HTTM: Head-wise Temporal Token Merging for Faster VGGT

Supplementary Material

A. RoPE's Effect on the Similarity Pattern

In this section, we investigate the Rotary Position Embedding (RoPE [23])'s effect on the strong periodic patterns observed in Fig. 3. To show that the high similarity values near the off-diagonals do emerge from the RoPE, in Fig. 11, we visualize the similarity map of query tokens in the early Frame Attention layers with **non-overlapping** input frames. It can be observed that the input feature of the first Frame Attention layer (DINO features) does not exhibit high temporal values near off-diagonals (Fig. 11a), which align with the non-overlapping input frames. However, after applying the frame-wise RoPE for the first time, high similarity values near the off diagonals emerge as shown in Fig. 11b. After that, it can be observed that, in each layer, the spatial distinctiveness within frames is enhanced after applying the frame-wise RoPE.

In Fig. 12, we visualize the similarity maps with temporally continuous input frames. With these inputs, the changes in similarity patterns before and after applying RoPE are similar, but high similarity values near the off-diagonals are more vivid.

B. Theoretical Proofs For Block-Wise Token Merging

We provide proofs for the three statements made in Sec. 3.3. For clarity, we omit the head index i in this section. Let the global source and destination token sets be $\mathcal{S}$ and $\mathcal{D}$, $\mathcal{S} \cap \mathcal{D} = \emptyset$. The entries of the global similarity matrix $W \in \mathbb{R}^{|\mathcal{S}|\times|\mathcal{D}|}$ is defined as:

$$W_{ij} = \mathrm{sim}(s_i, d_i), \quad s_i \in \mathcal{S}, d_j \in \mathcal{D}$$

For a merging budget r, the merging rule selects the *top-r* source–destination pairs with the largest similarities. For any selected set $\mathcal{M}$ of merging candidates, the merging quality Q is defined as the *average* similarity between merged matches:

$$Q(\mathcal{M}) := \frac{1}{r} \sum_{(s,d)\in\mathcal{M}} \mathrm{sim}(s, d).$$

In block-wise token merging, we partition the tokens into K disjoint blocks $\{\mathcal{B}_1, \ldots, \mathcal{B}_K\}$. Assuming the same splitting strategy, the source and destination token sets $\mathcal{S}_k$ and $\mathcal{D}_k$ inside block $k \in \{1, \ldots, K\}$ are:

$$\mathcal{S}_k = \mathcal{S} \cap \mathcal{B}_k, \qquad \mathcal{D}_k = \mathcal{D} \cap \mathcal{B}_k,$$

After establishing the notations, we proceed to prove the aforementioned statements.

1. Block similarity matrices are submatrices of the global matrix

Prop. For every block $\mathcal{B}_k$, its block-wise similarity matrix $W^{(k)}$ is a submatrix of W.

Proof. For each $\mathcal{B}_k$, the entries of its similarity matrix is defined as:

$$W^{(k)}_{i,j} = \mathrm{sim}(s_i, d_j), \qquad s_i \in \mathcal{S}_k,\ d_j \in \mathcal{D}_k.$$

Since $\mathcal{S}_k \subseteq \mathcal{S}, \mathcal{D}_k \subseteq \mathcal{D}$, it follows that $s_i \in \mathcal{S},\ d_j \in \mathcal{D}$. Therefore, each entry $\mathrm{sim}(s_i, d_j)$ is also a entry in W.

2. Merging quality depends on how many high-similarity pairs fall inside blocks

Prop. Let $\mathcal{E} = \mathcal{S} \times \mathcal{D}$ be all possible source–destination pairs, and let

$$\mathcal{E}_{\mathrm{blk}} := \bigcup_{k=1}^{K} (\mathcal{S}_k \times \mathcal{D}_k)$$

be the set of pairs permitted by block-wise merging. If more large entries of W lie inside $\mathcal{E}_{\mathrm{blk}}$, then the block-wise merging quality increases.

Proof. As stated in Sec. 3.2, we pick the top-r best matches with the highest similarity, which yields the global optimal merging quality:

$$\mathcal{M}^\star = \arg\max_{\mathcal{M}} \mathrm{sim}(s, d) = \arg\max_{\mathcal{M}} Q(\mathcal{M})$$
$$\text{s.t. } (s, d) \subset \mathcal{M}, \quad \mathcal{M} \subseteq \mathcal{E}, \quad |\mathcal{M}| = r.$$

Block-wise merging is constrained to subsets of $\mathcal{E}_{\mathrm{blk}}$:

$$\mathcal{M}^\star_{blk} = \arg\max_{\mathcal{M}} \mathrm{sim}(s, d)$$
$$\text{s.t. } (s, d) \subset \mathcal{M}, \quad \mathcal{M} \subseteq \mathcal{E}_{blk}, \quad |\mathcal{M}| = r.$$

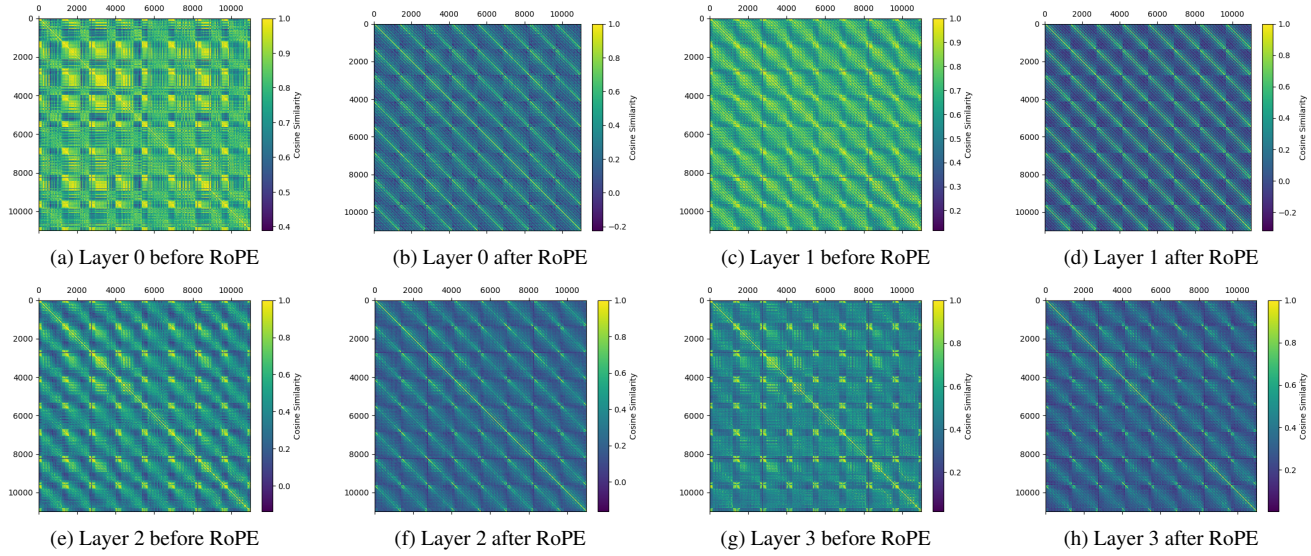
Since $\mathcal{E}_{\mathrm{blk}} \subseteq \mathcal{E}$, the feasible set of block-wise solutions is smaller, hence

$$Q(\mathcal{M}^\star_{\mathrm{blk}}) \le Q(\mathcal{M}^\star).$$

Define H_{blk} as the number of optimal merging candidates included in $\mathcal{M}^\star_{blk}$:

$$H_{blk} := |\mathcal{M}^\star_{blk} \cap \mathcal{M}^\star|$$

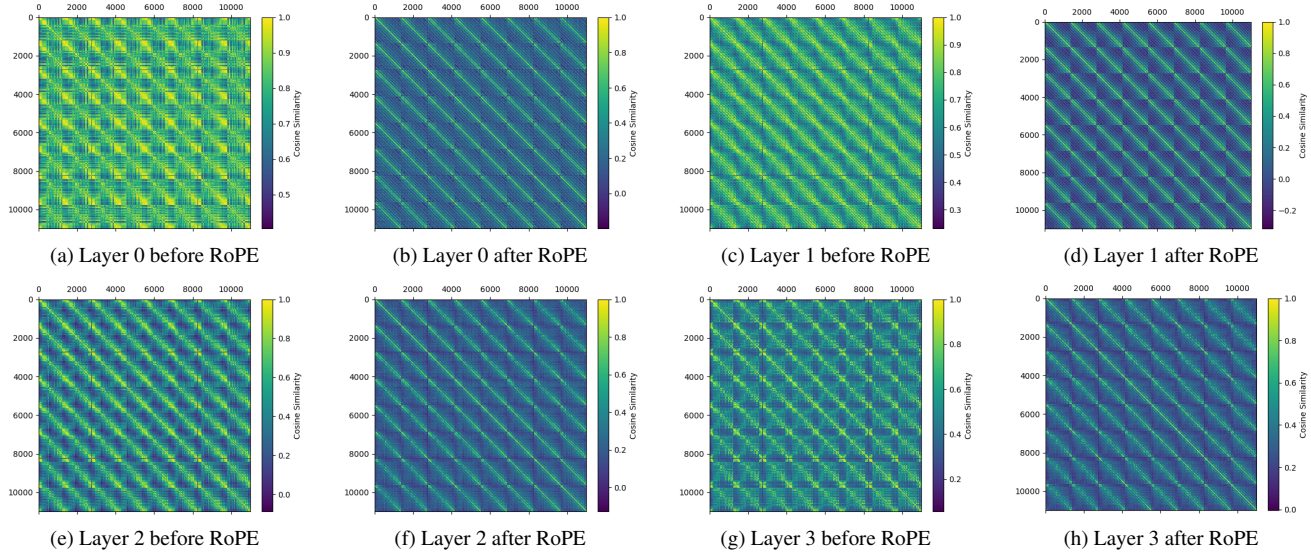
If $H_{blk} < r$, then $Q(\mathcal{M}^\star_{blk})$ is strictly smaller than $Q(\mathcal{M}^\star)$. By including more s, d pairs with high similarity in $\mathcal{E}_{blk}$, we can only increase H_{blk}. Therefore, block-wise merging quality is monotone in the number of high-similarity entries of W located inside the blocks.



(a) Layer 0 before RoPE (b) Layer 0 after RoPE (c) Layer 1 before RoPE (d) Layer 1 after RoPE

(e) Layer 2 before RoPE (f) Layer 2 after RoPE (g) Layer 3 before RoPE (h) Layer 3 after RoPE

Figure 11. Query token similarity maps before and after RoPE in Frame Attention layers with non-overlapping input frames



(a) Layer 0 before RoPE (b) Layer 0 after RoPE (c) Layer 1 before RoPE (d) Layer 1 after RoPE

(e) Layer 2 before RoPE (f) Layer 2 after RoPE (g) Layer 3 before RoPE (h) Layer 3 after RoPE

Figure 12. Query token similarity maps before and after RoPE in Frame Attention layers with temporally continuous input frames.

3. Larger blocks improve merging quality but require more computations

Prop. Fix a splitting strategy that forms blocks of size n_b. If the block size increases, then (i) the achievable merging quality does not decrease, and (ii) the computational cost grows approximately linearly in n_b.

Proof. *(i) Larger blocks improve or maintain quality.*

Let two block sizes $n_b^{(1)} < n_b^{(2)}$ be given. Under the same splitting strategy, every small block is contained in a unique larger block. Hence

$$\mathcal{E}_{\text{blk}}^{(1)} \subseteq \mathcal{E}_{\text{blk}}^{(2)}.$$

Thus, the feasible merging sets under smaller blocks are a subset of those under the larger blocks, so

$$\max_{\substack{\mathcal{M}\subseteq\mathcal{E}_{\text{blk}}^{(1)} \\ |\mathcal{M}|=r}} Q(\mathcal{M}) \;\leq\; \max_{\substack{\mathcal{M}\subseteq\mathcal{E}_{\text{blk}}^{(2)} \\ |\mathcal{M}|=r}} Q(\mathcal{M}).$$

Therefore, the optimal block-wise average similarity is non-decreasing in n_b.

			NRGBD (Stride 3)			ScanNet (500 Frames)			ScanNet (1000 Frames)		
	Q Ratio	K/V Ratio	Acc.↓	Comp.↓	Time↓	Acc.↓	Comp.↓	Time↓	Acc.↓	Comp.↓	Time↓
VGGT* [26]	1.00	1.00	0.010	0.009	135.1s	0.011	0.011	177.5s	0.028	0.022	724.6s
FastVGGT [21]	0.34	0.34	0.014	0.020	51.2s	0.012	0.011	52.3s	0.027	0.021	175.2s
VGGT*+HTTM	**0.20**	0.30	0.010	0.008	26.4s	0.011	0.010	35.8s	0.027	0.021	**102.8s**

Table 5. 3D reconstruction performance with longer sequence input. With longer sequence inputs, HTTM constantly shows similar performance to the original VGGT with substantially shorter latency.

(ii) Cost increases linearly with block size.

Let $n_s^{(k)} = |\mathcal{S}_k|$ and $n_d^{(k)} = |\mathcal{D}_k|$, with $n_s^{(k)} + n_d^{(k)} = n_b$. Computing similarities inside block k costs $\Theta(n_s^{(k)} n_d^{(k)} d)$ operations, where d is the head dimension. Assuming a fixed source/destination ratio:

$$n_s^{(k)} = \alpha n_b \quad n_d^{(k)} = (1-\alpha) n_b \,,$$

We have that

$$n_s^{(k)} n_d^{(k)} = \alpha(1-\alpha) n_b^2 \,.$$

With $K \approx N/n_b$ blocks, the total cost can be approximated as

$$\sum_{k=1}^{K} \Theta(\alpha(1-\alpha) n_b^2 d) \approx \Theta\big(\alpha(1-\alpha)\frac{N}{n_b} n_b^2 d\big) = O(N n_b d),$$

which grows linearly with block size n_b.

Combining both parts, larger blocks always improve (or maintain) merging quality but incur proportionally higher computational costs.

B.1. Experiments on Longer Sequence

In this section, we show more experiments on longer sequences with NRGBD [1] and ScanNet [8] dataset in Table 5.

Setup For the NRGBD dataset, we evaluate HTTM by sampling keyframes every 3 frames. For the ScanNet dataset, we randomly select 15 scenes with over 2000 input frames and sample keyframes every 2 frames. The setup of FastVGGT and our HTTM is the same as in Sec. 4.1.

Results As shown in Table 5, with longer sequence inputs, HTTM constantly shows similar performance to the original VGGT with substantially shorter latency.

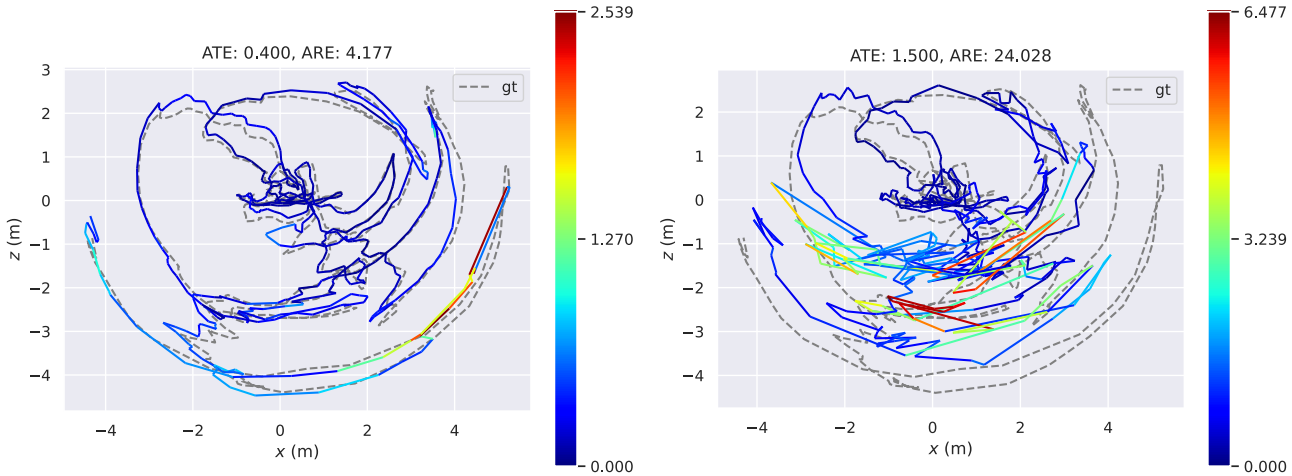
B.2. Error Mitigation

In this section, we discuss the error mitigation effect of token merging on VGGT reported by FastVGGT [21].

The original VGGT shows a large error in camera pose estimation when the camera movement is high. For example, in Fig. 13, we visualize the camera pose estimation error on a big scene from ScanNet with large camera movement, using 500 keyframes sampled every 10 frames. Compared to FastVGGT (Fig. 13a), the original VGGT (Fig. 13b) shows much higher error in camera pose estimation. HTTM shows a smaller error compared to the original VGGT, but still higher than FastVGGT. Although FastVGGT offers a discussion of the observed improvement, the specific mechanism responsible for the enhanced error mitigation ability remains insufficiently clarified.

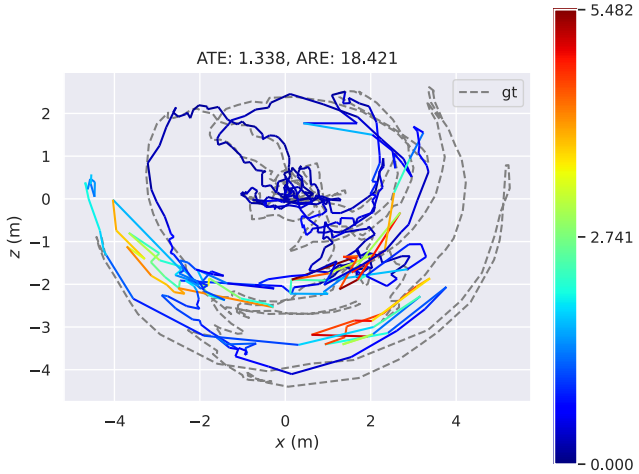
In order to understand this higher error mitigation ability of FastVGGT, we further investigated it, and we found that the error mitigation effect comes from the first-frame anchoring design in FastVGGT. FastVGGT assigns all tokens in the first frame as `dst` tokens, referring to them as "Reference Tokens" to preserve their strong representativeness. However, we find that the crucial factor is to reduce tokens from subsequent frames that are highly similar to those in the first frame. As shown in Fig. 13d,13e,13f, by adding first frame anchoring and allowing more temporal merging (so that more tokens from subsequent frames can be merged to the first frame), HTTM achieves a similar error mitigation effect to FastVGGT. We speculate that tokens from later frames can mislead the Global Attention module. Because these tokens are highly similar to first-frame tokens, the model may incorrectly treat them as part of the reference frame, weakening the coordinate anchor and amplifying drift. By explicitly designating all first-frame tokens as `dst` tokens and merging highly similar tokens from subsequent frames into them, the ambiguity is suppressed and the reference frame remains stable.

Note that in a large scene with long-sequence input, the tokens from the first frames consist less than 1% of the whole token set, so activating first-frame anchoring introduces negligible overhead for HTTM.

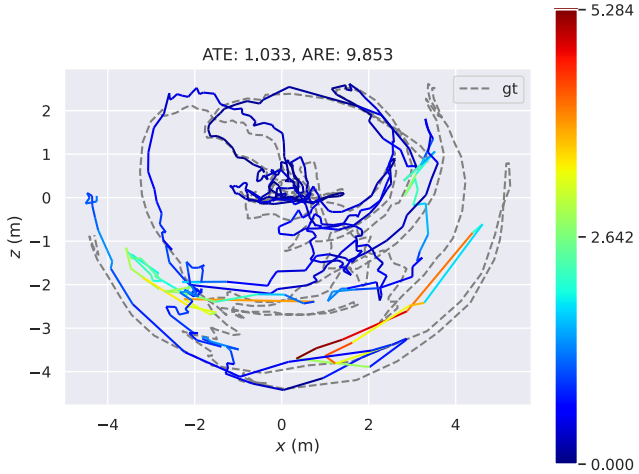


(a) FastVGGT

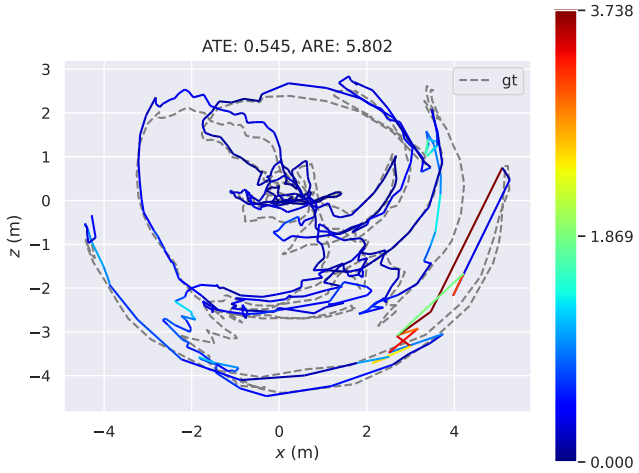
(b) VGGT



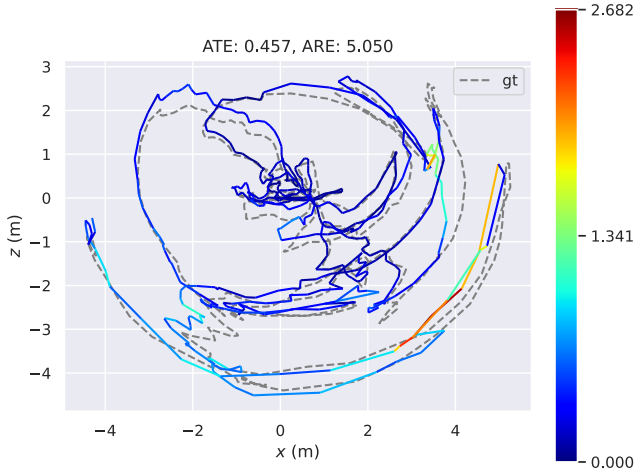
(c) HTTM without first frame anchoring



(d) HTTM* with 10 frames of temporal merging



(e) HTTM* with 30 frames of temporal merging



(f) HTTM* with 40 frames of temporal merging

Figure 13. Comparison of camera pose estimation performance between FastVGGT, VGGT, HTTM without first frame anchoring, and HTTM with first frame anchoring (denoted as HTTM*) across different numbers of temporal frames. Colors indicate the deviation from the ground-truth camera trajectory.

C. Algorithm Pseudo Codes

In this section, we provide the pseudo code of some algorithms in the hope of helping people who are interested in reimplementing our work.

C.1. HTTM Accelerated Global Attention

Algorithm 1 HTTM: Global Attention with Temporal Reordering and Blockwise Token Merging

Require: Input tokens $\boldsymbol{X} \in \mathbb{R}^{L\times D}$, query cluster number k_q, key cluster number k_k, spatial block size B_s, temporal block size B_t, query outlier quantile γ_q, key outlier quantile γ_k

Ensure: Output tokens $\boldsymbol{O} \in \mathbb{R}^{L\times D}$

Compute multi-head projections $\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V} \in \mathbb{R}^{H\times L\times E}$ from $\boldsymbol{X}$
$B \leftarrow B_s \cdot B_t$
Compute temporal reordering index $\pi \leftarrow$ TEMPORALREORDER(L, B_s, B_t)
$\tilde{\boldsymbol{Q}} \leftarrow$ REORDER$(\boldsymbol{Q}, \pi)$, $\tilde{\boldsymbol{K}} \leftarrow$ REORDER$(\boldsymbol{K}, \pi)$, $\tilde{\boldsymbol{V}} \leftarrow$ REORDER$(\boldsymbol{V}, \pi)$
$\boldsymbol{C_q} \leftarrow$ BLOCKWISETOME$(\tilde{\boldsymbol{Q}}, B, k_q)$
$\boldsymbol{C_k} \leftarrow$ BLOCKWISETOME$(\tilde{\boldsymbol{K}}, B, k_k)$
$\boldsymbol{Q}_m, \boldsymbol{M_q} \leftarrow$
    BLOCKAGGREGATEQUANTILE$(\tilde{\boldsymbol{Q}}, \boldsymbol{C_q}, B, k_q, \gamma_q)$
$\boldsymbol{K}_m \leftarrow$ BLOCKAGGREGATE$(\tilde{\boldsymbol{K}}, \boldsymbol{C_k}, B, k_k)$
$\boldsymbol{V}_m \leftarrow$ BLOCKAGGREGATE$(\tilde{\boldsymbol{V}}, \boldsymbol{C_k}, B, k_k)$
$\boldsymbol{O}_m \leftarrow$ ATTENTION$(\boldsymbol{Q}_m, \boldsymbol{K}_m, \boldsymbol{V}_m)$
$\tilde{\boldsymbol{O}} \leftarrow$ UNMERGEBYCLUSTER$(\boldsymbol{O}_m, \boldsymbol{C_q}, \boldsymbol{M_q})$
$\tilde{\boldsymbol{O}}[\boldsymbol{M_q}] \leftarrow$ ATTENTION$(\tilde{\boldsymbol{Q}}[\boldsymbol{M_q}], \boldsymbol{K}_m, \boldsymbol{V}_m)$
$\boldsymbol{O} \leftarrow$ INVERSEREORDER$(\tilde{\boldsymbol{O}}, \pi)$
$\boldsymbol{O} \leftarrow$ OUTPUTPROJ$(\boldsymbol{O})$
return $\boldsymbol{O}$

C.2. Adaptive Outlier Filtering

Algorithm 2 BLOCKAGGREGATEQUANTILE: Blockwise Token Aggregation with Quantile-based Outlier Filtering

Require: Tokens $\boldsymbol{Q} \in \mathbb{R}^{H\times L\times E}$, cluster assignments $\boldsymbol{C_q} \in \mathbb{Z}^{H\times L}$, block size B, cluster number k, quantile γ

Ensure: Aggregated tokens $\boldsymbol{Q_c}$, outlier mask $\boldsymbol{M} \in \{True, False\}^{H\times L}$

Partition $\boldsymbol{Q}$ and $\boldsymbol{C_q}$ into $H \times \lceil L/B \rceil$ blocks
Initialize centroid tensor $\boldsymbol{Q_c}$, error tensor $\boldsymbol{\varepsilon} \in \mathbb{R}^{H\times L}$, and outlier mask $\boldsymbol{M} \leftarrow$ False
for each block i **do**
    Compute cluster centroids $\boldsymbol{Q_c}^{(i)}$ from $\boldsymbol{Q}^{(i)}$ using $\boldsymbol{C_q}^{(i)}$
    for each token j in block i **do**
        $c \leftarrow \boldsymbol{C_q}^{(i)}[j]$
        if $c \geq 0$ **then**
            $\boldsymbol{\varepsilon}^{(i)}[j] \leftarrow \|\boldsymbol{Q}^{(i)}[j] - \boldsymbol{Q_c}^{(i)}[c]\|_2$
        else
            $\boldsymbol{\varepsilon}^{(i)}[j] \leftarrow 0$
        end if
    end for
end for
$\tau \leftarrow$ QUANTILE$(\boldsymbol{\varepsilon}, \gamma)$
for each block i **do**
    Mark tokens with $\boldsymbol{\varepsilon}^{(i)}[j] > \tau$ as outliers: $\boldsymbol{C_q}^{(i)}[j] \leftarrow -1$, $\boldsymbol{M}^{(i)}[j] \leftarrow$ True
    Recompute only the centroids of clusters affected by outlier removal
    Append final centroids and retained outliers of block i to $\boldsymbol{Q_c}$
end for
return $\boldsymbol{Q_c}, \boldsymbol{M}$

References

[1] Dejan Azinović, Ricardo Martin-Brualla, Dan B Goldman, Matthias Nießner, and Justus Thies. Neural RGB-D Surface Reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6290–6301, 2022. 3, 7

[2] Daniel Bolya and Judy Hoffman. Token Merging for Fast Stable Diffusion. *CVPR Workshop on Efficient Deep Learning for Computer Vision*, 2023. 1, 2, 3, 4

[3] Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. Token Merging: Your ViT But Faster. In *The Eleventh International Conference on Learning Representations*, 2023. 1, 3, 4

[4] Mengzhao Chen, Wenqi Shao, Peng Xu, Mingbao Lin, Kaipeng Zhang, Fei Chao, Rongrong Ji, Yu Qiao, and Ping Luo. Diffrate: Differentiable Compression Rate for Efficient Vision Transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 17164–17174, 2023. 1

[5] Xuanyao Chen, Zhijian Liu, Haotian Tang, Li Yi, Hang Zhao, and Song Han. Sparsevit: Revisiting activation sparsity for efficient high-resolution vision transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2061–2070, 2023. 2

[6] Xingyu Chen, Yue Chen, Yuliang Xiu, Andreas Geiger, and Anpei Chen. TTT3R: 3D Reconstruction as Test-Time Training. *arXiv preprint arXiv:2509.26645*, 2025. 2

[7] Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. Generating Long Sequences with Sparse Transformers. *arXiv preprint arXiv:1904.10509*, 2019. 1, 2

[8] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2017. 3

[9] Tri Dao. FlashAttention-2: Faster Attention with Better Parallelism and Work Partitioning. In *International Conference on Learning Representations (ICLR)*, 2024. 7

[10] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pages 4171–4186, 2019. 3

[11] Zhanzhou Feng, Jiaming Xu, Lei Ma, and Shiliang Zhang. Efficient Video Transformers via Spatial-temporal Token Merging for Action Recognition. *ACM Transactions on Multimedia Computing, Communications and Applications*, 20 (4):1–21, 2024. 3

[12] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, and et al. The llama 3 herd of models, 2024. 2

[13] Jeongseok Hyun, Sukjun Hwang, Su Ho Han, Taeoh Kim, Inwoong Lee, Dongyoon Wee, Joon-Young Lee, Seon Joo Kim, and Minho Shim. Multi-Granular Spatio-Temporal Token Merging for Training-Free Acceleration of Video LLMs. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 23990–24000, 2025. 3

[14] Minchul Kim, Shangqian Gao, Yen-Chang Hsu, Yilin Shen, and Hongxia Jin. Token Fusion: Bridging the Gap between Token Pruning and Token Merging. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1383–1392, 2024. 1, 3

[15] Nikita Kitaev, Lukasz Kaiser, and Anselm Levskaya. Reformer: The Efficient Transformer. In *International Conference on Learning Representations*, 2020. 1

[16] Seon-Ho Lee, Jue Wang, Zhikang Zhang, David Fan, and Xinyu Li. Video Token Merging for Long-form Video Understanding. In *Proceedings of the 38th International Conference on Neural Information Processing Systems*, Red Hook, NY, USA, 2024. Curran Associates Inc. 3

[17] Vincent Leroy, Yohann Cabon, and Jerome Revaud. Grounding Image Matching in 3D with MASt3R, 2024. 2

[18] Hao Liu, Matei Zaharia, and Pieter Abbeel. Ring attention with blockwise transformers for near-infinite context. *arXiv preprint arXiv:2310.01889*, 2023. 2

[19] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-Resolution Image Synthesis with Latent Diffusion Models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. 3

[20] Thomas Schöps, Johannes L. Schönberger, 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 *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 7

[21] You Shen, Zhipeng Zhang, Yansong Qu, and Liujuan Cao. FastVGGT: Training-Free Acceleration of Visual Geometry Transformer. *arXiv preprint arXiv:2509.02560*, 2025. 1, 2, 4, 7, 3

[22] Jamie Shotton, Ben Glocker, Christopher Zach, Shahram Izadi, Antonio Criminisi, and Andrew Fitzgibbon. Scene Coordinate Regression Forests for Camera Relocalization in RGB-D Images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2930–2937, 2013. 7

[23] Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. RoFormer: Enhanced Transformer with Rotary Position Embedding. *Neurocomputing*, 568: 127063, 2024. 3, 1

[24] Apoorv Vyas, Angelos Katharopoulos, and François Fleuret. Fast Transformers with Clustered Attention. *Advances in Neural Information Processing Systems*, 33:21665–21674, 2020. 1

[25] Chung-Shien Brian Wang, Christian Schmidt, Jens Piekenbrinck, and Bastian Leibe. Faster VGGT with Block-Sparse Global Attention, 2025. 2

[26] Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. VGGT: Visual Geometry Grounded Transformer. In *Proceedings of*

the Computer Vision and Pattern Recognition Conference, pages 5294–5306, 2025. 1, 2, 3, 7

[27] Norman P. Jouppi Wang, Cliff Young, and David Patterson. BFloat16: The secret to high performance on Cloud TPUs. In *Google White Paper*, 2019. 7

[28] Qianqian Wang, Yifei Zhang, Aleksander Holynski, Alexei A Efros, and Angjoo Kanazawa. Continuous 3D Perception Model with Persistent State. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 10510–10522, 2025. 2, 7

[29] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. DUSt3R: Geometric 3D Vision Made Easy. In *CVPR*, 2024. 2

[30] Weitian Wang, Rai Shubham, Cecilia De La Parra, and Akash Kumar. Mixa-q: Revisiting activation sparsity for vision transformers from a mixed-precision quantization perspective. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22143–22152, 2025. 2

[31] Zhenhailong Wang, Senthil Purushwalkam, Caiming Xiong, Silvio Savarese, Heng Ji, and Ran Xu. DyMU: Dynamic Merging and Virtual Unmerging for Efficient VLMs. *arXiv preprint arXiv:2504.17040*, 2025. 1

[32] Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient Streaming Language Models with Attention Sinks. In *The Twelfth International Conference on Learning Representations*, 2024. 1

[33] Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, et al. Big Bird: Transformers for Longer Sequences. *Advances in neural information processing systems*, 33:17283–17297, 2020. 1, 2

[34] Yuan Zhang, Chun-Kai Fan, Junpeng Ma, Wenzhao Zheng, Tao Huang, Kuan Cheng, Denis Gudovskiy, Tomoyuki Okuno, Yohei Nakata, Kurt Keutzer, et al. SparseVLM: Visual Token Sparsification for Efficient Vision-Language Model Inference. In *International Conference on Machine Learning*, 2025. 2