

Co-Me: Confidence Guided Token Merging for Visual Geometric Transformers

co-me-tokens.github.io

Yutian Chen^{1,2}

yutianchen@andrew.cmu.edu

Yuheng Qiu¹

yuhengq@andrew.cmu.edu

Ruogu Li¹

ruoguli@andrew.cmu.edu

Ali Agha²

ali@fieldai.com

Shayegan Omidshafiei²

shayegan@fieldai.com

Jay Patrikar²

jay@fieldai.com

Sebastian Scherer^{1,2}

basti@fieldai.com

¹Carnegie Mellon University

²Field AI

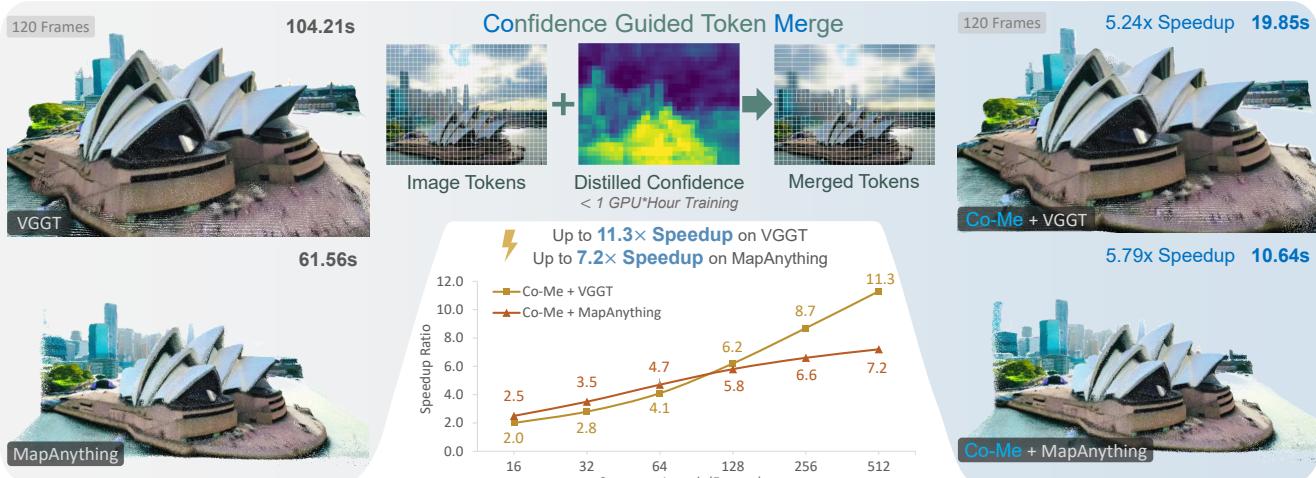


Figure 1. **Co-Me** accelerates visual geometric transformers by selectively merging low-confidence tokens guided by a distilled confidence predictor. When applied to VGGT and MapAnything, Co-Me achieves up to $11.3\times$ and $7.2\times$ speedup without retraining or architectural changes to the ViT backbone, turning geometric transformers into real-time-capable models for 3D perception.

Abstract

We propose *Confidence-Guided Token Merging (Co-Me)*, an acceleration mechanism for visual geometric transformers without retraining or finetuning the base model. Co-Me distilled a light-weight confidence predictor to rank tokens by uncertainty and selectively merge low-confidence ones, effectively reducing computation while maintaining spatial coverage. Compared to similarity-based merging or pruning, the confidence signal in Co-Me reliably indicates regions emphasized by the transformer, enabling substantial acceleration without degrading performance. Co-Me applies seamlessly to various multi-view and streaming visual geometric transformers, achieving speedups that scale with sequence length. When applied to VGGT and MapAnything, Co-Me achieves up to $11.3\times$ and $7.2\times$ speedup, making visual geometric transformers practical for real-time 3D perception and reconstruction.

1. Introduction

Reasoning about 3D structure from visual input serves as a fundamental capability for intelligent systems, such as au-

tonomous navigation, robotic manipulation, and augmented reality. Recent breakthroughs in visual geometry models, exemplified by Visual Geometry Grounded Transformers (VGGT) [35] and MapAnything [12], have demonstrated remarkable progress in geometric reconstruction and scene understanding tasks. However, these advances come at a significant computational cost: Vision Transformers (ViTs) incur quadratic complexity with respect to the input sequence length. This severely limits real-time deployment in resource-constrained environments. To empower embodied intelligence with fast and accurate 3D reconstruction, there is a critical need for acceleration methods that preserve geometric understanding while reducing computational cost.

The main barrier of efficient ViT inference lies in the quadratic time complexity $O(n^2d)$ of attention with respect to the number of tokens n and feature dimension d [33]. Although efficient attention mechanisms like FlexAttention [9] reduce memory overhead, their computational complexity remains high. Other than the attention, the multi-layer perceptron (MLP) also takes a considerable amount of computation [14]. This motivates token pruning and merg-

ing, which directly reduces the number of tokens to mitigate quadratic cost while maintaining similar performance.

A major line of work, exemplified by DynamicViT [25], progressively removes uninformative tokens to accelerate ViT inference. However, such approaches are mainly effective for inherently sparse tasks like image classification [25, 40, 42] or segmentation [30], as discarding tokens in dense geometric tasks often eliminate contextual cues required by accurate 3D reconstruction. Additionally, these methods need costly retraining, which is impractical as model and dataset scales grow, e.g., foundation model like VGGT already approach a billion parameters [35].

Compared to token pruning, token merging offers a more balanced acceleration strategy; however, existing practices mostly rely on heuristics. For instance, ToMe [3] merges tokens according to feature similarity for image classification, while FastVGGT [28] leverages both feature norms and cosine similarity to guide the merging process in the global attention operator. These methods are effective in scenarios with extremely long inputs, such as a 1,000-image sequence [28]. However, in real-world settings, the speedups are modest because global attention only constitutes a small fraction of the runtime when using efficient attention.

Inspired by human foveal vision [29], where high-acuity processing targets key regions while peripheral areas are coarsely perceived, we aim to reduce computation without sacrificing 3D reconstruction fidelity. A key observation is that most image tokens in ViTs do not actively contribute to 3D reconstruction. We further find that the high-confidence region predicted by the network strongly correlates with the region that the ViT emphasizes. In contrast, low-confidence regions often correspond to background, which provides coarse contextual cues rather than precise geometric estimation. Moreover, these regions exhibit poor quality and are typically discarded by downstream tasks like 3D reconstruction [8, 12, 35] or visual SLAM [24]. These findings raise a fundamental question: **How can we identify and reduce redundant tokens in visual geometry transformers without compromising geometric fidelity?**

In this work, we propose a novel *confidence-guided token merging* approach for accelerating visual geometric transformers. We observed that the confidence jointly estimated from the visual geometric model suggests the necessary information for scene geometric understanding. Based on this observation, we distilled a confidence module that predicts per-patch confidence rankings in a self-supervised manner. Guided by the distilled confidence, our method selectively merges low-confidence tokens, reducing computation in both the attention and MLP without sacrificing reconstruction quality. Compared to existing methods, our method preserves high-fidelity results in geometrically critical areas while significantly reducing the inference time. This efficiency further enables practical on-device deploy-

ment: when integrated with MapAnything, our accelerated model runs on an NVIDIA Jetson Thor at 3.5FPS with chunked 4-frame input and is $1.5\times$ faster than the original model, showing its suitability for real-time deployment.

The main contributions of this work are:

- We propose a novel confidence-guided token merging method that selectively merges tokens in low-confidence regions, delivering significant acceleration without re-training or architectural changes to the foundation model.
- We introduce a self-supervised confidence distillation module that estimates per-patch confidence rankings from intermediate encoder features to guide token merging.
- Experiments show that our method produces consistent speedup across various visual geometry transformers and input conditions with minimal performance degradation.
- We implement an efficient CUDA kernel for the proposed method to minimize the runtime overhead. We further deployed and validated our method on the edge device.

2. Related Works

2.1. Visual Geometric Transformer and Confidence

Visual geometric transformers have revolutionized geometric understanding by enabling single-pass 3D reconstruction without iterative optimization. DUSt3R [37] first demonstrated the pairwise 3D point-map regression, while MAS3R [18] incorporated explicit confidence modeling to improve geometric reliability. Recent extensions, including MUSt3R [5], Spann3R [34], CUT3R [36], and more [10, 11, 16, 20, 39] further generalize this paradigm to multi-view and streaming settings. Built upon these developments, VGGT [35] and MapAnything [12] unify camera pose, intrinsics, depth, and point-map prediction within a 1B-parameter ViT, thereby culminating the feed-forward paradigm and achieving state-of-the-art 3D reconstruction.

Crucially, these visual geometric transformers inherently predict confidence maps that quantify the reliability of their predictions [12, 35]. High-confidence regions typically correspond to well-textured, geometrically stable areas where multi-view cues are consistent, while low-confidence regions emerge in occluded, textureless, or ill-posed zones such as sky or reflective surfaces [13, 23, 24]. Despite this rich confidence information, current models allocate uniform computation to all tokens, leading to inefficient inference where equal resources are allocated to both geometrically reliable and uncertain regions. Such uniform processing paradigm offers an opportunity for acceleration, given the quadratic complexity of ViT with respect to token count.

2.2. Token Pruning and Merging

Recognizing that token importance varies across different regions, many studies accelerate ViT inference by processing tokens non-uniformly, primarily in 2D vision tasks such

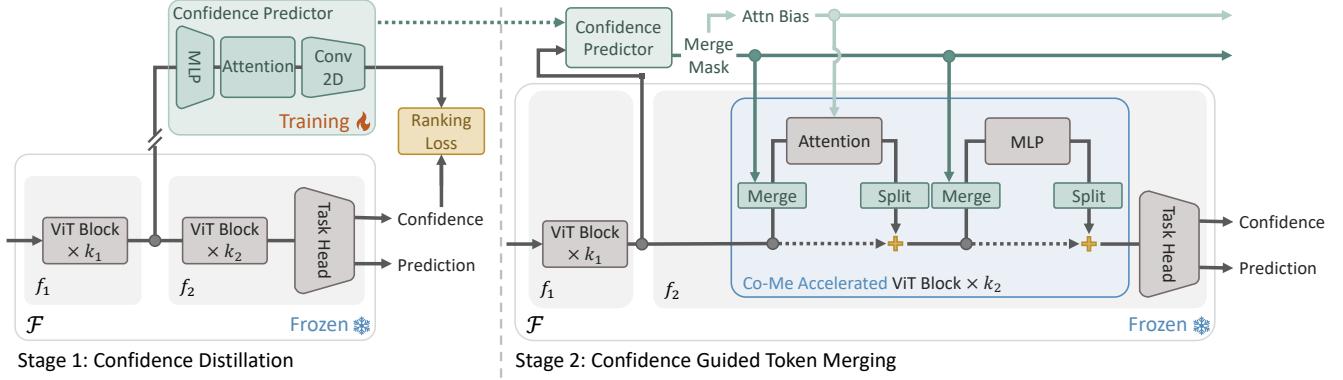


Figure 2. Overview of Co-Me. A lightweight module distilled from the frozen ViT backbone predicts per-token confidence from intermediate features. The predicted confidence is converted into a binary mask that guides token merging on the attention and MLP modules.

as image classification and segmentation. These approaches adapt the number of active tokens and generally fall into two categories: token pruning and token merging.

Token pruning methods dynamically drop tokens during inference to improve computation efficiency. Progressive pruning methods such as SparseViT [7], DynamicViT [25], and A-ViT [41] estimate token importance and selectively remove less significant tokens during forward passes, achieving substantial speedups in image classification. However, Liu et al. [19] revealed the fundamental limitation of token pruning on dense prediction tasks such as instance segmentation. Their study shows consistent performance degradation, which could only be partially alleviated by token reactivation strategies. Token pruning fundamentally suffers from spatial information loss, critical for maintaining spatial resolution and per-pixel consistency required by dense tasks like 3D reconstruction.

Token merging offers a promising alternative that aggregates similar tokens instead of discarding them. While this strategy preserves spatial coverage, it achieves a lower acceleration ratio due to the retained tokens. ToMe [3] pioneered training-free acceleration for off-the-shelf ViTs by merging tokens based on feature similarity, while advanced techniques like TokenLearner [26], PuMer [6], and ToFu [15] further refined the merging process for various applications. FastVGTT [28] introduces similarity and norm-based token merging in the global attention of visual geometric transformers, but its acceleration remains limited since it requires an extremely long input sequence (1000 frames) to yield a notable speedup. Our approach builds upon token merging while overcoming its limitations through two key innovations: a distilled confidence module that predicts per-token confidence to guide processing, and a confidence-guided merging strategy that preserves precision for 3D reconstruction with significantly less compute. This confidence-guided paradigm represents a novel intersection of geometric understanding and computational efficiency, accelerating visual geometric transformers with-

out finetuning the base ViT while addressing the unique demands of real-time 3D reasoning for embodied intelligence.

3. Method

Illustrated in Fig. 2, Co-Me contains two stages. In the first stage, we distill a light-weight confidence prediction module from the original ViT model (Sec. 3.1). In the second stage, we use the predicted per-patch confidence score to generate a merge mask during inference, and use this mask to guide the token merging and splitting (Sec. 3.2). Additionally, we employed several efficient implementations to minimize the overhead of token merging (Sec. 3.3).

3.1. Confidence Distillation

To avoid the dilemma of needing result of full inference to accelerate inference, token merging must rely on confidence estimates available *beforehand*. Since the encoder features already contain rich cues of confidence estimation, we distill a model that predicts per-token confidence from these features. This enables confidence-guided merging in the remaining part of the network. Formally, given a network \mathcal{F} that predicts a confidence map \mathcal{C} with input x , we split it into two parts $\mathcal{F} = f_2 \circ f_1$. The goal is to distill a lightweight network $f' : \text{Im } f_1 \mapsto \mathcal{C}'$ that estimates per patch confidence map \mathcal{C}' , which resembles \mathcal{C} on the token-level. Importantly, we never update or back-propagate through the visual geometric model \mathcal{F} ; all training is confined to the predictor f' .

Model Design The confidence predictor consists of three lightweight components. First, an MLP layer projects encoder features into a compact latent space. Next, a single-head attention captures interactions between patches across frames, enabling global reasoning at low computational cost. Finally, a Conv2D head compresses the tokens into a confidence map while suppressing spatial noise and promoting smoother predictions. Such a distilled module only adds less than 0.2% runtime compared to the full network.¹

¹Measured on VGGT under input sequence length of 128.

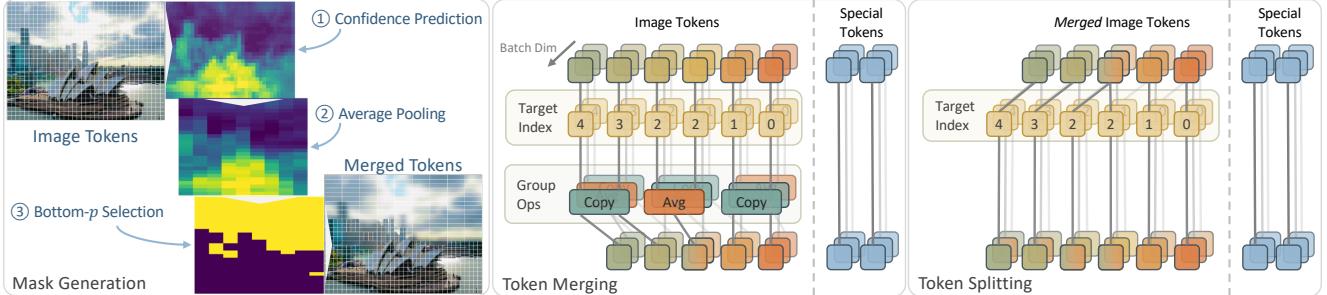


Figure 3. The proposed mask generation (left), merge (middle), and split (right) operators. Each sample generates an individual merge mask via confidence ranking and bottom- p selection. A shared index map is used by merging and splitting, which aggregate (average or copy) and restore image tokens while preserving special tokens. Our custom CUDA kernel implementation supports merging masks with different shapes across samples in the batch as long as the number of merged tokens remains consistent.

Since the confidence predictor has limited modeling capacity, we relax supervision so that it focuses on learning relative confidence ordering between tokens. This relaxation is valid because Co-Me only requires knowledge of the relative ordering to identify which tokens exhibit lower confidence than others. Specifically, we employed a logistic ranking loss [4] instead of the direct mean square error (MSE), and we define the loss function $\mathcal{L}(\mathcal{C}', \mathcal{C})$ as:

$$\mathcal{L}(\mathcal{C}', \mathcal{C}) = \frac{1}{|\mathcal{P}|} \sum_{(i,j) \in \mathcal{P}} \log(1 + \exp(\mathcal{C}'_j - \mathcal{C}'_i)), \quad (1)$$

where $\mathcal{P} = \{(i, j) \mid \text{Avg}(\mathcal{C})_i > \text{Avg}(\mathcal{C})_j\}$.

Where $\text{Avg}(\cdot)$ represents the average pooling that takes the average over all pixels in each patch as the patch’s overall confidence score. We further verify the advantage of this loss function over direct MSE loss in Sec. B.2.

Training Details Since the distillation aims to replicate the confidence of the original ViT model, the training is entirely self-supervised and does not rely on any ground-truth labels. For the experiments in this paper, we used the Tar-tanAir [38] dataset, a synthetic dataset containing more than 500,000 sequential images from diverse environments and motion patterns, for confidence distillation. The distillation converges within approximately 2,000 steps and takes less than an hour on a single NVIDIA A100 80G PCIe GPU².

For VGGT, StreamVGGT, and MapAnything, we insert the confidence predictor at the 15th layer of the encoder. This provides a good trade-off: layers too early yield inaccurate confidence, while layers too late reduce acceleration. The predictor generalizes to unseen data without finetuning.

3.2. Confidence-Guided Token Merging

The key insight is that merging low-confidence tokens barely affects the predicted geometry in high-confidence regions. Since we can receive a confidence prediction only af-

ter inference on f_1 , all operations described in this section are only applied to the remaining part f_2 of the network.

Mask Generation Illustrated in Fig. 3, we construct a binary merging mask under a predefined parameter merge ratio p from predicted per-token confidence scores. We first partition tokens along the spatial order into fixed-size groups of n consecutive image tokens. For each group, if the average confidence falls below the p -th percentile across all groups in this sample, it is marked for merging. This design ensures that all samples in the batch retain the same number of tokens after merging, thereby maintaining compatibility with efficient batched inference.

Token Merging Before each attention or MLP in f_2 , we apply token-merging to reduce the number of tokens and thereby accelerate inference. The design of the merge operator is shown in Fig. 3, where each group of image tokens is either averaged or preserved based on the merge flag. Formally, for a group of n tokens G_i , if the merge flag m_i is *true*, we replace the group with their average; otherwise, G_i remains unchanged. This operation is applied to all groups, and the results are concatenated into a contiguous tensor:

$$\text{MergeGrp}(G_i, m_i) = \begin{cases} \left\{ \frac{1}{n} \sum_{x \in G_i} x \right\} & \text{if } m_i \\ G_i & \text{otherwise} \end{cases} \quad (2)$$

$$\text{Merge}(\{G_i\}, \{m_i\}) = \text{Cat}(\{\text{MergeGrp}(G_i, m_i)\})$$

Token Splitting After the attention or MLP, the processed token sequence is restored to its original shape through a splitting step illustrated in Fig. 3. If a processed token group G'_i was not merged in the previous step, we copy it to its original position; otherwise, we replicate the merged token $G'_i = \{x\}$ for n times and place them at their original index.

$$\text{SplitGrp}(G'_i, m_i) = \begin{cases} \{x, \dots, x\} & \text{if } m_i \\ G'_i & \text{otherwise} \end{cases} \quad (3)$$

$$\text{Split}(\{G'_i\}, \{m_i\}) = \text{Cat}(\{\text{SplitGrp}(G'_i, m_i)\})$$

²We generate the input-confidence dataset beforehand and do not count inference time of acceleration target (e.g. VGGT) in the training time.



Figure 4. Effect of attention bias correction. Merging tokens distorts the weight distribution after softmax operator without correction (left). Adding a bias term $\log n$ aligns the merged attention distribution with the original distribution (right).

Such replication-based splitting, inspired by ToMeSD [2], allows Co-Me to reduce computation cost while maintaining compatibility with downstream prediction heads.

Attention Bias Correction As shown in Fig. 4, merging tokens into one concentrates multiple attention weights into a single entry. This causes the softmax operator to suppress that entry’s normalized attention weight and distorts the distribution. To counteract this, we introduce an *attention bias correction* that compensates for the merged entries. Specifically, for a merged group with n tokens and raw attention logit a_i , we add a bias term $\log n$ to get the corrected $\tilde{a}_i = a_i + \log n$. Since softmax is exponential, adding $\log n$ to the merged logit scales its weights by n and effectively restores the same total mass that n individual logits contributed before:

$$\text{softmax}(\tilde{a}_i) = \frac{e^{a_i + \log n}}{\sum_j e^{a_j}} \approx \sum_{k \in G_i} \frac{e^{a_k}}{\sum_j e^{a_j}}. \quad (4)$$

This correction realigns the post-softmax attention weights distribution with the original distribution. Although the attention bias introduces additional memory access and slows down the attention operator, our ablation in Sec. 5, H4. demonstrates that it significantly improves the performance.

3.3. Efficient Implementation

To minimize runtime overhead, we adopt several engineering optimizations. First, attention layers with non-zero bias terms are accelerated using FlexAttention [9], an efficient kernel that supports custom attention bias patterns. Second, given a binary merge mask, the mapping between original and merged token indices is fixed. We exploit this deterministic index relation to perform merge and split operations without the expensive `Cat` operator and communication between processes. This index computation is further accelerated with a single-pass exclusive scan in CUDA [21], and is reused across all subsequent merge-split operations.

4. Experiments

In this section, we demonstrate that our method accelerates both the state-of-the-art VGGT [35], its online streaming variant StreamVGGT [44], and MapAnything [12] with minimal performance change across three downstream tasks on NYUd-v2 [22], ETH3D [27], DTU-MVS [1], KITTI Depth [31], and RealEstate-10K [43]. These datasets span diverse domains, covering both indoor and outdoor scenes with varying motion, depth range, and length, thus providing a comprehensive evaluation of our method.

4.1. Experiment Setup

Baselines We evaluate Co-Me by applying it to VGGT, StreamVGGT, and MapAnything to measure speedup and performance on depth, pose, and point cloud prediction tasks. For fair comparison, we strengthen the VGGT baseline by replacing its naive attention with FlexAttention and incorporating the VRAM optimization trick from FastVGGT, yielding an enhanced version denoted as VGGT*. FlexAttention is used in both VGGT* and our acceleration for consistency and fairness. StreamVGGT and MapAnything already use an efficient fused attention, so we didn’t modify their implementation. For simplicity, we evaluate MapAnything with image-only input despite its ability to receive additional information as conditioning.

Both our method and FastVGGT use a hyperparameter in the range $(0, 1)$ to control the merge ratio. We set the merge ratio to 0.5 for Co-Me and 0.9 for FastVGGT, following the default value provided in its publicly released implementation. This setup is more favorable to the FastVGGT baseline, as a higher merge ratio allows more aggressive token reduction. We set the token group size n to 4 for all results in this paper to balance efficiency and performance.

Environment All experiments are conducted on a single NVIDIA A100 80G PCIe GPU and AMD EPYC 7543 CPU with one round of warmup before the benchmarking.

4.2. Depth Estimation

Depth estimation aims to predict a dense per-pixel depth map from one or multiple input images. We evaluate our method on both monocular and multi-view settings, following same input setup as the original models, respectively.

Metrics We employed L1 and $\delta_{1.25}$ as depth estimation metrics to capture the absolute and relative depth accuracy. Global scale alignment is applied to resolve the scale ambiguity. For all methods, we only evaluate regions where tokens are not merged by our method to ensure a fair comparison. Specifically, given the ground truth depth d and

Method	NYUd-v2 (1 Frame)				ETH3D (1 Frame)			
	Latency	Speedup	L1↓	$\delta_{1.25} \uparrow$	Latency	Speedup	L1↓	$\delta_{1.25} \uparrow$
VGGT	178.6	1.00×	0.106	0.969	178.0	1.00×	0.224	0.984
VGGT*	179.5	1.00×	0.106	0.969	178.8	1.00×	0.224	0.984
Fast VGGT	-	-	-	-	-	-	-	-
Ours	159.5	1.12×	0.106	0.970	161.4	1.10×	0.234	0.976

Table 1. Evaluation of Latency (ms), L1 depth error (m), and $\delta_{1.25}$ (unitless) of **monocular depth estimation** with scale alignment over NYUd-v2 and ETH3D datasets. FastVGGT does not support inferring single image.

Method	DTU-MVS (32 Frames)				KITTI Depth (48 Frames)			
	Latency	Speedup	L1↑ ↓	$\delta_{1.25} \uparrow$	Latency	Speedup	L1↓	$\delta_{1.25} \uparrow$
VGGT	8795	1.00×	0.744	0.990	17017	1.00×	0.512	0.996
VGGT*	4464	1.97×	0.744	0.990	7216	2.36×	0.512	0.996
Fast VGGT	4958	1.77×	0.802	0.990	7846	2.17×	0.548	0.996
Ours	3149	2.79×	0.788	0.990	4960	3.43×	0.566	0.994
StreamVGGT	205.8	1.00×	2.926	0.988	222.7	1.00×	1.829	0.822
Ours	174.4	1.18×	3.056	0.986	186.2	1.19×	1.849	0.828
MapAnything	2945	1.00×	4.893	0.981	4610	1.00×	0.700	0.987
Ours	1026	2.87×	5.012	0.980	1398	3.30×	0.762	0.986

↑ in unit cm to preserve the significant digits.

Table 2. Latency (ms), L1 depth error (m), and $\delta_{1.25}$ (unitless) of **multi-view depth estimation** with global scale alignment over DTU and KITTI depth datasets. Our method consistently delivers the best speedup with little performance change from the original models.

predicted depth \hat{d} , the metrics are defined as:

$$\begin{aligned} L_1 &= \frac{1}{N} \sum_{i=1}^N |\hat{d}_i - d_i|, \\ \delta_{1.25} &= \frac{1}{N} \sum_{i=1}^N \mathbf{1} \left[\max \left(\frac{\hat{d}_i}{d_i}, \frac{d_i}{\hat{d}_i} \right) < 1.25 \right], \end{aligned} \quad (5)$$

where $\mathbf{1}[\cdot]$ is the indicator function.

Monocular Depth Tab. 1 shows the results on NYUd-v2 and ETH3D. Our method achieves speedups of $1.1\times$ with nearly identical L1 and $\delta_{1.25}$ compared to VGGT. However, our accelerated MapAnything runs slightly slower than the original one due to additional memory accessing introduced by the token arrangement used in the MapAnything model.

Multi-view Depth Tab. 2 shows performance on DTU and KITTI. Our method yields a $2.79\times$ speedup on DTU (32 frames) and $3.43\times$ on KITTI (48 frames) compare to VGGT, with only minor accuracy loss. On StreamVGGT, we obtained a $1.2\times$ per-frame speedup with minimal performance degradation. Despite MapAnything’s smaller model, our method still achieves about $3\times$ speedup.

Analysis Speedup and accuracy retention vary across datasets. In KITTI, image tokens have less spatial overlap, so token merging causes larger information loss. In contrast, datasets like NYUd-v2 and DTU have more redundant fields of view, allowing our method to be more effective.

4.3. Pose Estimation

The pose estimation task seeks to predict the 6-DoF camera positions and orientations for all input views, providing a consistent frame for 3D reconstruction and understanding.

Method	DTU-MVS (32 Frames)				RealEstate-10K (128 Frames)			
	Latency	Speedup	AUC ^r ₃₀	AUC ^t ₃₀	Latency	Speedup	AUC ^r ₃₀	AUC ^t ₃₀
VGGT	8558	1.00×	0.810	0.786	101338	1.00×	0.996	0.952
VGGT*	4322	1.98×	0.807	0.783	30090	3.37×	0.996	0.962
Fast VGGT	4817	1.78×	0.805	0.793	27200	3.73×	0.995	0.945
Ours	3003	2.85×	0.806	0.783	18857	5.37×	0.995	0.950
StreamVGGT	207.1	1.00×	0.808	0.802	305.9	1.00×	0.996	0.953
Ours	175.4	1.18×	0.808	0.799	241.3	1.27×	0.994	0.937
MapAnything	2955	1.00×	0.855	0.869	41482	1.00×	0.996	0.951
Ours	1042	2.83×	0.850	0.856	8744	4.74×	0.993	0.939

Table 3. Latency (ms), Area Under Curve for relative rotation accuracy at 30deg (AUC_{30}^r) and relative translation accuracy at 30cm (AUC_{30}^t) for **pose estimation** with global Sim(3) alignment on DTU and RE10K.

Metrics Following VGGT [35], we employed area under curve for relative translation accuracy at 30cm (AUC_{30}^t) and relative rotary accuracy at 30° (AUC_{30}^r) as pose evaluation metrics. Sim(3) Umeyama alignment [17] is applied to remove scale and reference frame ambiguity. Specifically, the metrics are defined as follows:

$$\begin{aligned} AUC_{30}^t &= \int_0^{30} \frac{1}{|\mathcal{P}|} \sum_{i,j \in \mathcal{P}} \mathbf{1} [\|t_{i,j} - \hat{t}_{i,j}\|_2 < x] \, dx \\ AUC_{30}^r &= \int_0^{30} \frac{1}{|\mathcal{P}|} \sum_{i,j \in \mathcal{P}} \mathbf{1} [\angle(R_{i,j}^{-1} \hat{R}_{i,j}) < x] \, dx, \end{aligned} \quad (6)$$

where $\hat{t}_{i,j}$ and $t_{i,j}$ denote the predicted and ground-truth relative translation between frame i, j and $\hat{R}_{i,j}, R_{i,j}$ the corresponding rotations. $\angle(\cdot)$ measures the geodesic angle between rotations and \mathcal{P} includes all unordered camera pairs.

Camera Pose Tab. 3 reports the performance on DTU and RealEstate-10K³ (RE10K). Compared to VGGT, our method achieves $2.85\times$ and $5.37\times$ speedups on DTU (32 frames) and RE10K (128 frames) with minimal drop in AUC_{30}^t and AUC_{30}^r . Comparing to FastVGGT, our method attains higher acceleration with comparable pose accuracy.

Analysis We further analyze the variation in pose estimation accuracy across datasets. RE10K’s straight, hand-held trajectories allow easier relative pose estimation, while DTU’s SfM sampling trajectory introduces more diverse viewpoint shifts and featureless background making pose estimation more challenging.

4.4. Point Cloud Estimation

Point cloud estimation aims to reconstruct a dense and geometrically consistent 3D representation of the scene from multi-view inputs, serving as a core task for 3D vision.

Metrics We employed the completeness and accuracy from the Chamfer distance to evaluate the predicted point cloud. To remove global scale and reference-frame ambiguity, we apply a global Sim(3) Umeyama alignment [32] to align the predicted point cloud to the reference point cloud. Let P and G denote the predicted and ground-truth

³We only evaluate on the first 100 samples due to runtime limitation.

Method	DTU-MVS (32 Frames)				ETH3D (16 Frames)			
	Latency	Speedup	Comp. \downarrow	Acc. \downarrow	Latency	Speedup	Comp. \downarrow	Acc. \downarrow
VGGT	8739	1.00x	0.407	0.242	3190	1.00x	0.547	0.230
VGGT*	4470	1.95x	0.382	0.252	2076	1.54x	0.547	0.230
Fast VGGT	4926	1.77x	0.416	0.261	2357	1.35x	0.540	0.226
Ours	3124	2.79x	0.408	0.257	1538	2.07x	0.547	0.228
StreamVGGT	206.3	1.00x	0.338	0.402	197.3	1.00x	0.521	0.226
Ours	174.9	1.18x	0.342	0.413	171.2	1.15x	0.514	0.229
MapAnything	2902	1.00x	0.624	0.555	735.9	1.00x	0.558	0.202
Ours	1025	2.82x	0.650	0.584	362.3	2.03x	0.542	0.202

Table 4. Latency (ms), Completeness (cm) and Accuracy (cm) for **point cloud estimation** with global Sim(3) alignment on DTU and ETH3D datasets. On ETH3D, Co-Me outperforms VGGT while being $2.1\times$ faster.

point sets, respectively. After alignment, the completeness (Comp) and accuracy (Acc) are defined as following:

$$\text{Comp}(P, G) = \frac{1}{|G|} \sum_{g \in G} \min_{p \in P} \|g - p\|_2,$$

$$\text{Acc}(P, G) = \frac{1}{|P|} \sum_{p \in P} \min_{g \in G} \|p - g\|_2. \quad (7)$$

Similar to depth evaluation, for all methods, we only evaluate regions where tokens are not merged by our method.

Point Cloud Tab. 4 summarizes results on DTU and ETH3D. On DTU (32 frames), our method achieves a $2.79\times$ speedup, while maintaining essentially identical completeness and accuracy. On ETH3D (16 frames), we observe a $2.07\times$ speedup without losing completeness and at similar accuracy level as VGGT. For the online variant, our method accelerates StreamVGGT by $1.18\times$ per frame on DTU and $1.15\times$ per frame on ETH3D, with only minor changes in chamfer distance.

Analysis DTU’s redundant viewpoints enable efficient acceleration with minimal impact, whereas ETH3D shows *improved* performance for several accelerated models, as Co-Me removes low-confidence tokens that would otherwise introduce noise in wide-baseline reconstruction.

5. Analysis

We organize our analysis around six hypotheses to examine the efficiency and effectiveness of Co-Me. Together, they reveal how Co-Me achieves robust acceleration across various visual geometric models with little performance drop.

H1 The speedup of Co-Me scales with input size.

Fig. 5 illustrates the acceleration ratio over varying sequence lengths from 1 to 512, measured relative to the original VGGT baseline. Our method, when running on a merge ratio of 0.5, reaches up to $11.3\times$ on 512-frame sequences, consistently outperforming FastVGGT and VGGT*. When pushing the merge ratio to 0.9, we gain the astonishing $26.65\times$ speedup over the baseline. Notably, our approach also provides measurable acceleration even on single-frame inputs, where methods such as FastVGGT offer no benefit.

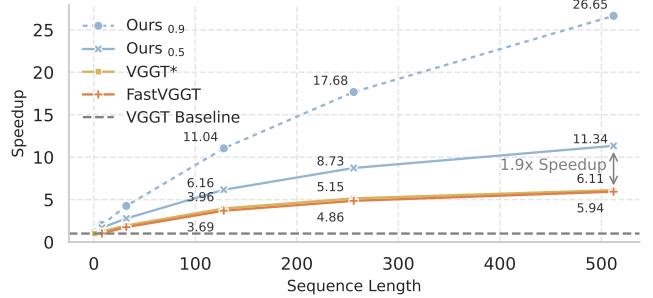


Figure 5. Acceleration ratio of Co-Me-accelerated VGGT across sequence lengths. The speedup increases with sequence length and reaches up to $26.65\times$ when using a higher merge ratio $p = 0.9$.

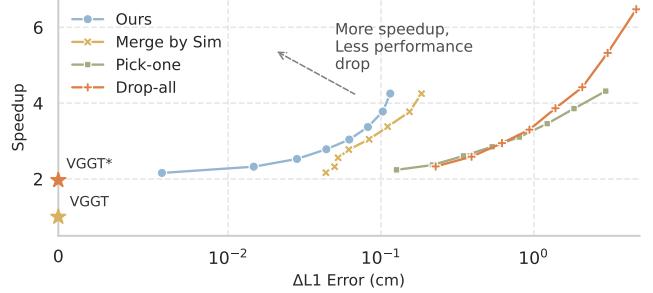


Figure 6. Performance v. Speedup trade off curves on multi-view depth estimation by various merging thresholds on the DTU-MVS dataset with 32 frames. Our method delivers the optimal speed-performance tradeoff.

fit. This demonstrates that the proposed acceleration mechanism is effective across different sequence lengths and scales favorably with longer inputs.

H2 Co-Me is better than similarity-based merging.

Prior approaches, such as ToMe [3] and ToMeSD [2], merge tokens based on cosine similarity without using confidence information. We therefore investigate how similarity based approaches perform in comparison to our method.

To this end, we establish a baseline method *Merge by Sim*, which uses token similarity instead of predicted confidence for merge mask. Specifically, tokens with average cosine similarity above $(1-p)$ -percentile threshold when the merging ratio is set to p . We plot the speedup-accuracy trade-off curve on DTU multi-view depth estimation task with $p \in [0.2, 0.9]$ in Fig. 6. Results indicate that our method yields a superior trade-off curve, achieving lower error increments than *Merge by Sim* at the same speedup.

H3 Merging is better than dropping or picking tokens.

We further explore whether averaging tokens offers a stronger coalescing scheme. We evaluate two alternatives to merging: 1) *Pick-one*, which randomly selects one token from each low-confidence group, and 2) *Drop-all*, which removes low-confidence tokens entirely. We present the speed-accuracy trade-off curve of these setups in Fig. 6. Results show that averaging via merging is significantly more robust, leading to over $10\times$ smaller performance degrada-

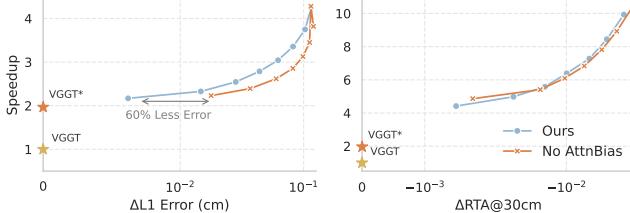


Figure 7. Adding attention bias correction improves performance across tasks. On DTU multi-view depth (left), our method reduces $\Delta L1$ error by $4\times$. On RE10K pose (right), the attention bias correction continuously offers a better performance-speed tradeoff.

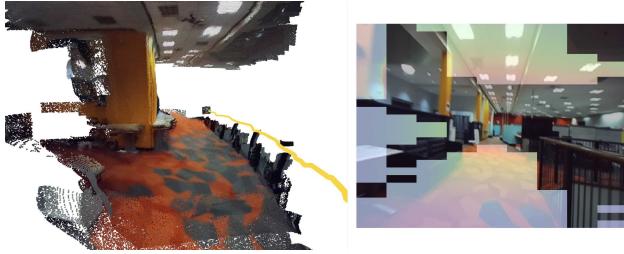


Figure 8. 3D reconstruction with camera trajectory (left) and predicted depth at un-merged regions (right) of Co-Me-accelerated MapAnything on edge device in real time. Despite the aggressive merging, the model produces consistent 3D reconstruction.

tion compared to the pick or drop variants. This reveals the important role of low-confidence tokens in visual geometric transformers for providing vague contextual information.

H4 Attention bias correction improves accuracy.

To demonstrate the necessity of attention bias correction, we conducted an ablation study by removing the bias correction term from Co-Me while keeping all other components identical. The evaluation was performed on multi-view depth estimation (DTU) and pose estimation (RE-10K). As shown in Fig. 7, excluding the bias term reduces the runtime overhead by avoiding additional memory access and element-wise addition, resulting in a slight speedup. However, this comes at the cost of a substantial performance degradation. The results highlight a clear trade-off: the attention bias correction introduces minor computational overhead but significantly improves the overall accuracy.

H5 Co-Me yields practical speedups on edge devices.

To assess real-world feasibility, we deploy MapAnything and our Co-Me-accelerated variant on an NVIDIA Jetson Thor with a Zed stereo camera. The system processes streaming inputs in fixed 4-image segments and registers each reconstruction in a global frame to emulate an online visual-odometry pipeline. As illustrated in Fig. 8, the system produces consistent 3D reconstructions and accurate depth over high-confidence, unmerged regions.

Under this setup, the accelerated model achieves a 3.5 FPS update rate, providing a $1.5\times$ speedup over the original MapAnything. This offers near real-time responsive-

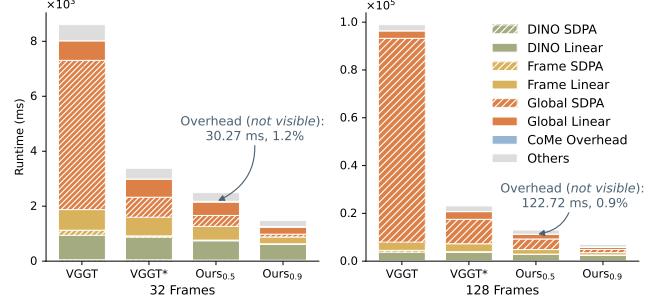


Figure 9. Runtime breakdown of VGGT, VGGT*, and Co-Me accelerated VGGT* with $p = \{0.5, 0.9\}$. When using efficient attention implementation, the MLPs account for a significant proportion of runtime in network inference. Co-Me can accelerate all modules in the network with minimal system overhead.

ness under edge compute platform, demonstrating that Co-Me delivers practical acceleration. Further runtime analysis and payload configuration details are provided in Sec. C.

H6 With efficient attention kernel, MLP become the new bottleneck for ViT acceleration.

Co-Me applies token merging not only on the scaled dot product attention (SDPA) but also the MLPs. We hypothesize that with an efficient attention implementation, the fraction of runtime attributable to the SDPA is significantly reduced, motivating the acceleration on MLPs.

In Fig. 9, we show the detailed runtime decomposition of VGGT, VGGT*, and Co-Me with merge ratio $p = \{0.5, 0.9\}$ and sequence lengths of 32 and 128. In vanilla VGGT, SDPA dominates the runtime. However, with the efficient SDPA, linear layers still take a considerable portion of inference time even under long, 128-frame input. This trend highlights MLP become the new bottlenecks as the SDPA is thoroughly optimized by previous works.

Moreover, the efficient implementation in Sec. 3.3 ensures Co-Me incurs negligible overhead in Fig. 9, accounting for $\approx 1\%$ of inference time on the accelerated VGGT.

6. Conclusion

We presented Confidence-Guided Token Merging (Co-Me), a novel method to accelerate visual geometric transformers by merging low-confidence tokens guided by a distilled confidence predictor without retraining or finetuning the base model. When applied on VGGT and MapAnything, Co-Me achieves up to $11.3\times$ and $7.2\times$ speedup while preserving accuracy in depth, pose, and point estimation. Furthermore, Co-Me is orthogonal to existing acceleration strategies and can be composed for additional speedups.

Future extensions of this work include supporting non-uniform batching for variable merge ratios, applying merging to time dimension in streaming input, and utilizing Co-Me in the training pipeline to improve training efficiency.

References

- [1] Henrik Aanæs, Rasmus Ramsbøl Jensen, George Vogiatzis, Engin Tola, and Anders BJORHOLM Dahl. Large-scale data for multiple-view stereopsis. *International Journal of Computer Vision*, pages 1–16, 2016. 5
- [2] Daniel Bolya and Judy Hoffman. Token merging for fast stable diffusion. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. IEEE, 2023. 5, 7
- [3] Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. Token merging: Your vit but faster, 2023. 2, 3, 7
- [4] Chris Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Greg Hullender. Learning to rank using gradient descent. In *Proceedings of the 22nd International Conference on Machine Learning*, page 89–96, New York, NY, USA, 2005. Association for Computing Machinery. 4
- [5] Yohann Cabon, Lucas Stoffl, Leonid Antsfeld, Gabriela Csurka, Boris Chidlovskii, Jerome Revaud, and Vincent Leroy. Must3r: Multi-view network for stereo 3d reconstruction, 2025. 2
- [6] Qingqing Cao, Bhargavi Paranjape, and Hannaneh Hajishirzi. Pumer: Pruning and merging tokens for efficient vision language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15365–15377. Association for Computational Linguistics, 2023. 3
- [7] Xuanyao Chen, Zhijian Liu, Haotian Tang, Li Yi, Hang Zhao, and Song Han. Sparsevit: Revisiting activation sparsity for efficient high-resolution vision transformer. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, page 2061–2070. IEEE, 2023. 3
- [8] 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. 2
- [9] Juechu Dong, Boyuan Feng, Driss Guessous, Yanbo Liang, and Horace He. Flex attention: A programming model for generating optimized attention kernels, 2024. 1, 5
- [10] Siyan Dong, Shuzhe Wang, Shaohui Liu, Lulu Cai, Qingnan Fan, Juho Kannala, and Yanchao Yang. Reloc3r: Large-scale training of relative camera pose regression for generalizable, fast, and accurate visual localization, 2025. 2
- [11] Wonbong Jang, Philippe Weinzaepfel, Vincent Leroy, Lourdes Agapito, and Jerome Revaud. Pow3r: Empowering unconstrained 3d reconstruction with camera and scene priors, 2025. 2
- [12] Nikhil Keetha, Norman Müller, Johannes Schönberger, Lorenzo Porzi, Yuchen Zhang, Tobias Fischer, Arno Knapitsch, Duncan Zauss, Ethan Weber, Nelson Antunes, Jonathon Luiten, Manuel Lopez-Antequera, Samuel Rota Bulò, Christian Richardt, Deva Ramanan, Sebastian Scherer, and Peter Kotschieder. Mapanything: Universal feed-forward metric 3d reconstruction, 2025. 1, 2, 5
- [13] Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision?, 2017. 2
- [14] Kyungmin Kim, Bichen Wu, Xiaoliang Dai, Peizhao Zhang, Zhicheng Yan, Peter Vajda, and Seon Kim. Rethinking the self-attention in vision transformers. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 3065–3069, 2021. 1
- [15] Minchul Kim, Shangqian Gao, Yen-Chang Hsu, Yilin Shen, and Hongxia Jin. Token fusion: Bridging the gap between token pruning and token merging, 2023. 3
- [16] Yushi Lan, Yihang Luo, Fangzhou Hong, Shangchen Zhou, Honghua Chen, Zhaoyang Lyu, Shuai Yang, Bo Dai, Chen Change Loy, and Xingang Pan. Stream3r: Scalable sequential 3d reconstruction with causal transformer, 2025. 2
- [17] Jim Lawrence, Javier Bernal, and Christoph Witzgall. A purely algebraic justification of the kabsch-umeyama algorithm. *Journal of Research of the National Institute of Standards and Technology*, 124, 2019. 6
- [18] Vincent Leroy, Yohann Cabon, and Jerome Revaud. Mast3r: Grounding image matching in 3d world, 2024. 2
- [19] Yifei Liu, Mathias Gehrig, Nico Messikommer, Marco Cannici, and Davide Scaramuzza. Revisiting token pruning for object detection and instance segmentation, 2024. 3
- [20] Jiahao Lu, Tianyu Huang, Peng Li, Zhiyang Dou, Cheng Lin, Zhiming Cui, Zhen Dong, Sai-Kit Yeung, Wenping Wang, and Yuan Liu. Align3r: Aligned monocular depth estimation for dynamic videos, 2024. 2
- [21] Duane Merrill and Michael Garland. Single-pass parallel prefix scan with decoupled lookback. In *Nvidia*, 2016. 5
- [22] Pushmeet Kohli, Nathan Silberman, Derek Hoiem, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012. 5
- [23] Matteo Poggi, Filippo Aleotti, Fabio Tosi, and Stefano Maticoccia. On the uncertainty of self-supervised monocular depth estimation, 2020. 2
- [24] Yuheng Qiu, Yutian Chen, Zihao Zhang, Wenshan Wang, and Sebastian Scherer. MAC-VO: Metrics-aware covariance for learning-based stereo visual odometry mac-vo. github. io. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3803–3814. IEEE, 2025. 2
- [25] Yongming Rao, Wenliang Zhao, Benlin Liu, Jiwen Lu, Jie Zhou, and Cho-Jui Hsieh. Dynamicvit: Efficient vision transformers with dynamic token sparsification, 2021. 2, 3
- [26] Michael S. Ryoo, AJ Piergiovanni, Anurag Arnab, Mostafa Dehghani, and Anelia Angelova. Tokenlearner: What can 8 learned tokens do for images and videos? In *Advances in Neural Information Processing Systems*, pages 13728–13741. Curran Associates, Inc., 2021. 3
- [27] 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. 5
- [28] You Shen, Zhipeng Zhang, Yansong Qu, and Liujuan Cao. Fastvggt: Training-free acceleration of visual geometry transformer, 2025. 2, 3

- [29] Emma E. M. Stewart, Matteo Valsecchi, and Alexander C. Schütz. A review of interactions between peripheral and foveal vision. *Journal of Vision*, 20(12):2–2, 2020. [2](#)
- [30] Quan Tang, Bowen Zhang, Jiajun Liu, Fagui Liu, and Yifan Liu. Dynamic token pruning in plain vision transformers for semantic segmentation, 2023. [2](#)
- [31] Jonas Uhrig, Nick Schneider, Lukas Schneider, Uwe Franke, Thomas Brox, and Andreas Geiger. Sparsity invariant cnns. In *International Conference on 3D Vision (3DV)*, 2017. [5](#)
- [32] S. Umeyama. Least-squares estimation of transformation parameters between two point patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(4):376–380, 1991. [6](#)
- [33] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023. [1](#)
- [34] Hengyi Wang and Lourdes Agapito. 3d reconstruction with spatial memory. *arXiv preprint arXiv:2408.16061*, 2024. [2](#)
- [35] Jianyuan Wang, Minghao Chen, Longfei Huang, and Xiaoming Liu. Vggt: Visual geometry grounded transformer, 2024. [1, 2, 5, 6](#)
- [36] Qianqian Wang, Yifei Zhang, Aleksander Holynski, Alexei A Efros, and Angjoo Kanazawa. Continuous 3d perception model with persistent state. *arXiv preprint arXiv:2501.12387*, 2025. [2](#)
- [37] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20697–20709, 2024. [2](#)
- [38] Wenshan Wang, Delong Zhu, Xiangwei Wang, Yaoyu Hu, Yuheng Qiu, Chen Wang, Yafei Hu, Ashish Kapoor, and Sebastian Scherer. Tartanair: A dataset to push the limits of visual slam, 2020. [4](#)
- [39] Jiani 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, 2025. [2](#)
- [40] Hongxu Yin, Arash Vahdat, Jose M. Alvarez, Arun Mallya, Jan Kautz, and Pavlo Molchanov. A-vit: Adaptive tokens for efficient vision transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10809–10818, 2022. [2](#)
- [41] Hongxu Yin, Arash Vahdat, Jose M. Alvarez, Arun Mallya, Jan Kautz, and Pavlo Molchanov. A-vit: Adaptive tokens for efficient vision transformer. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2022. [3](#)
- [42] Lu Yu and Wei Xiang. X-pruner: explainable pruning for vision transformers, 2023. [2](#)
- [43] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: learning view synthesis using multiplane images. *ACM Trans. Graph.*, 37(4), 2018. [5](#)
- [44] Dong Zhuo, Wenzhao Zheng, Jiahe Guo, Yuqi Wu, Jie Zhou, and Jiwen Lu. Streaming 4d visual geometry transformer. *arXiv preprint arXiv:2507.11539*, 2025. [5](#)

Supplementary Materials for

Co-Me: Confidence-Guided Token Merging for Visual Geometric Transformers

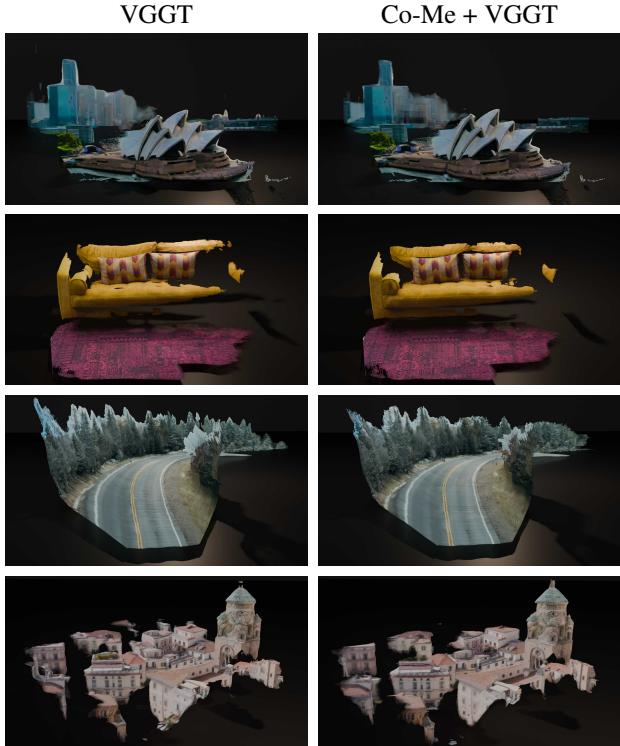


Figure 10. Qualitative comparison between VGGT (left) and Co-Me-accelerated VGGT (right). Best viewed digitally.

A. Qualitative Results

In this section we present qualitative results of Co-Me-accelerated VGGT and MapAnything. Specifically all results are created with the exact same configuration in Sec. 4 without finetuning or modification.

A.1. Success Cases

VGGT In Fig. 10, we show a qualitative comparison between VGGT (left) and Co-Me-accelerated VGGT (right) across eight representative scenes. Co-Me preserves the global scene structure and fine-grained geometry, including planar surfaces and prominent edges, despite operating with significantly fewer tokens. Minor differences appear primarily along the boundaries between high-confidence foreground regions and low-confidence background areas. These examples illustrate that confidence-guided merging maintains reconstruction fidelity with reduced computation.

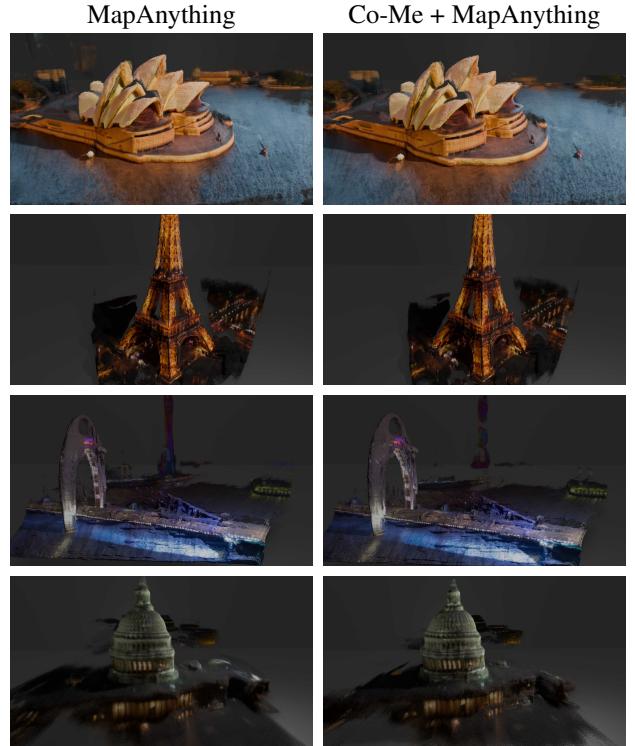


Figure 11. Qualitative comparison between MapAnything (left) and Co-Me-accelerated version (right). Best viewed digitally.

MapAnything Figure 11 reports qualitative reconstructions from MapAnything and its Co-Me-accelerated variant across four diverse outdoor scenes. Despite aggressive token reduction, the accelerated model retains the characteristic large-scale structure that MapAnything recovers—such as façade geometry, smooth water surfaces, and distant skyline contours. Most observable differences are confined to peripheral regions where texture cues are weak or depth ambiguity is intrinsic to the input views. In these areas, Co-Me may slightly simplify fine-scale geometry, but the dominant scene layout and salient landmarks remain stable. These results show that token merging integrates cleanly with the MapAnything pipeline, preserving its strong global consistency while reducing inference cost.

A.2. Failure Modes

Figure 12 highlights scenarios where Co-Me introduces noticeable degradation. In both examples, the lost geometry corresponds to thin, high-frequency structures that occupy

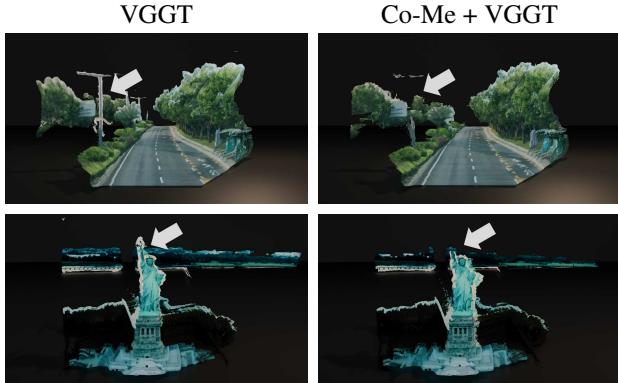


Figure 12. Failure cases of Co-Me-accelerated VGGT. Comparison between VGGT (left) and Co-Me-accelerated VGGT (right). Arrows indicate corrupted thin structures after token merging.

a small portion of the corresponding token. When these regions have low predicted confidence, merging discards their local resolution enough that the downstream decoder over-smooths the structure, causing incomplete reconstruction of the streetlight pole and the Statue of Liberty’s raised arm. While these elements do not affect the global scene layout, they reveal a limitation of confidence-guided merging in handling small or elongated objects.

B. Additional Experiments

B.1. Confidence Distillation Layer Ablation

To investigate where the confidence predictor should be inserted within the ViT backbone, we trained the predictor on features extracted from different encoder layers of VGGT under identical training setups. Fig. 13 illustrates the ranking loss curves for predictors attached to layers 6, 9, 12, 15, 18, and 21 respectively. We observe that the predictor distilled from layer 15 achieves the lowest ranking loss across all layers. Earlier layers (e.g., 6, 9) provide insufficient semantic and geometric cues, leading to noisy confidence estimates, while later layers (e.g., 18, 21) have stronger geometric reasoning but reduced token diversity, which limits generalization and causes slower convergence. Therefore, we use the layer-15 configuration in all experiments, as it provides an optimal trade-off between confidence ranking accuracy and computational overhead.

B.2. Confidence Distillation Loss Ablation

In the Sec. 3.1, we replace the MSE objective with a ranking loss that supervises the relative ordering of token confidences. To validate the effectiveness of loss formulation, we conduct an ablation by retraining the model using MSE under identical settings. We then compare the resulting predictions by measuring the intersection-over-union (IoU) between the top- p merge masks derived from the distilled pre-

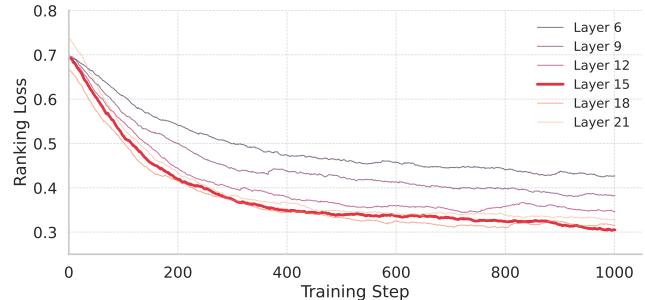


Figure 13. Distillation loss of confidence predictors distilled from various VGGT encoder layers. Layer 15 yields the lowest loss, indicating that mid-level encoder features contain the most information for confidence estimation. For readability, curves are smoothed with an exponential moving average with factor of 0.99.

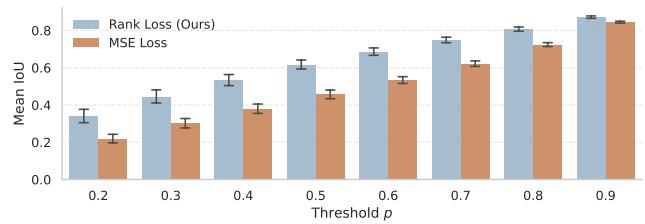


Figure 14. Confidence distillation with ranking loss achieves significantly higher IoU on DTU dataset than MSE loss. Error bar shows the 95% confidence interval.

dictor and those obtained from the full VGGT model, with $p \in [0.2, 0.9]$. The IoU metric is defined as:

$$\text{IoU} = \frac{|\mathcal{M}_{\text{pred}} \cap \mathcal{M}_{\text{gt}}|}{|\mathcal{M}_{\text{pred}} \cup \mathcal{M}_{\text{gt}}|}, \quad (8)$$

where $\mathcal{M}_{\text{pred}}$, \mathcal{M}_{gt} are the predicted and reference masks.

In Fig. 14, we can see that the ranking loss consistently outperforms MSE, demonstrating that supervising the relative ordering of confidences is more effective than regressing the confidence numerically for predicting merge masks.

B.3. Token Group Size Ablation

To evaluate the influence of token group size on the speed-accuracy trade-off, we tested Co-Me with group sizes of 2, 4, and 6 under identical merging ratios on DTU-MVS (32 frames). As shown in Fig. Fig. 15, smaller group sizes generally offer better accuracy retention for a given speedup, as they introduce finer-grained control and less information loss over which merged tokens. In contrast, larger groups provide stronger acceleration due to more aggressive token reduction, but incur slightly higher reconstruction error. Overall, group size 4 achieves the best balance between efficiency and accuracy and is therefore used for all experiments in Secs. 4 and 5.

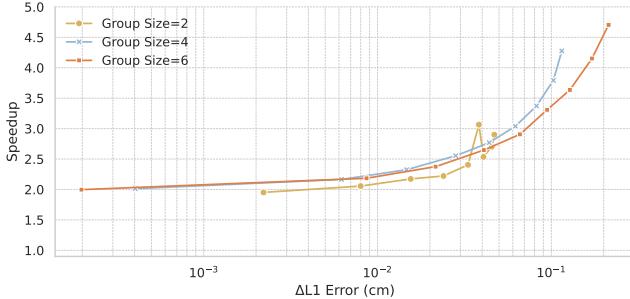


Figure 15. Speedup-accuracy trade-off of Co-Me-accelerated VGGT across various token group sizes on multi-view depth estimation (DTU-MVS, 32 frames). Smaller group sizes yield slightly better accuracy, while larger groups provide higher acceleration. Curves are plotted on a log-scaled error axis for clarity.

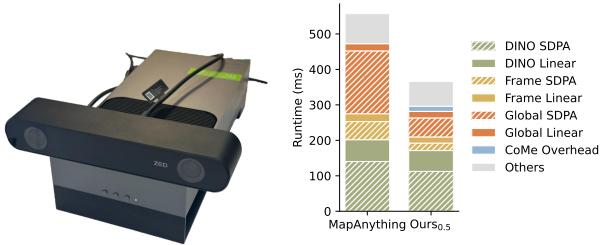


Figure 16. NVIDIA Jetson Thor and the Zed 2i stereo camera payload for real-world deployment test. We run MapAnything on chunks of 4 images and stack the results under the world coordinate frame to simulate a visual odometry.

C. Edge Compute Deployment

In Fig. 16, we illustrate the real-world deployment setup and runtime profile used to evaluate edge performance. An NVIDIA Jetson Thor runs MapAnything and our Co-Me-accelerated variant while receiving stereo input from a Zed 2i camera. The system groups incoming frames into fixed segments of four images and accumulates the resulting reconstructions in a global world coordinate frame, effectively simulating a streaming visual-odometry pipeline.

The stacked runtime bars in Fig. 16 decompose per-segment latency into DINO, frame-level, and global attention components, linear projections, Co-Me overhead, and other operations. Applying Co-Me shrinks the attention-dominated portions while adding only a small confidence-prediction cost, yielding an overall $1.5\times$ reduction in end-to-end runtime. On this platform, processing 4-image segments reaches 3.5 FPS, providing near real-time responsiveness under edge-compute constraints while preserving the stable 3D geometry observed in Sec. 5, H5.