DriveVGGT: Visual Geometry Transformer for Autonomous Driving

Xiaosong Jia[3,*], Yanhao Liu[1,2,*], Junqi You[1], Renqiu Xia[1], Yu Hong[1,2], Junchi Yan[1,†]
[1]Shanghai Jiao Tong University
[2]Shanghai Innovation Institute
[3]Institute of Trustworthy Embodied AI, Fudan University
[*]Equal Contributions [†]Corresponding authors

Abstract

Feed-forward reconstruction has recently gained significant attention, with VGGT being a notable example. However, directly applying VGGT to autonomous driving (AD) systems leads to sub-optimal results due to the different priors between the two tasks. In AD systems, several important new priors need to be considered: (i) The overlap between camera views is minimal, as autonomous driving sensor setups are designed to achieve 360° coverage at a low cost. (ii) The camera intrinsics and extrinsics are known, which introduces more constraints on the output and also enables the estimation of absolute scale. (iii) Relative positions of all cameras remain fixed though the ego vehicle is in motion.

To fully integrate these priors into a feed-forward framework, we propose DriveVGGT, a scale-aware 4D reconstruction framework specifically designed for autonomous driving data. Specifically, we propose a Temporal Video Attention (TVA) module to process multi-camera videos independently, which better leverages the spatiotemporal continuity within each single-camera sequence. Then, we propose a Multi-camera Consistency Attention (MCA) module to conduct window attention with normalized relative pose embeddings, aiming to establish consistency relationships across different cameras while restricting each token to attend only to nearby frames. Finally, we extend the standard VGGT heads by adding an absolute scale head and an ego vehicle pose head. Experiments show that DriveVGGT outperforms VGGT, StreamVGGT, fastVGGT on autonomous driving dataset while extensive ablation studies verify effectiveness of the proposed designs.

1. Introduction

4D reconstruction is a computer vision task that predicts geometric information from visual sensors [4, 31, 43]. Compared to other sensors, camera-based reconstruction has been widely researched and implemented as a low-cost solution in various fields, particularly in autonomous driving [24, 52**?**] and robotics [8, 28, 63]. Generally, there are two types of reconstruction forms. The first one is iterative-based methods, such as [7, 10]. These methods require selecting specific scenes or objects and proposing iterative reconstruction to achieve optimized results. However, due to the lack of sufficient generalization [49], iterative-based methods require retraining the model when scenes or objects are changed or modified. The second one is feed-forward methods [61]. These methods can directly output prediction results without updating any model parameters. The represented model, VGGT, can simultaneously predict 4 geometry tasks in various scenes, which marks a significant breakthrough.

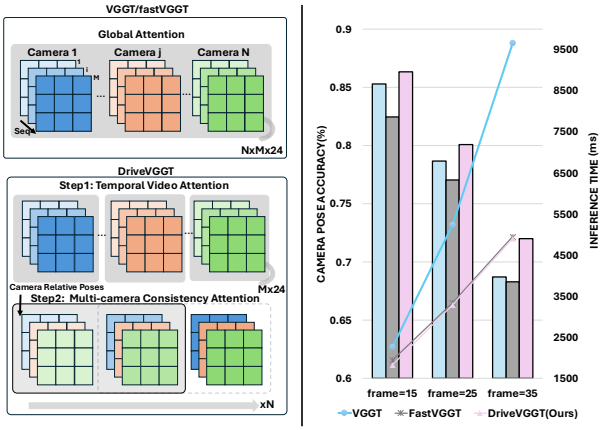

Figure 1. **Model comparison.** Our method can effectively implement multi-camera relative poses with faster speed and better performance.

Although feed-forward methods have achieved sufficient generalization, several limitations arise when implementing them in autonomous driving scenes [12, 17, 26, 45]. Initially, for cameras on self-driving vehicles, to keep a balance between FOV and cost [35], these cameras' views are always extremely different [16, 19, 44], and the captured images of each camera have low overlap [20, 53, 56]. Thus, it is not easy for a model to recognize similar features and eventually predict an effective image pose relationship [14, 55**?**]. Secondly, although the calibration of camera relative poses

is easy to obtain in autonomous driving systems [6, 18, 29], the camera relative poses cannot be directly implemented in feed-forward methods. Due to the scale difference between the predictions of feed-forward models and real-world relative poses, direct aggregation will cause scale ambiguity among geometry tokens [60]. Meanwhile, in most previous feed-forward architectures, each image token only contains one camera pose token [40], which means that the relative pose cannot be effectively represented.

To sufficiently aggregate camera relative poses in a multi-camera system [15, 21, 64], we propose a multi-camera visual geometry transformer with relative poses to achieve effective fusion between relative pose and geometry tokens from VGGT. The model comprises two components. Firstly, the Temporal Video Attention module is proposed to implement camera-level geometry aggregation among all cameras. As the video's temporal continuity of each camera, VGGT can effectively process the single-camera video into geometry tokens. The geometry tokens of each image are made of image pose tokens and depth tokens, which will be utilized to predict image pose and depth separately. However, the image pose tokens only represent the relationship between the current image and the first image. Therefore, to establish the pose relationship among all cameras on the vehicle, the Multi-camera Consistency Attention module is proposed to inject relative pose as extra pose tokens for each image. Specifically, we propose a relative pose embedding to normalize real-world camera poses and subsequently align them to the same dimension as geometry tokens. To implement interaction among images from different cameras, we utilize window attention to enhance adjacent multi-camera tokens sequentially. The proposed method outperforms other models on nuScenes dataset [2], which contains 6 low-overlap cameras on the vehicle. In detail, the proposed methods can achieve better reconstruction results with lower latency.

In conclusion, our contributions are as follows:

1) We propose DriveVGGT, a feed-forward framework, a feed-forward framework to achieve 4D reconstruction for multi-camera systems in autonomous driving. Compared to VGGT, DriveVGGT sufficiently incorporates data priors within AD systems and the unique settings of multi-camera systems. As a result, DriveVGGT achieves faster inference speed and higher prediction accuracy, enabling more efficient and reliable execution of various autonomous driving tasks.

2) We introduce an efficient two-stage pipeline to deal with multi-camera images. Specifically, we propose a Temporal Video Attention (TVA) module to process multi-camera videos independently, which better leverages the spatiotemporal continuity within each single-camera sequence. We propose a Multi-camera Consistency Attention (MCA) module to conduct window attention with normalized relative pose embeddings, establishing consistency relationships across different cameras while restricting each token to attend only to nearby frames.

3) Extensive experiments on the nuScenes dataset show the superiority of our proposed DriveVGGT, where it outperforms other VGGT-based methods in both inference speed and prediction accuracy.

2. Related Works

2.1. 4D Scene Reconstruction

4D Scene reconstruction is widely implemented in autonomous driving [50, 51, 54] and robotics[3, 32, 37]. In recent years, feed-forward reconstruction has been widely researched due to their powerful generalization and clear 3D geometry outputs. [41] proposes the first end-to-end 3D reconstruction pipeline to predict images' poses, intrinsics and depths simultaneously. Furthermore, [58] expands 3D reconstruction from static scenes to dynamic scenes; they additionally predict dynamic masks to achieve better performance for dynamic objects. Subsequently, [27] implements the learned movement map to recognize motion probability of various objects in seniors exhaustively. To simplify the pipeline of end-to-end reconstruction, VGGT[40] proposes a transformer-based feed-forward method to decode various geometric information from image tokens. After that, [65] achieves streaming reconstruction with VGGT by storing tokens in a memory cache, which can significantly enhance time inference with temporal causal attention. To further optimize VGGT, fastvggt[34] applies region-based random sampling to improve inference time, which is especially significant when processing with considerable(1000+) image inputs. Furthermore, [39] implements block-sparse global attention instead of a fully global one to improve model efficiency.

2.2. Temporal-Spatial Geometry Consistency

Monocular camera geometry consistency mainly focuses on temporal continuity [9, 13, 38, 48]. [46] utilizes static point clouds to improve the video generation consistency of the diffusion transformer. In the training process, [62] implements an M-in N-out architecture to recover missing images in temporal sequences. Meanwhile, [57] proposes a training-free pipeline to consistently generate high-quality novel views. [1] implements a multi-view synchronization module to control the geometry consistency between two different views. [23] implements mask attention to maintain consistency of the overlap region in the generation process. In the autonomous driving field, [30] introduces decoupled attention to achieve efficient interaction and eventually keep temporal-spatial geometry consistency among 6-view cameras.

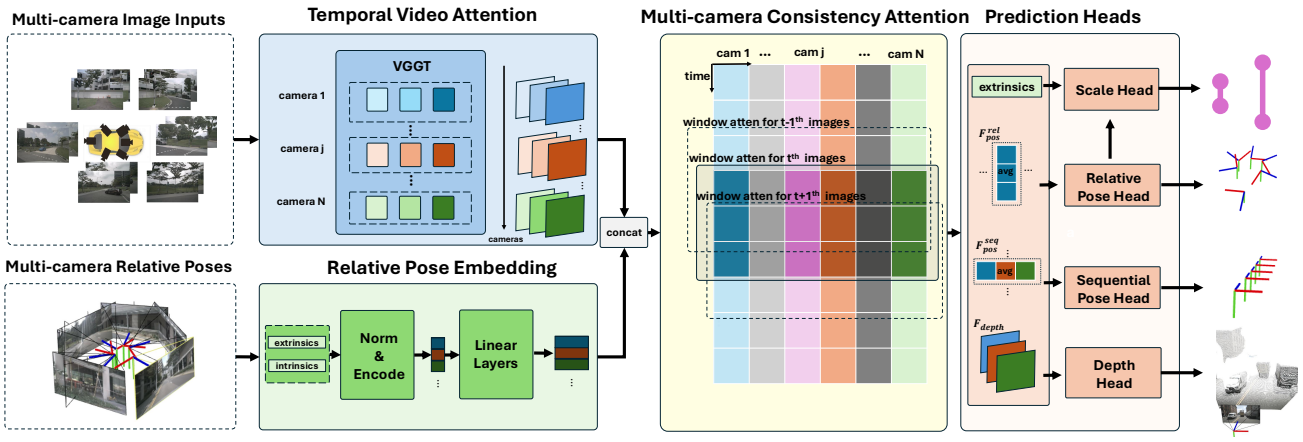

Figure 2. **Overview of DriveVGGT**. To inject camera relative poses to feed-froward reconstruction model, we introduce 3 sub-modules:1) Temporal Video Attention module is proposed to achieve video-level attention among images of each camera, and output sequential pose and geometry tokens initially;2) Relative Pose Embedding moudle is implemented to normalize real-world relative poses and pull them to the same dimensions of tokens;3) Multi-camera Consistency Attention Module apply window attention to achieve interaction among all cameras' images in limited sequential length. Finally, DriveVGGT can output 4 kinds of geometry tasks and complete scene reconstruction.

2.3. Position Geometry Representation

In typical scene reconstruction, estimating 6-D image poses is crucial for achieving better prediction results[11]. As to the representation of image position, [22] introduces a novel camera positional encoding method to represent both 4 DoF (object-centric) and 6 DoF camera poses. [47] implements spherical harmonics as positional encodings, which are specialized equivariant to represent relative poses. [33] explores a sufficient and effective transformer structure to deal with the specificity of position tokens. [25] proposes a novel relative positional encoding technique to process images similar to ROPE[36] in attention blocks. When it comes to recent reconstruction works, VGGT[40] adds extra image pose tokens to each image and finally decodes as image extrinsics and intrinsics. However, in VGGT, the first frame's camera pose needs to be initialized separately as a position reference for other frames, which may decrease prediction accuracy when feeding images to the model with different orders. Concerned the above problem, [42] introduce a fully permutation-equivariant architecture to eliminate this bias and propose different orders of images at the same level.

3. Proposed Approach

3.1. Overview

We propose DriveVGGT to fully utilize the information of relative camera poses to improve the model performance of geometry tasks, such as camera pose estimation and depth estimation. The model architecture is illustrated in Fig. 2. The model generally consists of three sub-modules. Firstly, the Temporal Video Attention (TVA) module is proposed to extract geometric features from each camera sequence, which contains sequence pose tokens to indicate the position relationship with the first frame of each video, and image tokens to indicate geometric features. Then, the Multi-camera Consistency Attention (MCA) module is proposed to implement multi-camera attention of adjacent images. To overcome the instability of low-overlapped images, we inject relative poses into the attention process to generate a unified geometry representation. Finally, the prediction heads decode the above features as the predictions of relative poses, sequence poses and depths separately.

3.2. Temporal Video Attention

Temporal Video Attention module is proposed to establish the initial geometry relationship among images captured by each camera. These images belong to a streaming video and are easy for feed-forward geometry models (like VGGT) to output effective reconstruction results. Specifically, as to N images, the simplest form of feed-forward geometry transformer is:

$$f : \{I_i\}_{i=1}^{N} \mapsto \{(F_{\text{pos}}(i), F_{\text{depth}}(i))\}_{i=1}^{N} \quad (1)$$

where $I(i)$ is the i-th image with resolution $H \times W$, $f(.)$ is transformer function to process these pictures to tokens. Subsequently, with the help of the decoder head, these tokens can be translated as parctical geometry information as:

$$\{(Depth(i), Geo(i), ...)\}_{i=1}^{N} = Head(f(\{I_i\}_{i=1}^{N})) \quad (2)$$

where $Depth_i$ is the prediction of the depth map, $Geo(i)$ is image extrinsics, which contain 6 dimensions to indicate the rotation and translation of the image in 3D world, and $F_{pos}(i), F_{depth}(i)$ are geometry tokens of each image.

In the case of multi-camera situations, unlike global attention in VGGT, Temporal Video Attention only implements attention for images captured by the same camera. For instance, as to M cameras which capture N images simultaneously, the function TVA module is:

$$TVA(.) = \{f(.)\}_{i=1}^{M} \tag{3}$$

which only aggregate features of each camera, and The outputs of TVA module is:

$$\{(F_{pos}^{seq}(i,j), F_{depth}(i,j))\}_{(i,j)} = TVA(\{I(i,j)\}_{j=1}^{N})) \tag{4}$$

where $F_{pos}^{seq}(i,j)$ indicates that the camera pose tokens only represent the sequential pose prediction results, which are aligned with the first image of each camera separately.

3.3. Relative Pose Embedding

Concerned with the uncertain scale for the final geometry outputs proposed by feed-forward visual geometry models, it is meaningful to pre-process relative poses among all cameras on the car or robot. Initially, to alleviate the scale difference between inputs and outputs, we normalize the translation (T) among all cameras as T_{norm} (mean = 0, std = 0.1).

Following the encoder methods of VGGT, we transform the intrinsic and extrinsic into a 10-D vector as:

$$P_{cam}(j) = Concat(T_{norm}(j), R(j), Intric(j)) \tag{5}$$

Concerned that the relative poses of cameras (num=M) on a self-driving vehicle are static at any time, we only need to process M camera poses. Then, we pull the P_{cam} to the same dimension of tokens from TVA module, and treat them as geometry information which indicates the relative pose relationship of all cameras on the vehicle as:

$$F_{pos}^{cam}(j) = Embed(P_{cam}(j)) \tag{6}$$

3.4. Multi-Camera Consistency Attention

The outputs of TVA module only implement attention among images from the same camera. However, there are two problems in this process. Firstly, the initial image pose of every camera video is set to the same position, which means the relative pose should be estimated to recover the camera pose in the global world. Secondly, the scale of each video is misaligned due to the attention isolation between each camera. To overcome the above problems, the Multi-camera Consistency Attention (MCA) module is proposed to get the unified reconstruction results. The MCA module is illustrated in detail in Fig. 3, which can achieve lower computational complexity for long-sequence attention.

3.4.1. Token Initialization

To optimize tokens from the TVA module, token initialization operation is proposed before implementing attention,

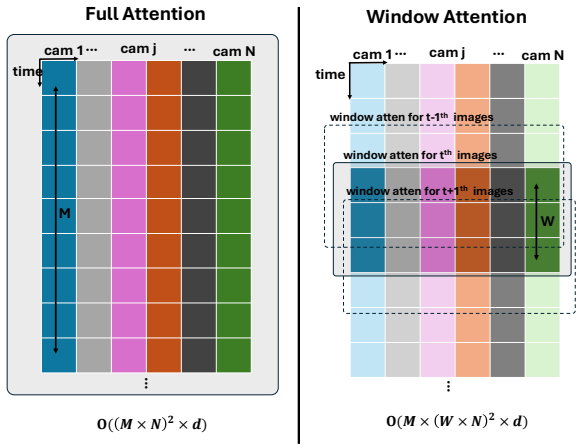

Figure 3. Illustration of window attention mechanism.

which aggregates relative pose tokens to initial tokens from the TVA module. Concerned that the following prediction heads merely utilize tokens from the selected 4 layers, we only extract and process the selected tokens in the MCA module. For each layer, we concat relative pose tokens from relative pose embedding module:

$$F_{token}(i,j) = Concat(F_{pos}^{cam}(j), F_{pos}^{seq}(i,j), F_{depth}(i,j)) \tag{7}$$

where i indicates frame index of each video, and j indicates the j-th camera on the vehicle. As the camera on the car is fixed, the relative camera pose of each frame is the same.

3.4.2. Window Attention

Unlike stream-based methods, the global reconstruction optimization can implement attention at all times, past or future[59]. In terms of long-sequence video reconstruction, global attention among all images is redundant and inefficient. Thus, we propose window attention to implement the attention operation on multi-camera images belonging to 3 adjacent time frames:

$$\{F(i,j)\}_{(i,j)} = Atten^{i}(\{\{F_{token}(i,j)\}_{1}^{M}\}_{i-1}^{i+1}) \tag{8}$$

where $Atten^{i}$ is the i-th global attention among (i-1)-th,i-th,(i+1)-th tokens. $F_i(i,j)$ is the final optimized i-th tokens. For N frame images of each camera sequence, the above attention operation implements N times.

Finally, after the above window attention, all tokens consist of 3 parts: relative pose tokens, sequential pose tokens, and image geometry tokens. Concerning the relative camera poses which are time-invariant, the MCA module finally outputs M relative poses and N sequential poses. Thus, we aggregate the sequential pose token as:

$$F_{agg}^{seq}(i,j) = Avg(\sum_{j=1}^{M} F^{seq}(i,j)) \tag{9}$$

and the relative pose token as:

$$F_{agg}^{rel}(i,j) = Avg(\sum_{j=1}^{N} F^{rel}(i,j)) \tag{10}$$

3.5. Prediction Heads

Camera Pose Head: Following the VGGT architecture, we reuse the camera head in VGGT to decode relative poses and sequential poses separately. Specifically, the Relative Pose Head outputs the time-invariant relative camera poses, which contain camera relative extrinsic and intrinsic. Meanwhile, the Sequential Pose Head proposes the extrinsic parameters of each video camera. Then the two results are aggregated together and propose the camera pose in global axis:

$$G^{global}(i,j) = G^{seq}(i,j) \times G^{rel}(i,j) \tag{11}$$

where G is the $\begin{pmatrix} R & T \\ 0 & 1 \end{pmatrix}$, which are decoded from different camera pose head, and $\times$ indicates matrix multiplication.

Depth Head: As VGGT architecture, we use DPT head[5] to decode the geometry tokens of each image to depths. It outputs prediction results of depths and depth confidence maps. The DPT head implements 4 refinement sub-modules to gradually decode high-resolution dense depths from spatial-compressed geometry tokens.

Scale Head: To transform the normalized geometry information to real-world scale, we predict the scale by comparing real-world relative poses and the predicted normalized relative poses as:

$$Scale = Avg(\sum_{j} \frac{T_{real}^{rel}(j)}{T_{norm}^{rel}(j)}) \tag{12}$$

where T indicate relative-pose translation from $\begin{pmatrix} R & T \\ 0 & 1 \end{pmatrix}$. After multiplying the scale to depths and translation of camera extrinsics, the final world points' prediction can recover to the real-world scale.

3.6. Loss

Following the loss design of VGGT, we implement the depth and camera pose loss to supervise DriveVGGT training process. In general, the total loss is calculated as:

$$L_{total} = \lambda_1 L_{depth} + \lambda_2 L_{rel} + \lambda_3 L_{seq} \tag{13}$$

where λ_1 is 0.1, λ_2 and λ_3 are set as 1.0.

Similar to VGGT, we calculate L_{depth} as 3 parts, which are depth error, depth grad error and uncertainty map error:

$$\begin{aligned} L_{\text{depth}} = \sum_{i=1}^{N}\sum_{j=1}^{M} \left\| \hat{\Sigma}_{ij}\left(\hat{D}_{ij} - D_{ij}\right)\right\| + \\ \left\| \hat{\Sigma}_{ij}\left(\nabla\hat{D}_{ij} - \nabla D_{ij}\right)\right\| - \alpha \log \hat{\Sigma}_{ij} \end{aligned} \tag{14}$$

Compared to VGGT, we decouple the camera pose to two kinds of posed, and L_{rel} is calculated as:

$$L_{\text{rel}} = \sum_{j=1}^{M} \left\| \Sigma_j \left(\hat{\mathbf{g}}_j - \mathbf{g}_j\right)\right\|_{\epsilon} \tag{15}$$

and $L_{seq}$is calculated as:

$$L_{\text{seq}} = \sum_{i=1}^{N} \|\Sigma_i(\hat{\mathbf{g}}_i - \mathbf{g}_i)\|_{\epsilon} \tag{16}$$

where $||.||_{\epsilon}$ using the Huber loss.

4. Experiments

4.1. Datasets

The nuScenes dataset consists of various driving scenes . For each scene, nuScenes records 20 seconds with sufficient multimodal information from 6 cameras, 1 lidar, vehicle ego poses, sensor calibration, etc. In our experiment, we primarily use images of relative poses from 6 cameras as the model's inputs. Similar to previous related works on nuScenes, we use 700 driving scenes for training, and 150 for validation. As to each scene, we use labeled samples recorded at 2Hz for training and testing.

Meanwhile, it is not probable to directly use sparse lidar points to generate the depth map as ground truth. Concerned about this weakness in the nuScenes dataset, we implement two effective steps to generate dense depth maps for training. Firstly, we aggregate multi-frame lidar points to build the point clouds of the entire scene with more detail. For labeled dynamic objects, we use their 3D box from each time step to aggregate their points. Furthermore, after projecting the points onto the depths, we utilize a depth augmentation algorithm to enhance the validity of the depth map. The two steps will bring some noise to the depth ground truth, while they can make it sufficient for training. The two-step data enhancement is illustrated in Fig. 4.

4.2. Implementation Details

For models' inputs, we decrease the initial image resolution of nuScenes from 1600x900 to 518x280, and implement the same changes to images' intrinsics during ground truth generation. Then, like VGGT, we propose scale normalization to the depth map, camera poses to keep the scale consistent, while we additionally use the scale for training. We train all models on 8 NVIDIA H200 GPUs and test them on 1 NVIDIA H200 GPU. As to the training process, initially, we randomly input 3-10 frame multi-camera images (18-60 images) from scenes for 20 epochs. Each epoch trains 1000 times with 2e-4 learning rate. Then we freeze the aggregator and fine-tune for another 5 epochs with 1e-5 learning rate. For fair comparisons, we train other models with the same method.

Table 1. **Multi-camera Pose Estimation on nuScenes dataset.**

Method	Relative poses	frame=15		frame=25		frame=35	
		AUC(30) ↑	AUC(15) ↑	AUC(30) ↑	AUC(15) ↑	AUC(30) ↑	AUC(15) ↑
VGGT [40]	✗	0.8531	<u>0.7689</u>	<u>0.7866</u>	<u>0.6719</u>	0.6871	0.5477
StreamVGGT [65]	✗	0.7005	0.5884	OOM	OOM	OOM	OOM
fastVGGT [34]	✗	0.8246	0.7191	0.7707	0.6435	0.6830	0.5357
VGGT [40]	✓	0.8164	0.7195	0.7403	0.6136	0.6445	0.5002
fastVGGT [34]	✓	0.7915	0.6764	0.7321	0.5954	0.6477	0.4976
DriveVGGT(VGGT)	✓	**0.8635**	**0.7706**	**0.8010**	**0.6778**	**0.7200**	**0.5811**
DriveVGGT(fastVGGT)	✓	<u>0.8534</u>	0.7498	0.7844	0.6514	<u>0.6995</u>	<u>0.5510</u>

Table 2. **Multi-camera Depth Estimation on nuScenes dataset.**

Method	Relative poses	frame=15		frame=25		frame=35	
		Abs rel ↓	δ^3 ↑	Abs rel ↓	δ^3 ↑	Abs rel ↓	δ^3 ↑
VGGT [40]	✗	0.3666	0.8791	<u>0.3654</u>	0.8817	0.3605	0.8858
StreamVGGT [65]	✗	**0.3636**	<u>0.8811</u>	OOM	OOM	OOM	OOM
fastVGGT [34]	✗	0.3684	0.8782	0.3693	0.8794	0.3660	0.8825
VGGT [40]	✓	0.3718	0.8779	0.3700	0.8805	0.3647	0.8844
fastVGGT [34]	✓	<u>0.3655</u>	0.8784	0.3691	0.8795	0.3658	0.8826
DriveVGGT(VGGT)	✓	0.3805	0.8747	0.3705	<u>0.8825</u>	<u>0.3601</u>	<u>0.8892</u>
DriveVGGT(fastVGGT)	✓	<u>0.3655</u>	**0.8854**	**0.3601**	**0.8894**	**0.3539**	**0.8935**

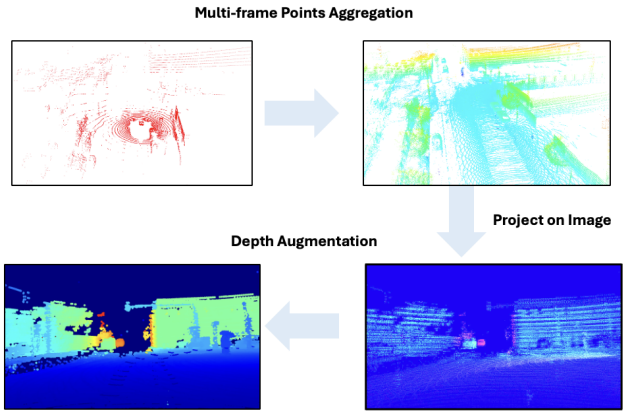

Figure 4. Two-step data enhancement for depth ground truth.

4.3. Pose Estimation

To compare the pose estimation of the proposed methods with other VGGT-based methods, we test VGGT, StreamVGGT, and fastVGGT on the nuScenes datasets. To illustrate the models' performance on different numbers of images, we set three kinds of image inputs: 15 frames (90 images), 25 frames (150 images), and 35 frames (210 images). Meanwhile, we incorporate relative pose embedding into VGGT and fastVGGT to demonstrate the role of relative poses in these models. Regarding our method, we implement two base geometry transformers to achieve temporal video attention in the TVA module, namely DriveVGGT (VGGT) and DriveVGGT (fastVGGT). The results are shown in Table 1. Initially, DriveVGGT (VGGT) achieves better performance than other methods, especially in scenes with 210 images. Meanwhile, as to the implementation of camera pose embedding, VGGT and fastVGGT exist performance degradation. However, as to DriveVGGT, the aggregation will improve the accuracy of camera pose estimation, which demonstrates the sufficient use of relative poses in DriveVGGT.

4.4. Depth Estimation

The comparison of depth estimation is shown in Table 2. As the evaluation of camera pose estimation, we test VGGT, StreamVGGT, fastVGGT and DriveVGGT on the nuScenes dataset. As to metric Abs Rel, DriveVGGT(fastVGGT) achieves the best depth estimation performance in scene-35 scenes, which indicates its ability to process long sequence multi-camera videos. StreamVGGT outperforms other methods in frame-15 scenes.

4.5. Inference Time Estimation

The comparison of inference time is shown in Table 3. In general, the proposed method achieves faster inference speed in contrast to VGGT and fastVGGT. The inference time of DriveVGGT(VGGT) is only 50% of VGGT's in frame-35 scenes. Meanwhile, DriveVGGT(fastVGGT)'s speed is lower than DriveVGGT(VGGT), which is due to the extra

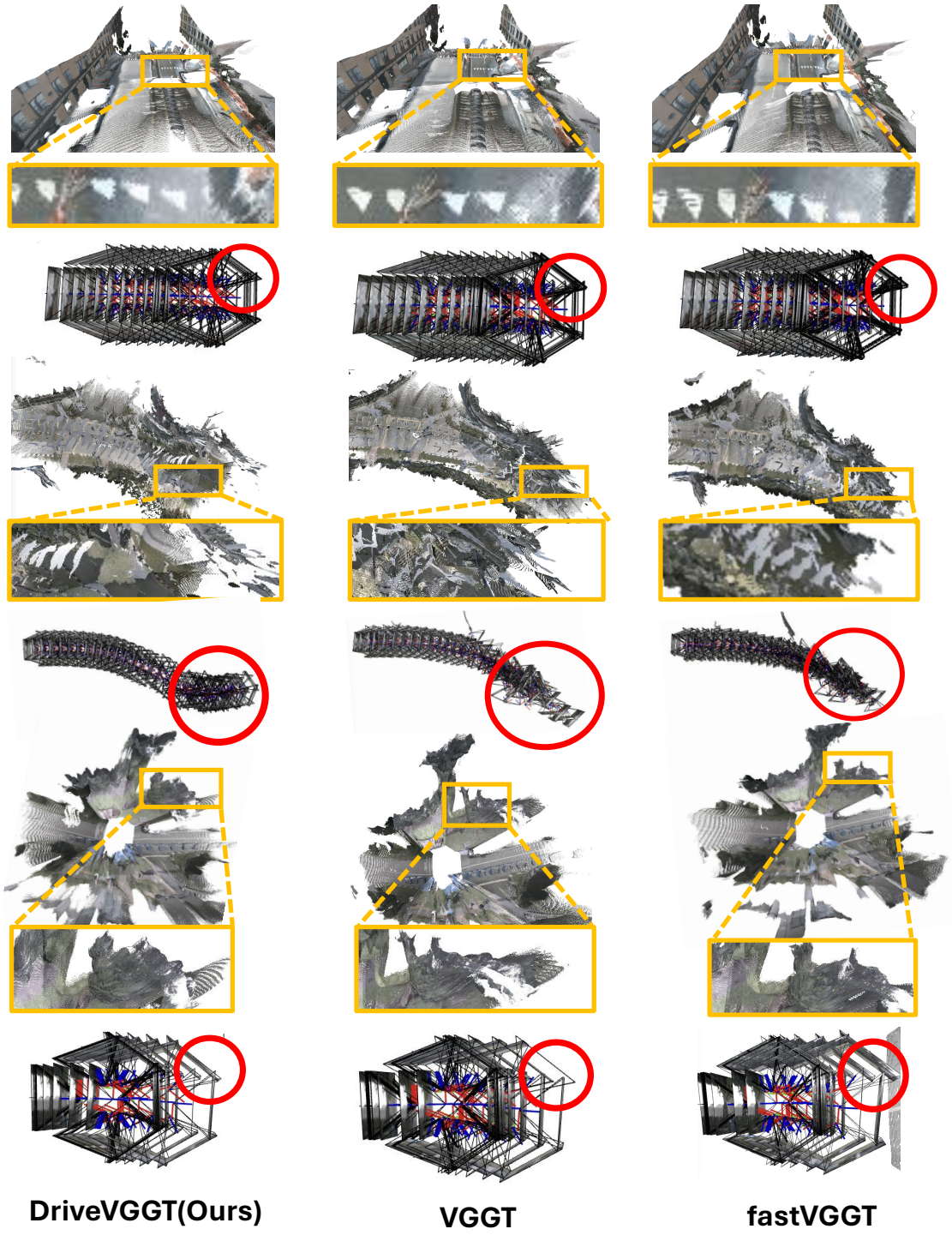

Figure 5. **Qualitative results** of DriveVGGT, VGGT, fastVGGT. We visualize predictions' global points and image poses to compare models' performance comprehensively.

token aggregation algorithm in fastVGGT, resulting in a delay in inference time when processing fewer images.

4.6. Visualization

To quantify the comprehensive performance of the proposed method, we compare the visualization results of VGGT,

Table 3. **Multi-camera inference time on nuScenes dataset.**

Method	Inference time (ms) ↓		
	frames=15	frames=25	frames=35
VGGT [40]	2268	5241	9666
StreamVGGT [65]	6916	OOM	OOM
fastVGGT [34]	1950	3341	4949
DriveVGGT(VGGT)	**1836**	**3294**	**4907**
DriveVGGT(fastVGGT)	2390	3823	5043

fastVGGT, and DriveVGGT in Fig.5. To generate the final point clouds, we project the depth map to global points with the guidance of the image extrinsic.

We visualize reconstruction results and camera pose outputs of 3 typical vehicle motion states in traffic scenes. We use 30×6 images as model input. In the first scene, the reconstruction results achieve great results. However, the camera pose outputs from fastVGGT exhibit slight misalignment compared to the other methods. When it comes to the second scene, although DriveVGGT can maintain stable pose predictions from the first image to the final, VGGT and fastVGGT exhibit terrible performance degradation, especially for images far from the initial image. Meanwhile, the serious pose misalignment causes ambiguous outputs of the point clouds. Regarding the final scene, the camera pose outputs of VGGT and fastVGGT exhibit slight misalignment, whereas DriveVGGT can output a stable relative pose prediction. Meanwhile, DriveVGGT achieves better reconstruction results for surrounding targets, such as vegetation.

4.7. Ablation Experiment

To validate the effectiveness of the proposed components, we conduct an ablation study by removing the proposed modules from DriveVGGT, and the detailed evaluation is presented in Table 4. The baseline module only utilizes the TVA module to implement attention among images in the video. The test results indicate that the baseline could not handle a multi-camera system as the lack of relative pose representation. After adding relative pose embedding, the model can output a correct pose prediction of the multi-camera system.

Table 4. **Ablations of different components.**

Method	frame=25		
	AUC(30) ↑	Abs rel ↓	Time ↓ (ms)
baseline(TVA)	0.039	0.3711	**2052**
+rel pose embed	0.7855	0.3707	2098
DriveVGGT	**0.8010**	**0.3705**	3294

To fully evaluate the function of window attention, we test 3 types of window sizes in Table 5. Compared to size-5 and size-7, size-3 can maintain a balance between performance and efficiency.

Table 5. **Ablation experiments of different window sizes.**

Window size	frame=25		
	AUC(30)↑	Abs rel↓	Time↓ (ms)
size=3(Ours)	0.8010	**0.3705**	**3294**
size=5	0.8033	0.3741	4924
size=7	**0.8087**	0.3744	7263

Table 6. **Depth estimation comparison between least squares method and scale-based method.**

Method	frame=15	
	Abs rel↓	δ^3 ↑
least squares method	0.3805	0.8747
scale-based method	0.3666	0.7412

To evaluate the effectiveness of the scale head, we compare the depth prediction results with the ground truth using two alignment methods: the least squares method and the scale-based method. The results are shown in Table 6. The results indicate that the scale prediction can transform depths to a real-world scale. Subsequently, we visualize the real-world scale point clouds and camera extrinsics in Fig. 6. The results indicate that the real-scaled point clouds maintain the similar geometry consistency as the normalized ones.

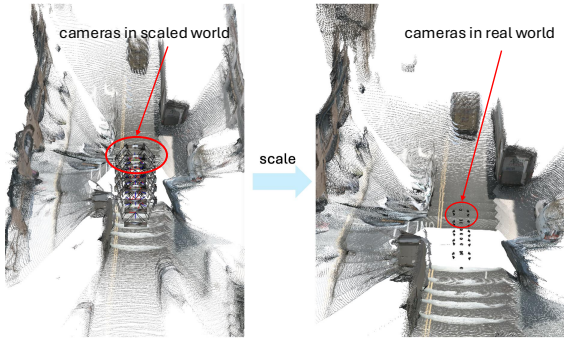

Figure 6. Comparison between scaled scenes and real scenes.

4.8. Conclusion

In this work, we propose MVGT, a feed-forward reconstruction model specializing in multi-camera geometry predictions. Compared to previous methods, MVGT can effectively utilize relative camera poses to enhance the accuracy of geometric predictions, such as camera pose and depth estimation. Comprehensive evaluations on the nuScenes dataset demonstrate the outperforming performance compared to

previous feed-forward methods, while maintaining lower computational consumption.

References

[1] Jianhong Bai, Menghan Xia, Xintao Wang, Ziyang Yuan, Zuozhu Liu, Haoji Hu, Pengfei Wan, and Di ZHANG. Syncammaster: Synchronizing multi-camera video generation from diverse viewpoints. In *The Thirteenth International Conference on Learning Representations*, 2024.

[2] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11621–11631, 2020.

[3] Yixin Chen, Junfeng Ni, Nan Jiang, Yaowei Zhang, Yixin Zhu, and Siyuan Huang. Single-view 3d scene reconstruction with high-fidelity shape and texture. In *2024 International Conference on 3D Vision (3DV)*, pages 1456–1467. IEEE, 2024.

[4] Yurui Chen, Junge Zhang, Ziyang Xie, Wenye Li, Feihu Zhang, Jiachen Lu, and Li Zhang. S-nerf++: Autonomous driving simulation via neural reconstruction and generation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

[5] Zhiyang Chen, Yousong Zhu, Chaoyang Zhao, Guosheng Hu, Wei Zeng, Jinqiao Wang, and Ming Tang. Dpt: Deformable patch-based transformer for visual recognition. In *Proceedings of the 29th ACM international conference on multimedia*, pages 2899–2907, 2021.

[6] 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. *arXiv preprint arXiv:2507.16443*, 2025.

[7] Tianchen Deng, Guole Shen, Tong Qin, Jianyu Wang, Wentao Zhao, Jingchuan Wang, Danwei Wang, and Weidong Chen. Plgslam: Progressive neural scene represenation with local to global bundle adjustment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19657–19666, 2024.

[8] Cunxin Fan, Xiaosong Jia, Yihang Sun, Yixiao Wang, Jianglan Wei, Ziyang Gong, Xiangyu Zhao, Masayoshi Tomizuka, Xue Yang, Junchi Yan, et al. Interleave-vla: Enhancing robot manipulation with interleaved image-text instructions. *arXiv preprint arXiv:2505.02152*, 2025.

[9] Xiaofeng Gao, Xiaosong Jia, Chaoqi Yang, and Guihai Chen. Using survival theory in early pattern detection for viral cascades. *IEEE Transactions on Knowledge and Data Engineering*, 34(5):2497–2511, 2020.

[10] Shuang Guo and Guillermo Gallego. Event-based mosaicing bundle adjustment. In *European Conference on Computer Vision*, pages 479–496. Springer, 2024.

[11] Ruiyang Hao, Haibao Yu, Jiaru Zhong, Chuanye Wang, Jiahao Wang, Yiming Kan, Wenxian Yang, Siqi Fan, Huilin Yin, Jianing Qiu, et al. Research challenges and progress in the end-to-end v2x cooperative autonomous driving competition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1828–1839, 2025.

[12] Yihan Hu, Jiazhi Yang, Li Chen, Keyu Li, Chonghao Sima, Xizhou Zhu, Siqi Chai, Senyao Du, Tianwei Lin, Wenhai Wang, Lewei Lu, Xiaosong Jia, Qiang Liu, Jifeng Dai, Yu Qiao, and Hongyang Li. Planning-oriented autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 17853–17862, 2023.

[13] Xiaosong Jia, Qitian Wu, Xiaofeng Gao, and Guihai Chen. Sentimem: Attentive memory networks for sentiment classification in user review. In *International Conference on Database Systems for Advanced Applications*, pages 736–751. Springer, 2020.

[14] Xiaosong Jia, Liting Sun, Hang Zhao, Masayoshi Tomizuka, and Wei Zhan. Multi-agent trajectory prediction by combining egocentric and allocentric views. In *Conference on Robot Learning*, pages 1434–1443. PMLR, 2022.

[15] Xiaosong Jia, Li Chen, Penghao Wu, Jia Zeng, Junchi Yan, Hongyang Li, and Yu Qiao. Towards capturing the temporal dynamics for trajectory prediction: a coarse-to-fine approach. In *Conference on Robot Learning*, pages 910–920. PMLR, 2023.

[16] Xiaosong Jia, Yulu Gao, Li Chen, Junchi Yan, Patrick Langechuan Liu, and Hongyang Li. Driveadapter: Breaking the coupling barrier of perception and planning in end-to-end autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7953–7963, 2023.

[17] Xiaosong Jia, Penghao Wu, Li Chen, Yu Liu, Hongyang Li, and Junchi Yan. Hdgt: Heterogeneous driving graph transformer for multi-agent trajectory prediction via scene encoding. *IEEE transactions on pattern analysis and machine intelligence*, 45(11):13860–13875, 2023.

[18] Xiaosong Jia, Penghao Wu, Li Chen, Jiangwei Xie, Conghui He, Junchi Yan, and Hongyang Li. Think twice before driving: Towards scalable decoders for end-to-end autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21983–21994, 2023.

[19] Xiaosong Jia, Shaoshuai Shi, Zijun Chen, Li Jiang, Wenlong Liao, Tao He, and Junchi Yan. Amp: Autoregressive motion prediction revisited with next token prediction for autonomous driving. *arXiv preprint arXiv:2403.13331*, 2024.

[20] Xiaosong Jia, Zhenjie Yang, Qifeng Li, Zhiyuan Zhang, and Junchi Yan. Bench2drive: Towards multi-ability benchmarking of closed-loop end-to-end autonomous driving. In *NeurIPS 2024 Datasets and Benchmarks Track*, 2024.

[21] Xiaosong Jia, Junqi You, Zhiyuan Zhang, and Junchi Yan. Drivetransformer: Unified transformer for scalable end-to-end autonomous driving. In *International Conference on Learning Representations (ICLR)*, 2025.

[22] Xin Kong, Shikun Liu, Xiaoyang Lyu, Marwan Taher, Xiaojuan Qi, and Andrew J Davison. Eschernet: A generative model for scalable view synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9503–9513, 2024.

[23] Zhengfei Kuang, Shengqu Cai, Hao He, Yinghao Xu, Hongsheng Li, Leonidas J Guibas, and Gordon Wetzstein. Collaborative video diffusion: Consistent multi-video generation with

camera control. *Advances in Neural Information Processing Systems*, 37:16240–16271, 2024.

[24] Qifeng Li, Xiaosong Jia, Shaobo Wang, and Junchi Yan. Think2drive: Efficient reinforcement learning by thinking with latent world model for autonomous driving (in carla-v2). In *European Conference on Computer Vision*, pages 142–158. Springer, 2025.

[25] Ruilong Li, Brent Yi, Junchen Liu, Hang Gao, Yi Ma, and Angjoo Kanazawa. Cameras as relative positional encoding. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

[26] Samuel Li, Pujith Kachana, Prajwal Chidananda, Saurabh Nair, Yasutaka Furukawa, and Matthew Brown. Rig3r: Rig-aware conditioning and discovery for 3d reconstruction. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

[27] Zhengqi Li, Richard Tucker, Forrester Cole, Qianqian Wang, Linyi Jin, Vickie Ye, Angjoo Kanazawa, Aleksander Holynski, and Noah Snavely. Megasam: Accurate, fast and robust structure and motion from casual dynamic videos. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 10486–10496, 2025.

[28] Liewen Liao, Weihao Yan, Ming Yang, and Songan Zhang. Learning-based 3d reconstruction in autonomous driving: A comprehensive survey. *arXiv preprint arXiv:2503.14537*, 2025.

[29] Han Lu, Xiaosong Jia, Yichen Xie, Wenlong Liao, Xiaokang Yang, and Junchi Yan. Activead: Planning-oriented active learning for end-to-end autonomous driving. *arXiv preprint arXiv:2403.02877*, 2024.

[30] Hannan Lu, Xiaohe Wu, Shudong Wang, Xiameng Qin, Xinyu Zhang, Junyu Han, Wangmeng Zuo, and Ji Tao. Seeing beyond views: Multi-view driving scene video generation with holistic attention. *arXiv preprint arXiv:2412.03520*, 2024.

[31] Jiageng Mao, Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. 3d object detection for autonomous driving: A comprehensive survey. *International Journal of Computer Vision*, 131(8):1909–1963, 2023.

[32] Qiaowei Miao, Kehan Li, Jinsheng Quan, Zhiyuan Min, Shaojie Ma, Yichao Xu, Yi Yang, Ping Liu, and Yawei Luo. Advances in 4d generation: A survey. *arXiv preprint arXiv:2503.14501*, 2025.

[33] Takeru Miyato, Bernhard Jaeger, Max Welling, and Andreas Geiger. Gta: A geometry-aware attention mechanism for multi-view transformers. In *The Twelfth International Conference on Learning Representations*.

[34] You Shen, Zhipeng Zhang, Yansong Qu, and Liujuan Cao. Fastvggt: Training-free acceleration of visual geometry transformer. *arXiv preprint arXiv:2509.02560*, 2025.

[35] Chen Shi, Shaoshuai Shi, Xiaoyang Lyu, Chunyang Liu, Kehua Sheng, Bo Zhang, and Li Jiang. Unisplat: Unified spatio-temporal fusion via 3d latent scaffolds for dynamic driving scene reconstruction. *arXiv preprint arXiv:2511.04595*, 2025.

[36] Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

[37] Jiaxiang Tang, Xiaokang Chen, Jingbo Wang, and Gang Zeng. Point scene understanding via disentangled instance mesh reconstruction. In *European conference on computer vision*, pages 684–701. Springer, 2022.

[38] Chen Wang, Hao-Yang Peng, Ying-Tian Liu, Jiatao Gu, and Shi-Min Hu. Diffusion models for 3d generation: A survey. *Computational Visual Media*, 11(1):1–28, 2025.

[39] Chung-Shien Brian Wang, Christian Schmidt, Jens Piekenbrinck, and Bastian Leibe. Faster vggt with block-sparse global attention. *arXiv preprint arXiv:2509.07120*, 2025.

[40] Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5294–5306, 2025.

[41] 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.

[42] Yifan Wang, Jianjun Zhou, Haoyi Zhu, Wenzheng Chang, Yang Zhou, Zizun Li, Junyi Chen, Jiangmiao Pang, Chunhua Shen, and Tong He. Pi3: Permutation-equivariant visual geometry learning. *arXiv preprint arXiv:2507.13347*, 2025.

[43] Kelvin Wong, Yanlei Gu, and Shunsuke Kamijo. Mapping for autonomous driving: Opportunities and challenges. *IEEE Intelligent Transportation Systems Magazine*, 13(1):91–106, 2020.

[44] Penghao Wu, Xiaosong Jia, Li Chen, Junchi Yan, Hongyang Li, and Yu Qiao. Trajectory-guided control prediction for end-to-end autonomous driving: A simple yet strong baseline. *Advances in Neural Information Processing Systems*, 35:6119–6132, 2022.

[45] Penghao Wu, Li Chen, Hongyang Li, Xiaosong Jia, Junchi Yan, and Yu Qiao. Policy pre-training for autonomous driving via self-supervised geometric modeling. In *International Conference on Learning Representations*, 2023.

[46] Tong Wu, Shuai Yang, Ryan Po, Yinghao Xu, Ziwei Liu, Dahua Lin, and Gordon Wetzstein. Video world models with long-term spatial memory. *arXiv preprint arXiv:2506.05284*, 2025.

[47] Yinshuang Xu, Dian Chen, Katherine Liu, Sergey Zakharov, Rares Ambrus, Kostas Daniilidis, and Vitor Guizilini. Se (3) equivariant ray embeddings for implicit multi-view depth estimation. *Advances in Neural Information Processing Systems*, 37:13627–13659, 2024.

[48] Haiwei Xue, Xiangyang Luo, Zhanghao Hu, Xin Zhang, Xunzhi Xiang, Yuqin Dai, Jianzhuang Liu, Zhensong Zhang, Minglei Li, Jian Yang, et al. Human motion video generation: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

[49] Jiazhi Yang, Shenyuan Gao, Yihang Qiu, Li Chen, Tianyu Li, Bo Dai, Kashyap Chitta, Penghao Wu, Jia Zeng, Ping Luo, et al. Generalized predictive model for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14662–14672, 2024.

[50] Jiazhi Yang, Kashyap Chitta, Shenyuan Gao, Long Chen, Yuqian Shao, Xiaosong Jia, Hongyang Li, Andreas Geiger,

Xiangyu Yue, and Li Chen. Resim: Reliable world simulation for autonomous driving. *Advances in Neural Information Processing Systems*, 2025.

[51] Kairui Yang, Zihao Guo, Gengjie Lin, Haotian Dong, Zhao Huang, Yipeng Wu, Die Zuo, Jibin Peng, Ziyuan Zhong, Xin Wang, et al. Trajectory-llm: A language-based data generator for trajectory prediction in autonomous driving. In *The Thirteenth International Conference on Learning Representations*, 2025.

[52] Zhenjie Yang, Xiaosong Jia, Hongyang Li, and Junchi Yan. Llm4drive: A survey of large language models for autonomous driving, 2023.

[53] Zhenjie Yang, Yilin Chai, Xiaosong Jia, Qifeng Li, Yuqian Shao, Xuekai Zhu, Haisheng Su, and Junchi Yan. Drivemoe: Mixture-of-experts for vision-language-action model in end-to-end autonomous driving. *arXiv preprint arXiv:2505.16278*, 2025.

[54] Zhenjie Yang, Xiaosong Jia, Qifeng Li, Xue Yang, Maoqing Yao, and Junchi Yan. Raw2drive: Reinforcement learning with aligned world models for end-to-end autonomous driving (in carla v2). *Advances in Neural Information Processing Systems*, 2025.

[55] Baijun Ye, Minghui Qin, Saining Zhang, Moonjun Gong, Shaoting Zhu, Hao Zhao, and Hang Zhao. Gs-occ3d: Scaling vision-only occupancy reconstruction with gaussian splatting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 25925–25937, 2025.

[56] Junqi You, Xiaosong Jia, Zhiyuan Zhang, Yutao Zhu, and Junchi Yan. Bench2drive-r: Turning real world data into reactive closed-loop autonomous driving benchmark by generative model. *arXiv preprint arXiv:2412.09647*, 2024.

[57] Meng You, Zhiyu Zhu, Hui Liu, and Junhui Hou. Nvs-solver: Video diffusion model as zero-shot novel view synthesizer. *arXiv preprint arXiv:2405.15364*, 2024.

[58] Junyi Zhang, Charles Herrmann, Junhwa Hur, Varun Jampani, Trevor Darrell, Forrester Cole, Deqing Sun, and Ming-Hsuan Yang. Monst3r: A simple approach for estimating geometry in the presence of motion. In *The Thirteenth International Conference on Learning Representations*, 2024.

[59] Jiahui Zhang, Yuelei Li, Anpei Chen, Muyu Xu, Kunhao Liu, Jianyuan Wang, Xiao-Xiao Long, Hanxue Liang, Zexiang Xu, Hao Su, et al. Advances in feed-forward 3d reconstruction and view synthesis: A survey. *arXiv preprint arXiv:2507.14501*, 2025.

[60] Jason Y Zhang, Deva Ramanan, and Shubham Tulsiani. Relpose: Predicting probabilistic relative rotation for single objects in the wild. In *Proceedings of the European Conference on Computer Vision*, pages 592–611. Springer, 2022.

[61] Wei Zhang, Yihang Wu, Songhua Li, Wenjie Ma, Xin Ma, Qiang Li, and Qi Wang. Review of feed-forward 3d reconstruction: From dust3r to vggt. *arXiv preprint arXiv:2507.08448*, 2025.

[62] Jensen Zhou, Hang Gao, Vikram Voleti, Aaryaman Vasishta, Chun-Han Yao, Mark Boss, Philip Torr, Christian Rupprecht, and Varun Jampani. Stable virtual camera: Generative view synthesis with diffusion models. *CoRR*, 2025.

[63] Xiaoyu Zhou, Zhiwei Lin, Xiaojun Shan, Yongtao Wang, Deqing Sun, and Ming-Hsuan Yang. Drivinggaussian: Composite gaussian splatting for surrounding dynamic autonomous driving scenes. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 21634–21643, 2024.

[64] Yutao Zhu, Xiaosong Jia, Xinyu Yang, and Junchi Yan. Flatfusion: Delving into details of sparse transformer-based camera-lidar fusion for autonomous driving. *IEEE International Conference on Robotics and Automation (ICRA)*, 2025.

[65] Dong Zhuo, Wenzhao Zheng, Jiahe Guo, Yuqi Wu, Jie Zhou, and Jiwen Lu. Streaming 4d visual geometry transformer. *arXiv preprint arXiv:2507.11539*, 2025.

DriveVGGT: Visual Geometry Transformer for Autonomous Driving

Supplementary Material

5. Rationale

5.1. The Explanation of DriveVGGT (fastVGGT) Inference Time Degradation (compared with DriveVGGT (VGGT))

In Table 5 of the manuscript, the inference time of DriveVGGT (fastVGGT) is longer than that of DriveVGGT (VGGT), which may cause confusion. The explanation is that in DriveVGGT, we process images belonging to each camera separately (the function of the Temporal Video Attention module). Thus, the VGGT or fastVGGT treats only 1/6 of the images as a batch in the entire multi-camera inputs, and the image number of each batch is 15, 25, 35 (not 90, 150, 210). The inference time of VGGT and fastVGGT for different numbers of images is shown in the following table. The results indicate that VGGT is faster than fastVGGT when dealing with fewer images. That's why DriveVGGT (fastVGGT) is slower than DriveVGGT (VGGT).

Table 7. **Multi-camera inference time on nuScenes dataset.**

Method	Inference time ↓(ms)			
	images=6	images=30	images=54	images=150
VGGT	**95**	**461**	**1027**	5241
fastVGGT	537	893	1277	**3341**

5.2. Token-level DriveVGGT Pipeline

We visualize the DriveVGGT Pipeline details to better illustrate the function of the two proposed modules for each token.

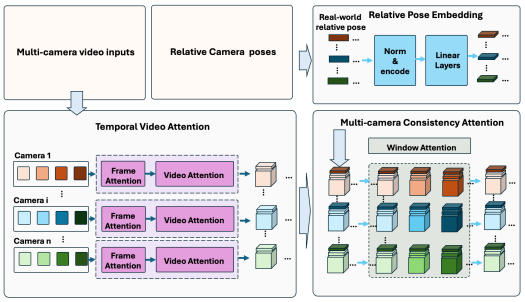

Figure 7. Token-level Visualization of DriveVGGT

5.3. DriveVGGT Prediction Visualization

We visualize the DriveVGGT outputs to better illustrate the relationship between prediction heads and tokens.

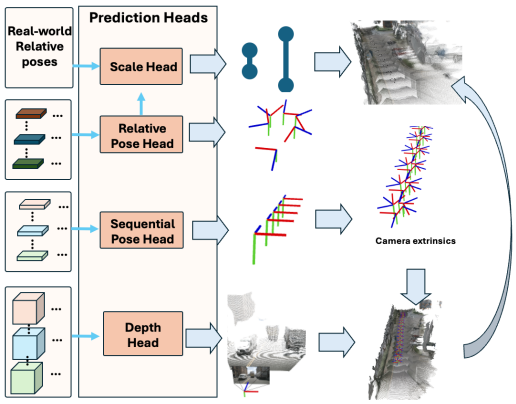

Figure 8. Prediction heads

5.4. Scale Head Visualization

To visualize the scaled global points from images, we visualize the scaled results and raw point clouds from the lidar sensor. To achieve better visualization results, we normalize the color of scaled image points. The scale comparison demonstrates that the model can achieve general real-world geometry accuracy with the scale head.

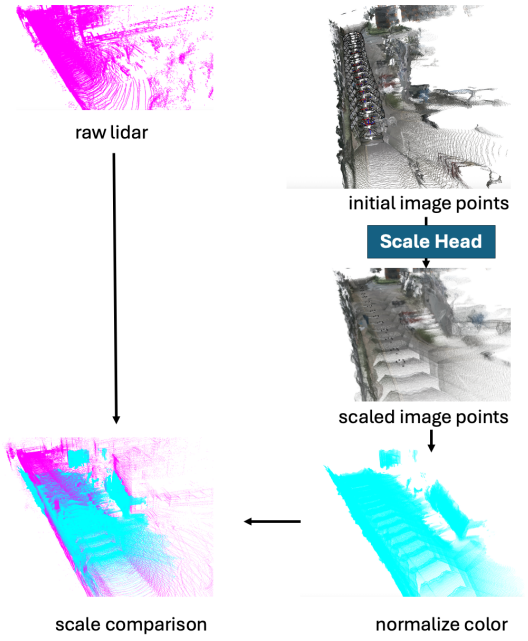

Figure 9. Comparison between raw lidar and scaled image points.

5.5. More Visualization Results

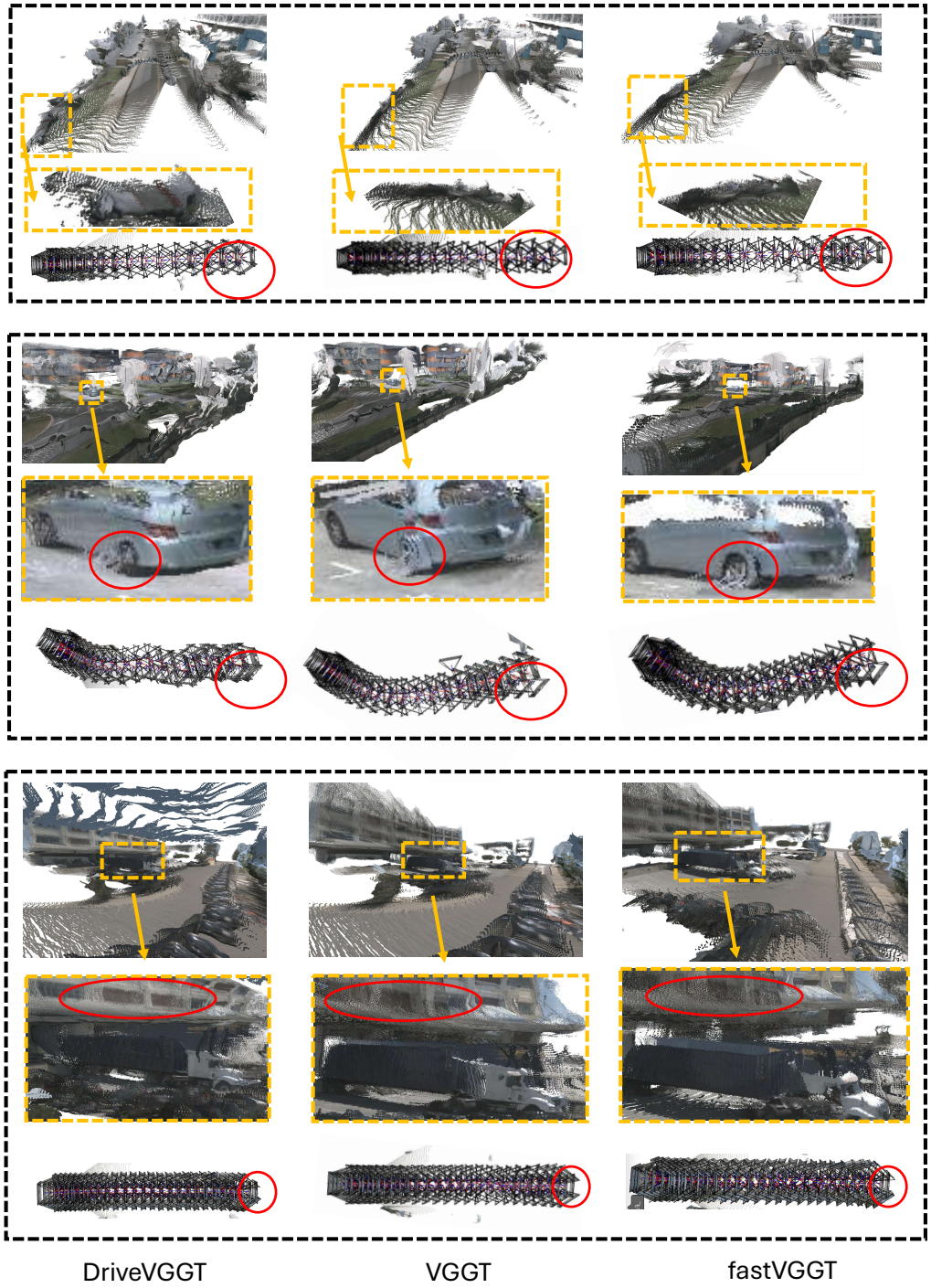

Figure 10. Visualization Results 1.

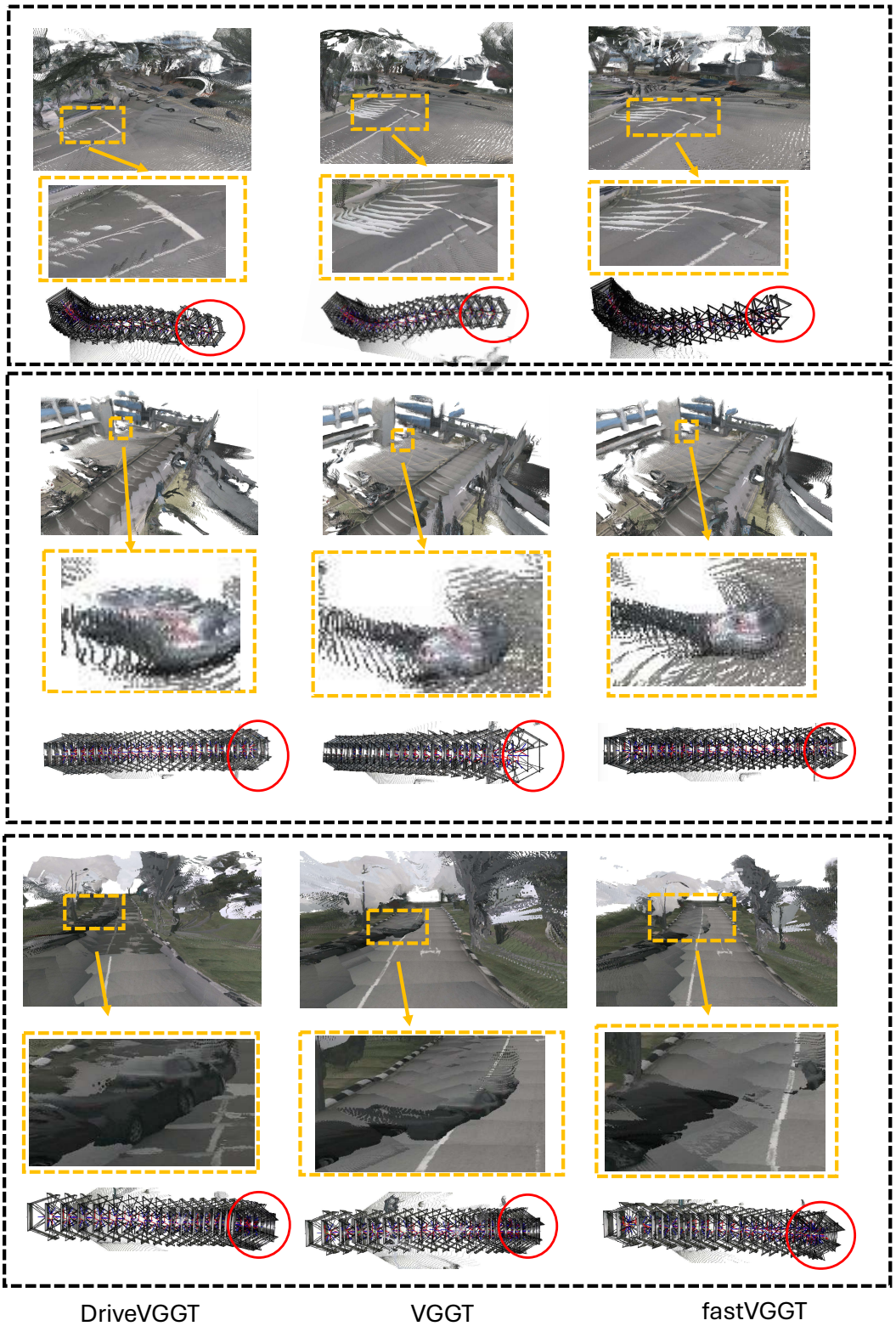

Figure 11. Visualization Results 2.