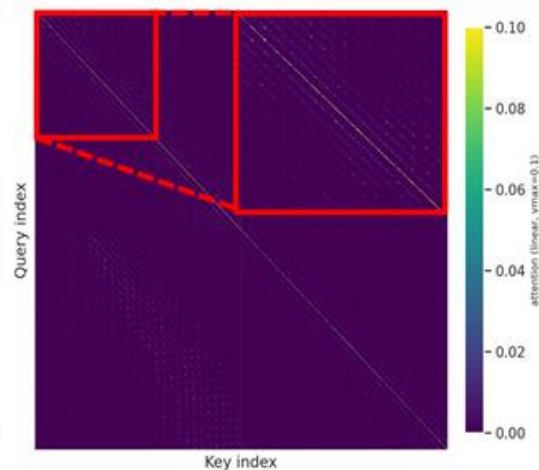


• Problem & Motivation

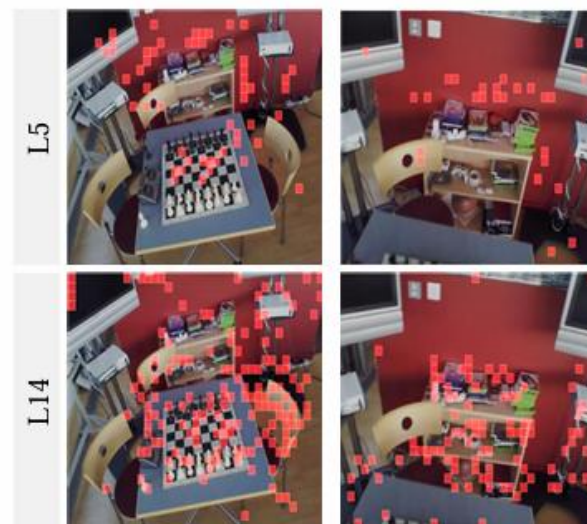
- This work focuses on fast multi-view 3D reconstruction in a feed-forward manner.
- The high computational cost of visual geometry transformers primarily lies in the dense global token interactions within global attention layers.
- Existing fast visual geometry reconstruction methods typically use the same sparsity ratio across layers and frames and lack mechanisms to adaptively learn representative tokens to capture global relationships, leading to suboptimal performance.



(a) Runtime comparison between frame attention and global attention.

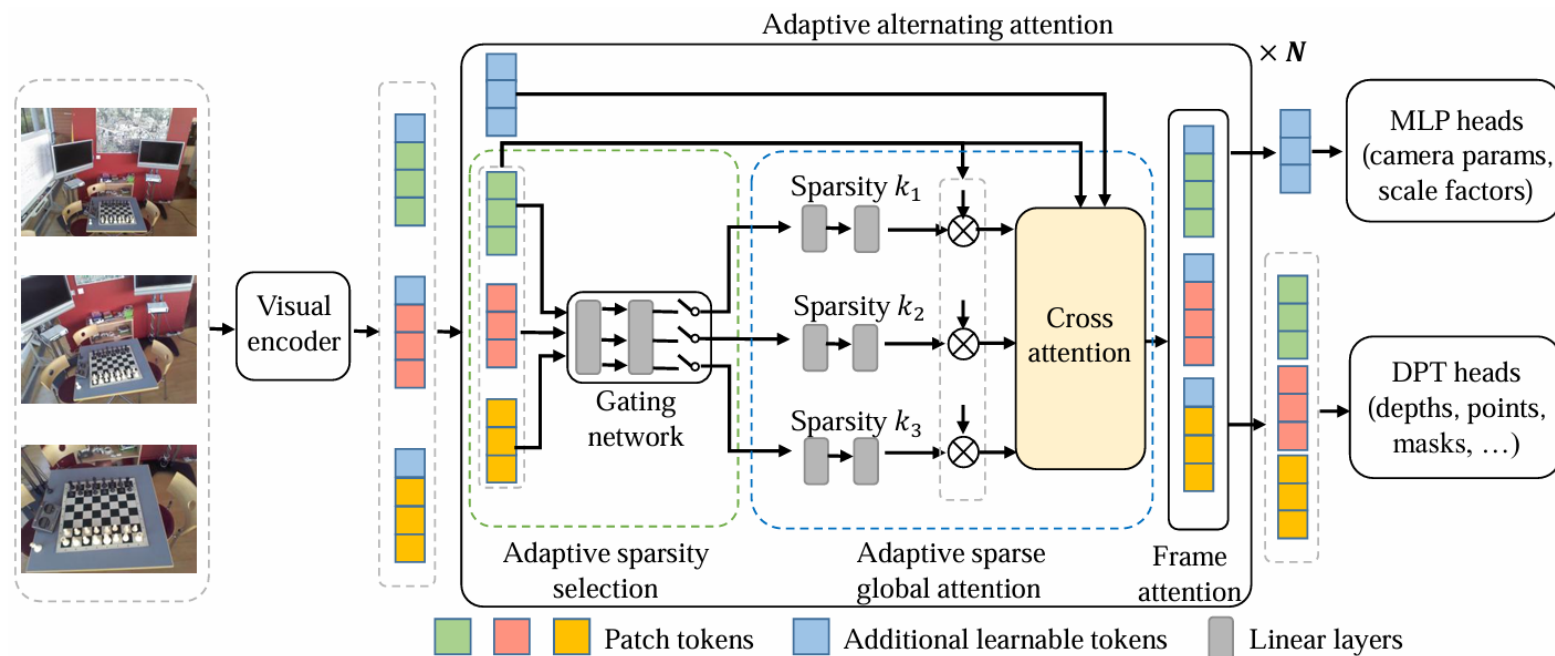


(b) Visualization of a global attention map.



(c) Illustration of highly activated patch tokens across layers and frames in global attention.

• The Proposed Method



Contributions:

- We present a novel visual geometry transformer with adaptive alternating attention blocks for fast multi-view 3D reconstruction.
- We propose an adaptive sparsity selection mechanism for visual geometry transformers, which adaptively selects different sparsity ratios for different frames across different layers.
- We propose adaptive sparse global attention for visual geometry transformers to learn representative tokens to model global geometry relationships.

• The Proposed Method

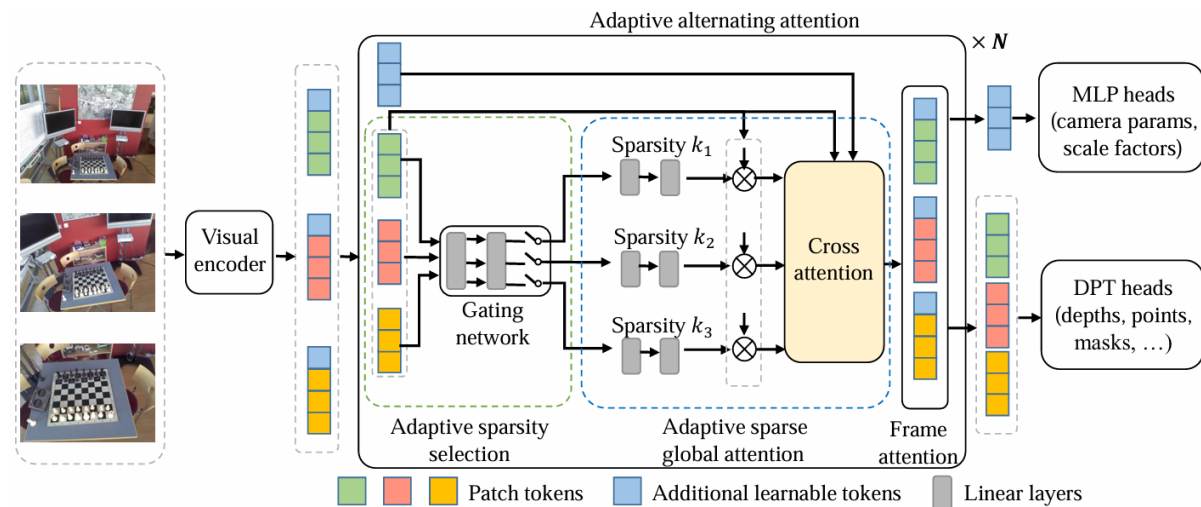
- **TurboVGGT** consists of a visual encoder, N adaptive alternating attention blocks, and multiple task-specific prediction heads.
- **Visual Encoder:** We use DINOv2 to extract patch tokens from each image and append additional learnable tokens.
- **Adaptive Alternating Attention Blocks:** Each adaptive alternating attention block consists of Adaptive Sparsity Selection

$$k_k = F_s(F_g(F_a(\{x_{i,j}\}_{j=1}^M)))$$

and Adaptive Sparse Global Attention

$$W_i = F_w(x_i) + B_k, \quad x_i^c = W_i^T x_i \quad x'' = \text{CrossAttn}(x'W^Q, x^cW^K, x^cW^V)$$

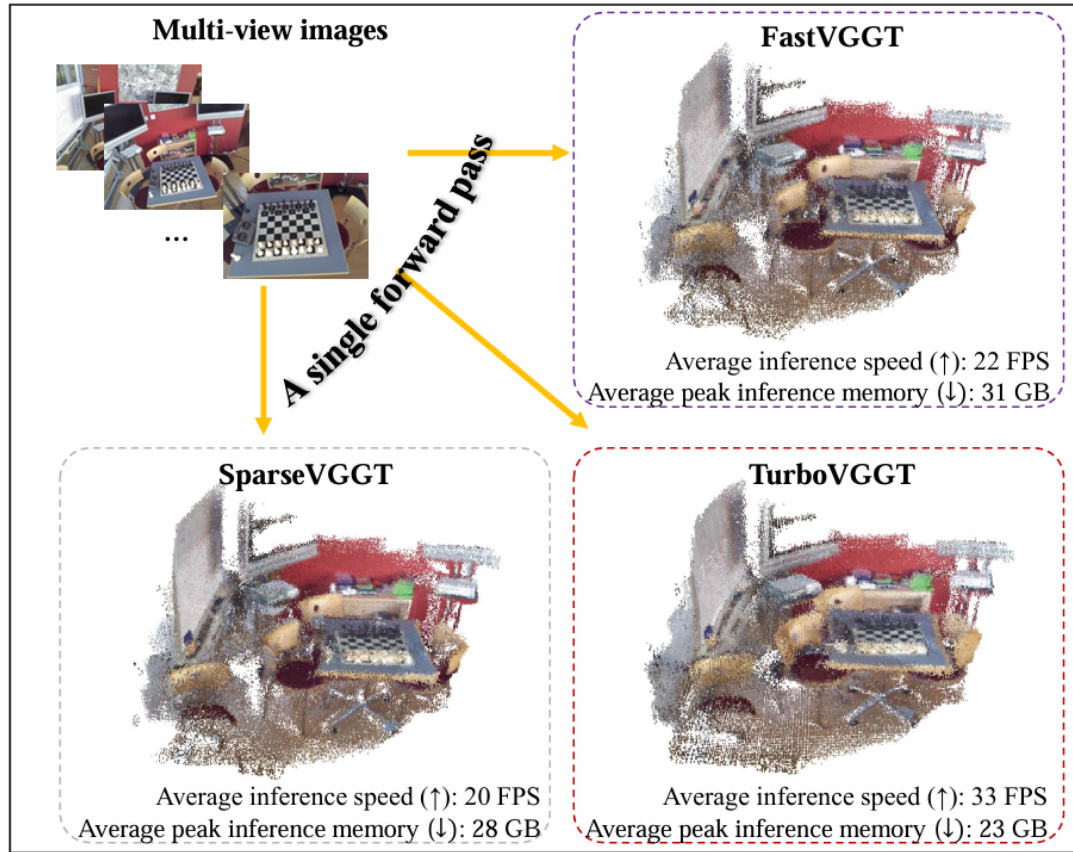
- **Task-Specific Prediction Heads:** We aggregate the output tokens and employ multiple task-specific prediction heads to decode the learned tokens into various 3D properties of each view.



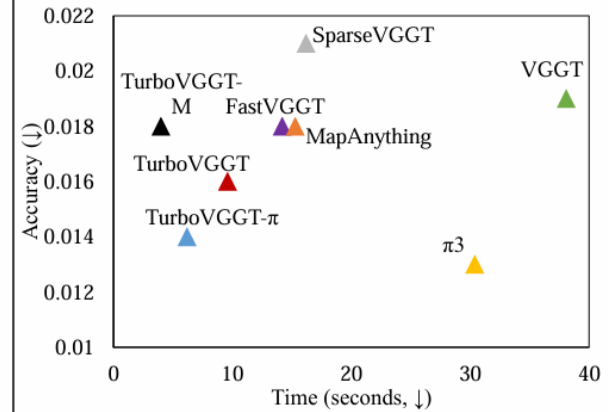
- **Model Optimization:** TurboVGGT is trained in an end-to-end manner using a combination of losses for different prediction heads and a regularization term for the adaptive sparsity selection.

$$\mathcal{L} = \mathcal{L}_{\text{recon}} + \lambda \mathcal{L}_{\text{reg}}$$

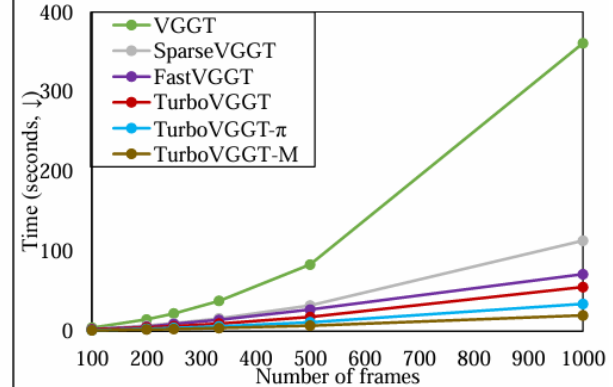
• Experiment



(a) Visualization of point cloud reconstruction results.



(b) Inference time ($\downarrow$) vs. Accuracy ($\downarrow$).



(c) Number of frames vs. Inference time ($\downarrow$).

- TurboVGGT improves computational efficiency in terms of inference time and peak GPU memory usage.
- TurboVGGT achieves a better trade-off between accuracy and efficiency.
- TurboVGGT scales more efficiently with the number of input frames.

• Experiment

Table: Evaluation of point cloud reconstruction on 7-Scenes, N-RGBD, and ScanNet.

Method	7-Scenes - Stride 3							7-Scenes - Stride 10						
	Acc ↓		Comp ↓		NC ↑		Time ↓	Acc ↓		Comp ↓		NC ↑		Time ↓
	Mean	Med.	Mean	Med.	Mean	Med.		Mean	Med.	Mean	Med.	Mean	Med.	
Fast3R [49]	0.045	0.027	0.047	0.010	0.616	0.627	21.2s	0.040	0.021	0.056	0.013	0.639	0.657	2.3s
VGGT [42]	0.019	0.009	0.027	0.010	0.622	0.699	38.1s	0.019	0.008	0.027	0.010	0.628	0.710	4.5s
SparseVGGT [41]	0.021	0.010	0.029	0.010	0.621	0.693	16.2s	0.020	0.010	0.028	0.010	0.629	0.707	2.6s
AVGGT [37]*	0.054	0.031	0.131	0.078	0.528	0.541	20.6s	-	-	-	-	-	-	-
FastVGGT [35]	0.018	0.008	0.026	0.010	0.627	0.707	14.2s	0.018	0.008	0.027	0.010	0.632	0.717	2.7s
TurboVGGT (ours)	0.016	0.009	0.026	0.010	0.639	0.714	9.6s	0.016	0.009	0.025	0.010	0.650	0.730	2.0s

Method	N-RGBD - Stride 3							N-RGBD - Stride 10						
	Acc ↓		Comp ↓		NC ↑		Time ↓	Acc ↓		Comp ↓		NC ↑		Time ↓
	Mean	Med.	Mean	Med.	Mean	Med.		Mean	Med.	Mean	Med.	Mean	Med.	
Fast3R [49]	0.074	0.036	0.024	0.011	0.658	0.682	37.0s	0.061	0.028	0.031	0.013	0.669	0.712	3.9s
VGGT [42]	0.028	0.018	0.018	0.010	0.657	0.779	65.3s	0.016	0.010	0.016	0.009	0.669	0.791	7.2s
SparseVGGT [41]	0.045	0.029	0.024	0.011	0.647	0.755	26.2s	0.041	0.027	0.027	0.012	0.652	0.763	4.1s
FastVGGT [35]	0.031	0.020	0.019	0.010	0.662	0.788	30.2s	0.017	0.011	0.017	0.009	0.671	0.791	4.0s
TurboVGGT (ours)	0.025	0.016	0.022	0.011	0.657	0.773	14.7s	0.021	0.014	0.021	0.011	0.664	0.783	2.9s

Method	ScanNet - 500 Frames		ScanNet - 300 Frames		ScanNet - 100 Frames		Average	
	CD ↓	Time ↓	CD ↓	Time ↓	CD ↓	Time ↓	CD ↓	Time ↓
Fast3R [49]	0.701	62.5s	0.711	17.2s	0.723	2.5s	0.712	27.4s
VGGT [42]	0.464	90.1s	0.454	33.6s	0.438	4.9s	0.452	42.9s
SparseVGGT [41]	0.459	34.2s	0.453	14.2s	0.446	2.6s	0.453	17.0s
FastVGGT [35]	0.453	28.4s	0.447	12.3s	0.425	2.8s	0.442	14.5s
TurboVGGT (ours)	0.416	20.3s	0.413	9.5s	0.400	2.2s	0.410	10.7s

Table: Evaluation of camera pose estimation on 7-Scenes, N-RGBD, and RealEstate10K.

Method	7-Scenes (dense)				N-RGBD (dense)				RealEstate10K		
	RRA@30↑	RTA@30↑	AUC@30↑	Time ↓	RRA@30↑	RTA@30↑	AUC@30↑	Time ↓	RRA@30↑	RTA@30↑	AUC@30↑
Fast3R [49]	-	-	-	-	-	-	-	-	99.05	81.86	61.68
FLARE [54]	-	-	-	-	-	-	-	-	99.69	95.23	80.01
VGGT [42]	100.00	96.58	77.76	38.1s	100.00	99.47	91.59	65.3s	99.97	96.22	85.32
SparseVGGT [41]	100.00	93.40	70.07	16.2s	97.20	97.03	83.16	26.2s	99.61	91.35	76.65
AVGGT [37]	100.00	96.48	78.29	20.6s	-	-	-	-	99.93	95.62	85.45
Speed3R [32]	-	-	-	-	-	-	-	-	-	-	74.81
FlashVGGT [47]	-	-	-	-	-	-	-	-	99.92	95.61	85.30
FastVGGT [35]	99.99	96.29	76.90	14.2s	100.00	99.65	92.47	30.2s	99.92	94.76	84.37
TurboVGGT (ours)	100.00	96.83	81.87	9.6s	100.00	99.71	93.28	14.7s	99.93	94.69	84.31

Table: Evaluation of camera pose estimation on 7-Scenes, N-RGBD, and Sintel.

Method	7-Scenes (dense)			N-RGBD (dense)			Sintel	
	AbsRel↓	$\delta < 1.25 \uparrow$	Time ↓	AbsRel↓	$\delta < 1.25 \uparrow$	Time ↓	AbsRel↓	$\delta < 1.25 \uparrow$
Fast3R [49]	-	-	-	-	-	-	0.544	0.509
FLARE [54]	-	-	-	-	-	-	0.606	0.402
VGGT [42]	0.264	0.958	38.1s	0.013	0.993	65.3s	0.335	0.599
SparseVGGT [41]	0.393	0.956	16.2s	0.015	0.990	26.2s	-	-
FlashVGGT [47]	-	-	-	-	-	-	0.346	0.586
FastVGGT [35]	0.394	0.953	14.2s	0.013	0.990	30.2s	0.337	0.582
TurboVGGT (ours)	0.296	0.980	9.6s	0.013	0.994	14.7s	0.287	0.650

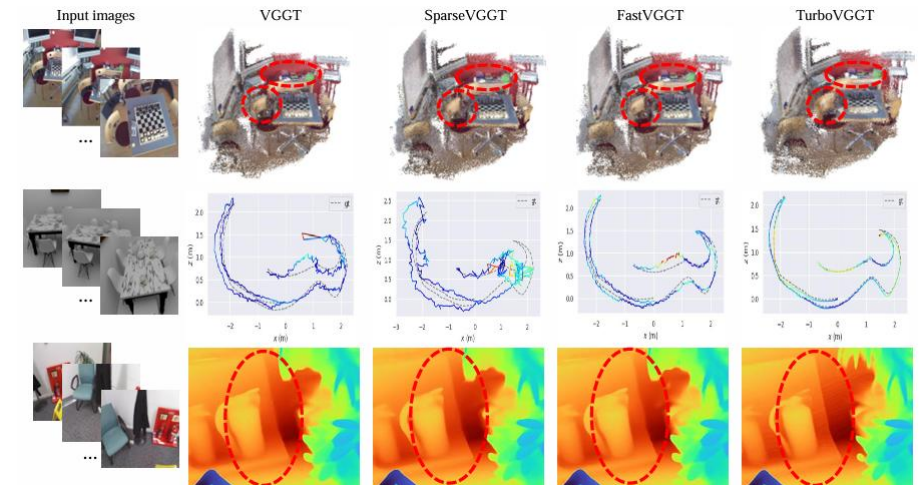


Figure: Qualitative comparison of point cloud reconstruction, camera pose estimation, and depth estimation.

- **Conclusion**

- In this work, we propose TurboVGGT for fast multi-view 3D reconstruction in a feed-forward manner.
- The core idea of TurboVGGT is to construct an efficient visual geometry transformer with adaptive alternating attention blocks, which consist of adaptive sparse global attention guided by adaptive sparsity selection for global geometry modeling and frame attention for local detail aggregation.
- Extensive experiments on multiple 3D reconstruction benchmarks demonstrate that TurboVGGT achieves a better balance between reconstruction quality and computational efficiency, outperforming state-of-the-art methods.

Thanks for your attention !