4D-VGGT: A SpatioTemporal Foundation Model for Dynamic Scene Geometry Estimation

Haonan Wang[1], Hanyu Zhou[2⋆], Haoyue Liu[1], and Luxin Yan[1]

[1] School of Artificial Intelligence and Automation, Huazhong University of Science and Technology
[2] School of Computing, National University of Singapore
{whn_aurora,yanluxin}@hust.edu.cn, hy.zhou@nus.edu.sg

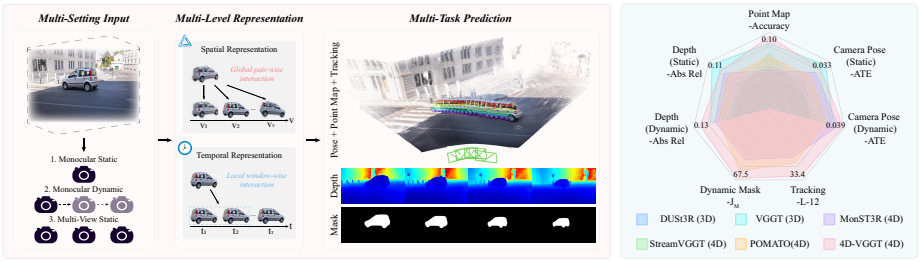
Abstract. We investigate a challenging task of dynamic scene geometry estimation, which requires representing both spatial and temporal features. Typically, existing methods align two features into a unified latent space to model scene geometry. However, this unified paradigm suffers from potentially mismatched representations due to the heterogeneous nature between spatial and temporal features. In this work, we propose **4D-VGGT**, a general foundation model with divide-and-conquer spatiotemporal representation for dynamic scene geometry. Our model is divided into three aspects: 1) **Multi-setting input.** We design an adaptive visual grid that supports input sequences with arbitrary numbers of views and time steps. 2) **Multi-level representation.** We propose a cross-view global fusion for spatial representation and a cross-time local fusion for temporal representation. 3) **Multi-task prediction.** We append multiple task-specific heads to spatiotemporal representations, enabling a comprehensive visual geometry estimation for dynamic scenes. Under this unified framework, these components enhance the feature discriminability and application universality of our model for dynamic scenes. In addition, we integrate multiple geometry datasets to train our model and conduct extensive experiments to verify the effectiveness of our method across various tasks on multiple dynamic scene geometry benchmarks. Our code is released at https://github.com/Haonan-Wang-aurora/4D-VGGT.

Keywords: Dynamic Scene Geometry Estimation · Foundation Models

1 Introduction

Foundation models aim to learn general-purpose representations from massive data and are widely applied in diverse vision tasks, such as pose [1, 2], depth [3–6], and point trajectories [7–9]. Recently, some researchers leverage the strong representation capacity of foundation models to learn the spatial features for geometry estimation through multi-task learning, *e.g.*, VGGT [10] and π3 [11]. Despite achieving great success in static scenes, these 3D methods still face challenges in dynamic scenes. Different from static scenes, dynamic scene geometry

⋆ Corresponding author.



(a) Paradigm of 4D-VGGT (b) Performance on Different Tasks

Fig. 1: Illustration of our model paradigm and performance. (a) Our 4D-VGGT accommodates various camera settings and adopts a divide-and-conquer spatiotemporal representation approach for different geometry tasks in dynamic scenes. (b) Performance comparison on multiple tasks. Our 4D-VGGT achieves consistently superior performance across various geometry tasks.

estimation requires the representation of both spatial and temporal features, which exceeds the representation scope of these 3D models. In this work, our goal is to learn the spatiotemporal representations for dynamic scene geometry.

Existing methods typically embed temporal cues into spatial features, thus representing the spatiotemporal features within a unified latent space. For instance, Zhang *et al.* [12] extend the 3D model DUSt3R [13] by estimating pair-wise point maps from spatial features and performing temporal consistency alignment. Zhuo *et al.* [14] augment VGGT [10] with a temporal causal attention after spatial attention. However, these methods suffer from the potential mismatched representation due to the heterogeneous nature between spatial and temporal features. This further causes these 4D models to produce unstable and unreliable geometry knowledge in dynamic scenes. Therefore, designing a divide-and-conquer spatiotemporal representation is crucial for dynamic scene geometry estimation.

Our design is based on two key insights: 1) *Factorized spatiotemporal representation enhances discriminability.* The spatiotemporal features in dynamic scenes exhibit significant heterogeneity: the spatial dimension follows structural consistency across different views, while the temporal dimension adheres to motion continuity across adjacent time steps. Leveraging this property is beneficial for representing the spatiotemporal characteristics of dynamic scenes. 2) *Adaptive visual representation improves universality.* Considering that conventional 3D geometry foundation models are capable of handling arbitrary views, we further extend 4D models to support inputs from arbitrary views and arbitrary time steps. This alteration improves the model universality in real-world applications.

In this work, we propose **4D-VGGT**, a general spatiotemporal foundation model for dynamic scene geometry estimation. As illustrated in Fig. 1, our model consists of three main components: 1) **Multi-Setting Input.** We design an adaptive visual grid that enables our model to accommodate visual features from diverse camera configurations through attention masks. 2) **Multi-Level Representation.** We propose a cross-view global fusion module to learn the spatial representation between various views, and a cross-time local fusion to model the temporal representation along continuous time steps. 3) **Multi-Task**

Prediction. We construct multiple task-specific heads and perform joint multi-task optimization to learn the corresponding spatiotemporal features for scene geometry. Under our unified framework, these components enable our model to utilize a shared spatiotemporal representation scheme to support diverse input configurations and accomplish a variety of visual tasks. Additionally, we also integrate diverse datasets to train our model for more effective dynamic scene geometry estimation. Our main contributions are summarized as follows:

- We present 4D-VGGT, a general foundation model with spatiotemporal awareness for dynamic scene geometry estimation. Our 4D-VGGT supports multi-setting input, multi-level representation, and multi-task prediction, providing a novel modeling paradigm for dynamic scenes.
- We observe that visual features extracted from various camera settings share similar distributions. This motivates the design of adaptive visual grid to enable our model to handle vision inputs with arbitrary views and time steps.
- We design cross-view global fusion and cross-time local fusion for feature discriminability. The former learns spatial representations across views, while the latter models temporal representations across adjacent time steps.
- We integrate diverse geometry datasets for training. Extensive experiments show the superiority of our method across various tasks in dynamic scenes.

2 Related Work

2.1 3D Geometry Estimation

The key to 3D geometry estimation lies in obtaining the spatial representation of static scenes. 3D methods can be broadly divided into optimization-based and learning-based approaches. Optimization-based methods, such as SfM [15–19], MVS [20–23], and visual SLAM [24–28], estimate static scene geometry by optimizing objective functions. However, their assumption of scene rigidity fails in dynamic environments with moving objects. Learning-based methods leverage deep networks to learn 3D priors of static scenes to estimate geometry properties such as pose, depth, and point trajectories. These methods either combine deep networks with traditional optimization [29–33], or use feedforward neural networks with point map representation [10, 11, 13, 34–42] to estimate scene geometry in a data-driven manner. The lack of temporal modeling limits their performance under dynamic conditions. Therefore, our method additionally introduces a temporal representation module to capture the varying dynamics patterns.

2.2 4D Geometry Estimation

4D geometry estimation requires effective spatiotemporal representation due to the highly complex and temporally-varying dynamics patterns inherent in dynamic scenes. Existing methods typically choose to directly embed temporal cues into spatial features for spatiotemporal representation. Some of these methods [12, 43–50] first obtain the local pair-wise point maps based on DUSt3R [13]

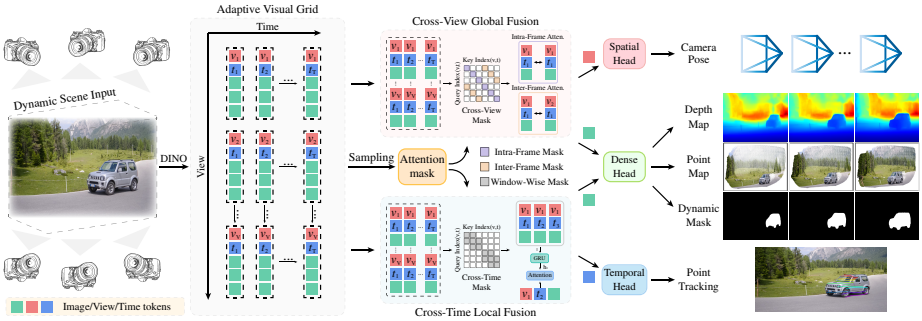


Fig. 2: Framework of our 4D-VGGT. The whole framework is divided into three main stages: 1) Multi-setting vision input. Encode input sequence with DINOv2 and construct adaptive visual grid. 2) Multi-level feature representation. Respectively capture spatial and temporal features in a divide-and-conquer manner. 3) Multi-task geometry prediction. Obtain results for multiple geometry tasks by specific prediction heads.

and then perform a global alignment using the temporal consistency constraint. Conversely, the others utilize additional temporal modules such as a memory-based framework [14,51–55] or a dynamic-aware module [56–59] to further enhance the underlying spatial representation module. However, these direct paradigms are often inefficient for dynamic scene geometry estimation since they lead to mismatched representations due to the inherently heterogeneous nature between spatial and temporal features. To address this limitation, in this work, we propose a novel divide-and-conquer strategy and explore two distinct masked-attention modules to separately represent spatial and temporal features.

2.3 Multi-Task Foundation Models

Recent multi-task foundation models aim to unify multiple perception or geometry tasks within a single architecture, typically by sharing a common encoder–decoder across tasks such as depth estimation, semantic segmentation, and normal prediction [60–65]. While these unified frameworks effectively exploit shared visual representations, they struggle to maintain task-specific distinctions, leading to suboptimal performance when tasks exhibit conflicting objectives. In this work, we follow a shared-but-distinct strategy and introduce various attention masks to guide the shared transformer-based spatiotemporal fusion module to perform distinct forms of attention computation for different tasks. This strategy can better adapt to different tasks while keeping the framework universal.

3 Our 4D-VGGT

3.1 Overall Framework

We propose 4D-VGGT, a general spatiotemporal foundation model for dynamic scene geometry estimation. As shown in Fig. 2, the whole framework is

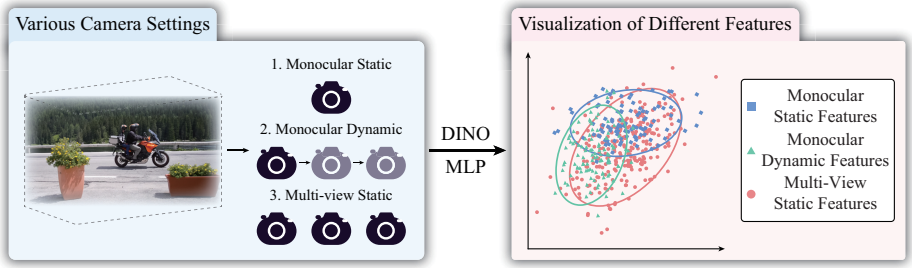


Fig. 3: Feature distribution of different camera settings. We extract DINOv2 features from all frames, organize them by the view-time configuration, and project them with the same MLP before UMAP visualization. Each point denotes a projected patch-level feature, colors indicate different camera settings, and the axes are the first two UMAP dimensions. The compact distributions show that DINOv2 provides a suitable basis for unifying different camera settings, motivating our adaptive visual grid.

divided into three parts: multi-setting input for adapting and encoding input sequences with arbitrary numbers of views and time steps, multi-level representation for divide-and-conquer spatiotemporal modeling, and multi-task prediction for dynamic scene geometry. In summary, our framework maps each frame I_{vt} to its corresponding geometry attributes, including camera parameter $g_{vt} \in \mathbb{R}^9$, depth map $D_{vt} \in \mathbb{R}^{H\times W}$, dynamic mask $M_{vt} \in \mathbb{R}^{H\times W}$, point map $P_{vt} \in \mathbb{R}^{3\times H\times W}$ and tracking results T_{vt} including 2D trajectory $T^{2D}_{vt} \in \mathbb{R}^{2\times N}$ and 3D trajectory $T^{3D}_{vt} \in \mathbb{R}^{3\times N}$, which is formulated as:

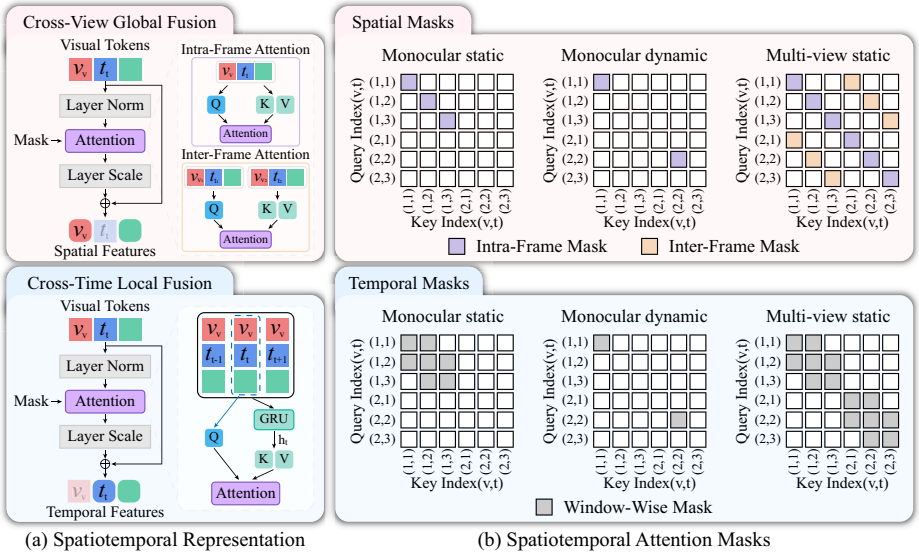
$$f((I_{vt})^{V,T}_{v,t=1}) = (g_{vt}, D_{vt}, M_{vt}, T_{vt})^{V,T}_{v,t=1}. \tag{1}$$

The framework is end-to-end learnable, highly versatile for diverse input sequences and output tasks, and consistently effective as shown by experiments.

3.2 Multi-Setting Vision Input

Various Camera Settings. Dynamic scenes can be captured under diverse camera configurations, including monocular-static, monocular-dynamic, and multi-view static setups. These configurations create significant variations in input distributions due to differences in appearance and motion patterns, making it difficult for a unified model to perform consistently across different settings. Therefore, it is crucial to explore an adaptive and invariant feature representation to effectively handle various camera settings.

Adaptive Visual Grid. To explore the possibility of unifying inputs from different camera settings at the feature level, we analyze their feature distributions. We utilize the pretrained DINOv2 model [66] to encode image sequences captured under three camera settings from the same scene and project them with the same MLP before UMAP [67] visualization. As shown in Fig. 3, features from different views and time steps within the same setting, as well as features across different settings, cluster closely together with an average KL divergence of approximately 0.02. This demonstrates spatiotemporal consistency within a setting and cross-setting consistency after DINOv2 encoding. The former motivates our cross-view



(a) Spatiotemporal Representation (b) Spatiotemporal Attention Masks

Fig. 4: Illustration of spatiotemporal representation modules and attention masks. Visual tokens are input into the masked attention module, which performs different calculations guided by the attention masks. Spatial masks enable interactions among all tokens from different views within the same time step, while temporal masks restrict interactions to tokens from the same view within a temporal window.

and cross-time fusion modules (Sec. 3.3) for capturing spatiotemporal consistency, while the latter inspires the design of our adaptive visual grid to accommodate input sequences with arbitrary numbers of views and time steps.

For an input image sequence I_{vt}, we first use DINOv2 [66] to encode frames into image tokens t^I. In addition, to identify view and time information, we add a view token t^V and a time token t^T corresponding to each image token. The adaptive visual grid unifies arbitrary view-time inputs by organizing frames into a shared grid, where each frame is represented as:

$$t_{vt} = [t^I_{vt}, t^V_{vt}, t^T_{vt}], t \in [1, T], v \in [1, V], \tag{2}$$

This provides explicit (v, t) indices for each input frame and serves as the basis for subsequent modules to construct cross-view and cross-time interactions for spatial consistency and temporal continuity.

To further improve generalization, we adopt a random sampling strategy during the training stage. We randomly sample some tokens from the grid for each training step, simulating various configurations of input sequences. In summary, the adaptive visual grid enables 4D-VGGT to handle vision inputs with arbitrary views and time steps.

3.3 Multi-Level Feature Representation

Spatiotemporal Representation with Masked Attention. The spatiotemporal representation of input sequences under different camera settings requires

Table 1: Overview of our training datasets.

Training Stage	Task	Samples	Data Source
Stage-1: Per-Task Optim.	Camera Pose	7.1M	Co3Dv2 [68], WildRGB-D [69], ScanNet [70], Hypersim [71], Habitat [72], Replica [73]
	Depth	15.6M	BlendedMVS [74], DL3DV [75], MegaDepth [76], Kubric [77], MVS-Synth [78], Virtual KITTI 2 [79]
	Dynamic Mask	5.8M	Kubric [77], HOI4D [80], Dynamic-Replica [81]
	Point Map	1.6M	Co3Dv2 [68], MegaDepth [76], WildRGB-D [69], ScanNet [70]
	Tracking	1.2M	TAP-Vid [82]
Stage-2: Multi-Task Optim.	Multi-task	8.2M	TartanAir [83], Spring [84], Omniworld [85], SpatialVID [86]

distinct attention interactions. We adopt masked attention that uses specific masks to guide corresponding operations. As illustrated in Fig. 4 (b), six attention masks are designed for two modules under three typical camera settings. Specifically, in the cross-view global fusion module, the mask allows interactions among all tokens from different views within the same time step, while in the cross-time local fusion module, the attention mask restricts interactions to tokens in a fixed temporal window within the same view.

Cross-View Global Fusion. As shown in Fig. 4 (a), guided by the attention masks, the cross-view global fusion module performs two types of attention within each time step: intra-frame attention, which focuses on key areas within a single frame, and inter-frame attention, which captures spatial correlations across frames. After applying these two attention modules L times, we obtain the output spatial feature F^S. Note that time tokens are used only to identify the time step corresponding to each token and do not participate in attention computation. This module is formulated as:

$$f_{CV}([t^I, t^V]) = F^S. \tag{3}$$

Cross-Time Local Fusion. As shown in Fig. 4 (a), the cross-time local fusion module adopts temporal sliding window attention within each view. Guided by the attention mask, tokens attend to those within the same window of size S to capture continuous temporal cues. Given a specific view, tokens within a window centered at t are sequentially input to a GRU to obtain the hidden state h_t, which serves as the key and value for self-attention with the central query token t_t. After processing all tokens, the module outputs the temporal feature F^T. Note that view tokens only identify the corresponding view and do not participate in computation. This module is formulated as:

$$f_{CT}([t^I, t^T]) = F^T. \tag{4}$$

This factorized design aims to improve feature discriminability, by which we mean separating task-relevant spatial and temporal cues. Camera pose estimation mainly relies on cross-view spatial consistency, whereas point tracking primarily depends on cross-time motion continuity; for these cue-specific tasks, indiscriminately mixing spatial and temporal features may introduce interference and degrade accuracy. In contrast, dense prediction tasks such as depth, point map, and dynamic mask estimation benefit from jointly exploiting spatial structure and temporal dynamics, where complementary spatiotemporal cues provide more complete and robust geometry representations. By separating these

Table 2: Quantitative comparison on camera pose estimation.

Category	Method	Sintel (dynamic) [87]			TUM (dynamic) [88]			Bonn (dynamic) [89]		
		ATE ↓	RTE ↓	RRE ↓	ATE ↓	RTE ↓	RRE ↓	ATE ↓	RTE ↓	RRE ↓
Specialized models	DPVO [1]	0.176	0.066	1.315	0.018	0.014	0.398	0.023	0.016	0.927
	LEAP-VO [2]	**0.037**	0.067	1.692	0.026	0.030	2.811	0.038	0.016	0.857
	Robust-CVD [90]	0.372	0.156	3.486	0.099	0.028	2.573	0.087	0.019	0.818
	CasualSAM [91]	0.141	0.041	0.648	0.037	0.017	0.758	0.025	0.015	0.864
Foundation models (3D)	DUSt3R [13]	0.423	0.247	5.751	0.081	0.019	3.541	0.029	0.022	1.328
	MASt3R [34]	0.189	0.062	1.478	0.037	0.016	0.454	0.032	0.018	1.222
	VGGT [10]	0.164	0.061	0.503	0.013	0.015	0.319	0.021	0.023	0.937
	DA-3-Giant [42]	0.147	0.058	0.536	0.014	0.013	0.354	0.027	0.020	0.908
Foundation models (4D)	MonST3R [12]	0.111	0.045	0.741	0.094	0.021	1.359	0.024	0.014	0.795
	StreamVGGT [14]	0.209	0.060	0.551	0.041	0.017	0.432	0.029	0.030	1.171
	π3 [11]	0.074	0.040	0.282	0.014	0.009	0.312	0.020	0.023	0.835
	POMATO [49]	0.183	0.052	0.598	0.034	0.013	0.519	0.036	0.026	1.132
	PAGE-4D [59]	0.082	0.043	0.317	0.016	0.011	0.323	0.022	0.015	1.046
	Ours	0.065	**0.026**	**0.249**	**0.011**	**0.008**	**0.283**	**0.017**	**0.013**	**0.774**

task-relevant cues, the two modules provide more discriminative spatial and temporal representations for downstream geometry prediction.

3.4 Multi-task Joint Geometry Prediction

Task-Specific Heads. We design five different prediction heads for five geometry tasks: camera parameters, depth, dynamic mask, point map, and tracking estimation. For camera parameters, we propose the spatial head including self-attention layers and linear layers, which are used to derive camera intrinsics and extrinsics $\hat{g}_{vt}$ from the spatial feature F^S. For depth map, dynamic mask and point map, we first convert spatial feature F^S and temporal feature F^T into dense feature maps F^D with a DPT module [92] and apply three different prediction heads to respectively derive depth map $\hat{D}_{vt}$, dynamic mask $\hat{M}_{vt}$ and point map $\hat{P}_{vt}$. For tracking, we follow the design of TAPIR [7] and CoTracker3 [9]. We initialize N query points $(q_i)_{i=1}^N$ by randomly sampling from the first frame (or another specified frame). Subsequently, the temporal head utilizes the temporal feature F^T as input to derive the 2D trajectory $\hat{T}_{vt}^{2D}$ and 3D trajectory $\hat{T}_{vt}^{3D}$ corresponding to the query points q_i in a coarse-to-fine manner [7].

Multi-Task Joint Optimization. For each geometry estimation task, we design a specific loss function. To supervise camera parameter estimation, we apply the Huber loss $\rho_\delta(\cdot)$ between the prediction $\hat{g}_{vt}$ and the ground truth g_{vt}:

$$\mathcal{L}_{cam} = \sum\nolimits_{vt} \rho_\delta(\hat{g}_{vt} - g_{vt}). \tag{5}$$

The depth estimation process is supervised by the corresponding ground truth D_{vt} using a combination of an L2 loss and a gradient consistency loss:

$$\mathcal{L}_{depth} = \sum\nolimits_{vt} (\| \hat{D}_{vt} - D_{vt} \|_2^2 + \| \nabla \hat{D}_{vt} - \nabla D_{vt} \|). \tag{6}$$

Furthermore, dynamic mask estimation is supervised by the ground truth mask M_{vt} employing the binary cross-entropy loss:

$$\mathcal{L}_{mask} = -\frac{1}{N} \sum\nolimits_{vt} (M_{vt} \log \hat{M}_{vt} + (1 - M_{vt}) \log(1 - \hat{M}_{vt})), \tag{7}$$

Table 3: Quantitative comparison on depth estimation.

Category	Method	Sintel (dynamic) [87]		Bonn (dynamic) [89]		KITTI (dynamic) [94]	
		Abs Rel ↓	$\delta < 1.25$ ↑	Abs Rel ↓	$\delta < 1.25$ ↑	Abs Rel ↓	$\delta < 1.25$ ↑
Specialized models	NVDS [3]	0.417	47.9	0.171	77.0	0.247	59.3
	DepthCrafter [6]	0.286	70.5	0.078	96.8	0.108	88.5
	Robust-CVD [90]	0.716	48.1	0.203	70.1	0.282	57.4
	CasualSAM [91]	0.395	55.1	0.165	74.2	0.251	61.7
Foundation models (3D)	DUSt3R [13]	0.605	47.6	0.146	87.1	0.136	84.1
	MASt3R [34]	0.568	50.3	0.221	75.6	0.158	76.7
	VGGT [10]	0.309	66.7	0.056	97.0	0.073	94.7
	DA-3-Giant [42]	0.226	<u>72.8</u>	**0.037**	97.4	0.041	97.8
Foundation models (4D)	MonST3R [12]	0.339	59.2	0.061	96.0	0.106	89.2
	StreamVGGT [14]	0.364	66.3	0.084	89.7	0.132	85.0
	π3 [11]	<u>0.210</u>	72.6	0.043	<u>97.5</u>	<u>0.037</u>	**98.5**
	POMATO [49]	0.351	58.6	0.074	96.1	0.082	93.0
	PAGE-4D [59]	0.243	70.9	0.047	97.0	0.046	97.3
	Ours	**0.197**	**72.9**	<u>0.038</u>	**98.2**	**0.035**	<u>98.2</u>

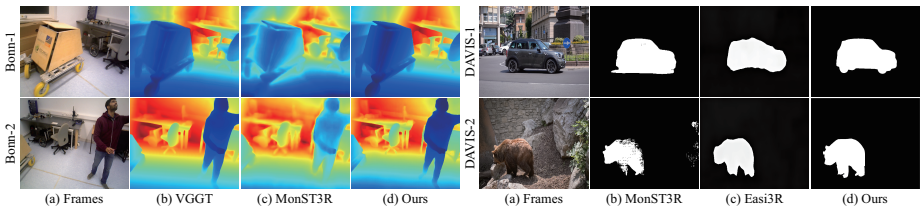


Fig. 5: Visual results of depth and dynamic mask estimation.

where the ground truth M_{vt} is binarized and the prediction $\hat{M}_{vt}$ is a probability value in the range of $[0, 1]$.

The point map estimation is jointly supervised by the ground truth coordinates P_{vt} using a combination of an L2 loss and a gradient consistency loss:

$$\mathcal{L}_{point} = \sum_{vt} (\| \hat{P}_{vt} - P_{vt} \|_2^2 + \| \nabla \hat{P}_{vt} - \nabla P_{vt} \|). \tag{8}$$

For tracking prediction, we calculate Chamfer Distance [93] between the prediction $\hat{T}_{vt}$ and ground truth T_{vt} as the tracking loss, which is formulated as:

$$\mathcal{L}_{track} = CD(\hat{T}_{vt}^{2D}, T_{vt}^{2D}) + CD(\hat{T}_{vt}^{3D}, T_{vt}^{3D}). \tag{9}$$

Overall, we train our 4D-VGGT with the multi-task loss formulated as:

$$\begin{aligned} \mathcal{L} = \lambda_{cam}\mathcal{L}_{cam} + \lambda_{depth}\mathcal{L}_{depth} + \lambda_{mask}\mathcal{L}_{mask} \\ + \lambda_{point}\mathcal{L}_{point} + \lambda_{track}\mathcal{L}_{track}, \end{aligned} \tag{10}$$

where λ_{cam}, λ_{depth}, λ_{mask}, λ_{point}, and λ_{track} are pre-defined weighting factors designed to balance the influence of the respective losses during training.

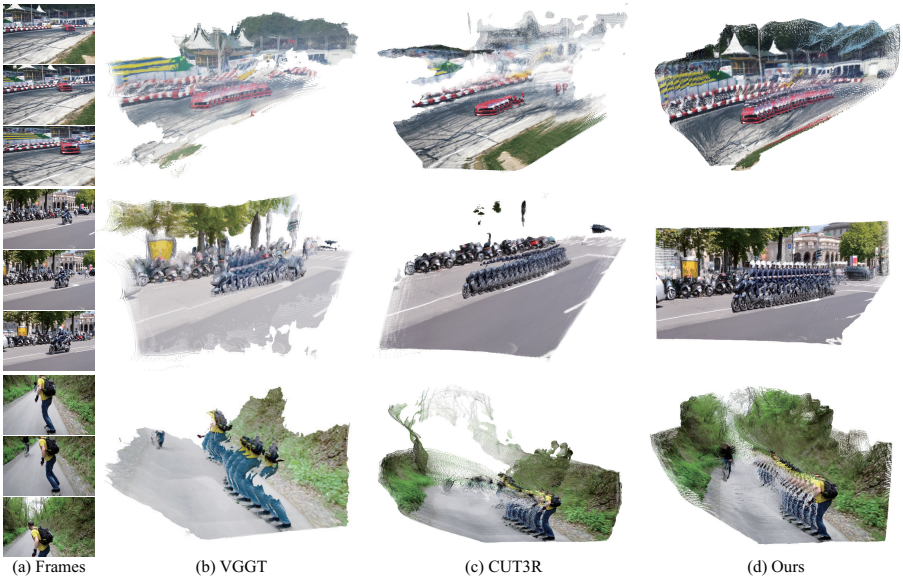
4 Training Datasets and Pipeline

4.1 Training Datasets

We train our 4D-VGGT model on a diverse collection of datasets, which provide a wide range of annotations including camera pose, depth, dynamic mask, point cloud and trajectory. The training datasets we used are shown in Tab. 1. For camera pose, we use datasets containing a total of 7.1 million samples, *e.g.*,

Table 4: Quantitative comparison on point map estimation.

Category	Method	7-Scenes (static) [95]						NRGBD (static) [96]						ETH3D (static) [97]					
		Acc. ↓		Comp. ↓		NC. ↑		Acc. ↓		Comp. ↓		NC. ↑		Acc. ↓		Comp. ↓		NC. ↑	
		Mean	Med.	Mean	Med.	Mean	Med.	Mean	Med.	Mean	Med.	Mean	Med.	Mean	Med.	Mean	Med.	Mean	Med.
3D	DUSt3R [13]	0.147	0.078	0.179	0.068	0.739	0.836	0.145	0.019	0.155	0.018	0.871	0.981	0.748	0.617	0.789	0.676	0.691	0.813
	Spann3R [35]	0.295	0.224	0.206	0.113	0.651	0.728	0.414	0.321	0.415	0.283	0.683	0.787	0.465	0.372	0.472	0.364	0.731	0.825
	VGGT [10]	0.087	**0.039**	0.092	0.039	0.788	0.889	0.072	0.018	0.078	0.021	0.911	0.989	0.281	0.186	0.306	0.183	0.852	0.949
	DA-3-Giant [42]	0.108	0.054	0.112	0.043	0.817	0.895	0.066	0.019	0.057	0.018	0.912	0.987	0.239	0.154	0.251	0.152	0.861	0.956
4D	MonST3R [12]	0.249	0.186	0.268	0.168	0.673	0.758	0.273	0.115	0.288	0.111	0.759	0.842	0.492	0.386	0.503	0.389	0.745	0.818
	CUT3R [51]	0.127	0.051	0.155	0.031	0.728	0.832	0.098	0.031	0.076	0.027	0.838	0.970	0.618	0.526	0.748	0.580	0.755	0.847
	StreamVGGT [14]	0.130	0.056	0.116	0.042	0.752	0.864	0.085	0.044	0.075	0.041	0.862	0.985	0.328	0.252	0.337	0.244	0.815	0.912
	π^3 [11]	0.095	0.049	0.108	0.052	0.776	0.880	0.055	0.017	**0.049**	0.017	0.915	0.974	0.195	0.132	0.211	0.128	**0.884**	0.968
	Mem4D [54]	0.184	0.136	0.177	0.082	0.740	0.814	0.270	0.195	0.213	0.114	0.734	0.823	0.478	0.365	0.491	0.359	0.738	0.809
	VGGT4D [58]	0.106	0.046	0.098	0.034	0.826	0.902	0.058	0.018	0.054	0.016	0.918	0.990	0.204	0.141	0.203	0.126	0.872	0.963
	PAGE-4D [59]	0.117	0.057	0.124	0.041	0.795	0.887	0.063	0.022	0.059	0.020	0.909	0.982	0.218	0.149	0.226	0.145	0.866	0.960
	Ours	**0.082**	0.048	**0.073**	**0.030**	**0.896**	**0.941**	**0.050**	**0.016**	0.052	**0.015**	**0.925**	**0.992**	**0.172**	**0.120**	**0.187**	**0.118**	0.879	**0.972**



(a) Frames (b) VGGT (c) CUT3R (d) Ours

Fig. 6: Visual results of point map estimation.

Co3Dv2 [68] and WildRGB-D [69]. For depth, we collect datasets including 15.6 million samples, *e.g.*, BlendedMVS [74] and DL3DV [75]. For dynamic mask, point map, and tracking, we select datasets containing 5.8 million, 1.6 million, and 1.2 million samples, respectively, including Kubric [77], MegaDepth [76], TAP-Vid [82], *etc.* For joint optimization, we use datasets containing a total of 8.2 million samples, *e.g.*, TartanAir [83] and Spring [84].

4.2 Training Pipeline

As shown in Tab. 1, our 4D-VGGT is trained in two separate stages: per-task optimization and multi-task optimization:

Stage 1: Per-Task Optimization. This stage equips the model with spatiotemporal representation and geometry estimation capabilities. We optimize the DINO encoder, feature representation modules, and prediction heads for each geometry task using their task-specific losses. Specifically, we train the model using camera

Table 5: Dynamic mask comparison.

Category	Method	DAVIS-16 [98]		DAVIS-17 [99]		SegTrackv2 [100]	
		$J_M \uparrow$	$J_R \uparrow$	$J_M \uparrow$	$J_R \uparrow$	$J_M \uparrow$	$J_R \uparrow$
Specialized models	OCLR [101]	67.4	78.2	62.1	69.8	64.7	68.5
	ABR [102]	70.9	81.3	64.8	73.5	**65.1**	**72.9**
Foundation models	D^2USt3R [44]	50.7	56.1	44.3	45.5	41.2	41.3
	MonST3R [12]	41.2	46.1	35.6	35.7	33.1	31.8
	Easi3R [50]	57.4	71.2	56.8	68.3	52.1	58.3
	POMATO [49]	49.5	55.2	43.4	44.3	43.1	43.9
	Ours	**71.0**	**82.2**	**67.1**	**75.0**	64.5	70.6

Table 6: Point tracking comparison.

Category	Method	PointOdyssey [103]		ADT [104]		PStudio [105]	
		L-12 ↑	L-24 ↑	L-12 ↑	L-24 ↑	L-12 ↑	L-24 ↑
Specialized models	TAPIR [7]	18.31	17.85	20.01	19.11	24.39	22.65
	SpatialTracker [8]	21.52	20.63	21.59	20.61	26.37	23.81
	CoTracker3 [9]	24.79	23.89	27.53	25.57	27.85	24.10
Foundation models	MonST3R [12]	27.35	27.88	28.27	26.09	16.48	11.02
	St4RTrack [48]	35.72	34.45	34.89	31.82	26.41	20.78
	POMATO [49]	33.18	33.54	31.61	28.26	24.63	19.83
	Ours	**36.12**	**34.98**	**35.27**	**32.11**	**28.79**	**25.03**

pose, depth, dynamic mask, point map and tracking tasks sequentially. Note that when training on one task, other prediction heads are frozen.

Stage 2: Multi-Task Optimization. This stage enhances the performance of multi-task joint prediction. We freeze the encoder and feature representation modules and fine-tune all prediction heads using the multi-task joint loss.

4.3 Implementation Details

Specifically, the size $V \times T$ of the view-time grid in the adaptive visual grid is determined by the number of views and time steps of the input sequence. At inference, the input modality type need not be predetermined. AVG assigns view and time tokens according to the available input indices, and the subsequent CVGF and CTLF modules automatically determine the corresponding cross-view and cross-time interactions. Thus, the model only requires the view-time organization of the input sequence, rather than explicit labels such as monocular static, monocular dynamic, or multi-view static. Regarding the attention mask, we directly generate two masks from this view-time organization to guide the subsequent two modules. During training, we perform random sampling based on the input sequence settings to simulate more possible settings. In the cross-view global fusion module, we repeat the intra-frame attention and the inter-frame attention for $L = 16$ times. In the cross-time local fusion module, we employ a sliding window of size $S = 5$ by default. For model training, we optimize the loss function shown in Eq. (10) using the AdamW optimizer with an initial learning rate of 10^{-5} and a weight decay of 0.01. For the weights in Eq. (10), we set $\lambda_{cam} = 1.0, \lambda_{depth} = 0.8, \lambda_{mask} = 0.8, \lambda_{point} = 0.9, \lambda_{track} = 0.1$. The training and evaluation are performed on 8 NVIDIA L20 GPUs.

5 Experiments

5.1 Comparison with State-of-the-Art Models

Comparison on Camera Pose. We comprehensively evaluate our proposed model for camera pose estimation on the Sintel [87], TUM-dynamics [88], and Bonn [89] datasets. Following LEAP-VO [2], we use the same evaluation split for Sintel, while all samples from TUM-dynamics and Bonn are included. Standard evaluation metrics include Absolute Translation Error (ATE), Relative Translation Error (RTE), and Relative Rotation Error (RRE). We compare our

Table 7: Ablation on proposed modules.

Modules			Camera pose			Point map			Tracking	
AVG	CVGF	CTLF	ATE ↓	RTE ↓	RRE ↓	Acc. ↓	Comp. ↓	NC. ↑	L-12 ↑	L-24 ↑
✗	✗	✗	0.045	0.038	2.095	0.291	0.329	0.714	17.36	16.41
✗	✓	✗	0.027	0.022	1.318	0.209	0.224	0.793	17.36	16.41
✗	✗	✓	0.045	0.038	2.095	0.234	0.256	0.762	25.81	23.68
✗	✓	✓	0.027	0.022	1.318	0.177	0.183	0.836	25.81	23.68
✓	✗	✗	0.030	0.026	1.492	0.215	0.237	0.780	24.76	23.43
✓	✓	✗	**0.017**	**0.013**	**0.774**	0.085	0.088	0.895	24.76	23.43
✓	✗	✓	0.030	0.026	1.492	0.091	0.094	0.883	**36.12**	**34.98**
✓	✓	✓	**0.017**	**0.013**	**0.774**	**0.050**	**0.052**	**0.925**	**36.12**	**34.98**

Table 8: Effect of hyperparameters.

Parameter	Value	Camera pose			Point map			Tracking		Inference Time (s)
		ATE ↓	RTE ↓	RRE ↓	Acc. ↓	Comp. ↓	NC. ↑	L-12 ↑	L-24 ↑	
Layers L in CVGF	12	0.022	0.021	0.923	0.068	0.069	0.894	36.12	34.98	2.56
	16	0.017	0.013	0.774	0.050	0.052	0.925	36.12	34.98	3.19
	20	0.016	0.014	0.768	0.048	0.049	0.929	36.12	34.98	3.85
Window size S in CTLF	3	0.017	0.013	0.774	0.071	0.074	0.908	31.52	29.83	2.06
	5	0.017	0.013	0.774	0.050	0.052	0.925	36.12	34.98	3.19
	7	0.017	0.013	0.774	0.049	0.052	0.931	36.43	35.77	4.32
	9	0.017	0.013	0.774	0.050	0.051	0.928	36.67	36.42	5.27

model with several specialized pose estimation methods, including DPVO [1] and LEAP-VO, Robust-CVD [90], and CasualSAM [91]. Moreover, we compare against recent foundation models for static scene geometry, such as DUSt3R [13], MASt3R [34], and VGGT [10], as well as those for dynamic scene geometry, including MonST3R [12], StreamVGGT [14], and POMATO [49].

As shown in Tab. 2, 4D-VGGT achieves consistently strong performance across all datasets and metrics, outperforming existing specialized models and foundation models for both static and dynamic scene geometry.

Comparison on Depth. Following DepthCrafter [6], we evaluate depth estimation on Sintel, Bonn, and KITTI [94]. We apply per-scene alignment to ensure scale- and shift-invariant comparisons. For qualitative evaluation, we visualize depth maps on Bonn dataset. For quantitative evaluation, we report the absolute relative error (Abs Rel) and the inlier ratio ($\delta < 1.25$). Our model is compared with specialized methods (NVDS [3], DepthCrafter, Robust-CVD, CasualSAM) and foundation models following the protocol as in the camera pose evaluation.

As shown in Fig. 5 and Tab. 3, our method achieves superior qualitative and quantitative results. It estimates accurate global structure and fine local details, and outperforms both specialized and foundation models across all datasets, demonstrating its strong depth estimation capability.

Comparison on Dynamic Mask. We systematically evaluate dynamic mask estimation on the widely-used DAVIS-16 [98], DAVIS-17 [99], and SegTrackv2 [100] datasets. Following DAVIS [98], we adopt the mean IoU (J_M) and the recall of IoU (J_R) as our primary evaluation metrics. We compare our model with both specialized models including OCLR [101] and ABR [102], as well as recent foundation models for dynamic scene geometry, including MonST3R, D^{2}USt3R [44], Easi3R [50], and POMATO.

As shown in Fig. 5 and Tab. 5, our 4D-VGGT surpasses all foundation models and achieves competitive performance with specialized segmentation methods.

Comparison on Point Map. We evaluate point map accuracy on the 7-Scenes [95], NRGBD [96], and ETH3D [97] datasets. Following VGGT and CUT3R [51], we use sparsely sampled inputs: 3–5 frames per scene for 7-Scenes, 2–4 for NRGBD, and 10 for ETH3D. Quantitative metrics include Accuracy (Acc.), Completion (Comp.), and Normal Consistency (NC.), with point maps aligned to ground truth via the Umeyama algorithm [106]. Our model is compared against foundation models for static scene geometry, including DUSt3R, Spann3R [35], VGGT, and π3 [11], as well as those for dynamic scene geometry, including MonST3R, CUT3R, StreamVGGT, and Mem4D [54].

Table 9: Influence of training strategy.

Training strategy	Camera pose			Point map			Tracking	
	ATE ↓	RTE ↓	RRE ↓	Acc. ↓	Comp. ↓	NC. ↑	L-12 ↑	L-24 ↑
w/SS	0.026	0.024	1.291	0.093	0.087	0.853	26.61	24.93
w/SS+Sampling	0.024	0.021	1.048	0.081	0.083	0.877	28.59	25.86
w/TS	0.020	0.015	0.854	0.065	0.066	0.905	33.49	32.07
w/TS+Sampling	**0.017**	**0.013**	**0.774**	**0.050**	**0.052**	**0.925**	**36.12**	**34.98**

Table 10: Impact of spatial and temporal masks.

Attention mask	Camera pose			Point map			Tracking	
	ATE ↓	RTE ↓	RRE ↓	Acc. ↓	Comp. ↓	NC. ↑	L-12 ↑	L-24 ↑
w/o	0.024	0.022	1.127	0.133	0.141	0.882	29.53	28.41
w/Spatial-mask	**0.017**	**0.013**	**0.774**	0.075	0.077	0.901	29.53	28.41
w/Temporal-mask	0.024	0.022	1.127	0.082	0.089	0.892	**36.12**	**34.98**
w/Both	**0.017**	**0.013**	**0.774**	**0.050**	**0.052**	**0.925**	**36.12**	**34.98**

As illustrated in Fig. 6 and Tab. 4, 4D-VGGT delivers visually superior point maps and achieves state-of-the-art performance across most metrics, consistently outperforming all static and dynamic geometry estimation methods.

Comparison on Tracking. We evaluate the tracking performance on PointOdyssey [103], Aria Digital Twin (ADT) [104], and Panoptic Studio (PStudio) [105]. Following prior work, we report Average Percent Deviation (APD) for trajectories of 12 and 24 frames (L-12 and L-24). We compare our model with specialized models such as TAPIR [7], SpatialTracker [8], and CoTracker3 [9], as well as foundation models, *e.g.*, MonST3R, St4RTrack [48], and POMATO.

As illustrated in Fig. 7 and Tab. 6, our 4D-VGGT produces smooth, coherent 2D and 3D trajectories and surpasses all specialized and foundation models.

5.2 Ablation Study

How Proposed Modules Work. We conduct ablation studies on camera pose, point map, and point tracking to evaluate the impact of the Adaptive Visual Grid (AVG), Cross-View Global Fusion (CVGF), and Cross-Time Local Fusion (CTLF) modules. The ablation configuration is explained below: 1) w/o AVG: image tokens are treated as flat sequences, lacking spatiotemporal identifiers (t^V, t^T); 2) w/o CVGF: cross-view fusion is replaced by inter-frame global self-attention [10], without spatial structural priors; 3) w/o CTLF: sliding-window modeling is replaced by intra-frame self-attention [10], ignoring motion continuity. As shown in Tab. 7, both modules significantly improve the performance of point map estimation, while CVGF and CTLF further enhance camera pose estimation and point tracking, respectively.

Influence of Training Strategy. We further conduct ablation experiments to verify the effectiveness of our training strategy. Our model adopts a two-stage (TS) manner and a random sampling strategy. We compare our training paradigm with those that employ single-stage (SS) training or without the random sampling strategy. Specifically, the single-stage baseline was actually trained on the full collection of datasets by mixing batches across all tasks. As shown in Tab. 9, both of our training strategies lead to notable performance gains across all tasks. Our two-stage training strategy functions as a progressive grounding approach: Stage-1 provides progressive grounding and mitigates gradient interference, which allows Stage-2 to refine the model effectively. Moreover, our random sampling strategy equips the model with stronger robustness.

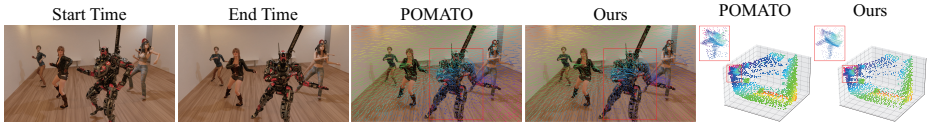
Ablation on input features for prediction heads. To validate the effectiveness of our divide-and-conquer spatiotemporal representation strategy across different geometric tasks, we ablate the input features for the prediction heads.

Table 11: Ablation on features for prediction.

Input features	Camera pose estimation				Tracking		
	ATE ↓	RTE ↓	RRE ↓	Time (ms) ↓	L-12 ↑	L-24 ↑	Time (ms) ↓
w/ Spatial	**0.039**	**0.017**	**0.462**	**86**	19.71	16.53	**253**
w/ Temporal	0.107	0.054	0.862	112	33.39	30.71	287
w/ Both	0.045	0.023	0.547	197	**33.87**	**31.02**	416

Table 12: Generalization for camera settings.

Camera setting	Camera pose			Point map			Tracking	
	ATE ↓	RTE ↓	RRE ↓	Acc. ↓	Comp. ↓	NC. ↑	L-12 ↑	L-24 ↑
Mono-S	0.023	0.019	0.920	0.063	0.065	0.797	30.90	29.45
Mono-D	0.021	0.017	0.880	0.059	0.061	0.815	31.35	30.05
Multi-S	0.021	0.016	0.845	0.058	0.058	0.832	32.72	31.60



(a) Input Frames (b) 2D Trajectory (c) 3D Trajectory

Fig. 7: Visual results of point tracking on the PointOdyssey dataset.

As shown in Tab. 11, for camera pose estimation, relying solely on spatial features achieves the best accuracy and efficiency, whereas introducing temporal features degrades performance. Conversely, for point tracking, temporal features alone provide strong results; integrating spatial features yields only marginal performance gains but substantially increases inference time (from 287 ms to 416 ms). These findings demonstrate that utilizing task-specific features achieves the optimal trade-off between performance and computational cost.

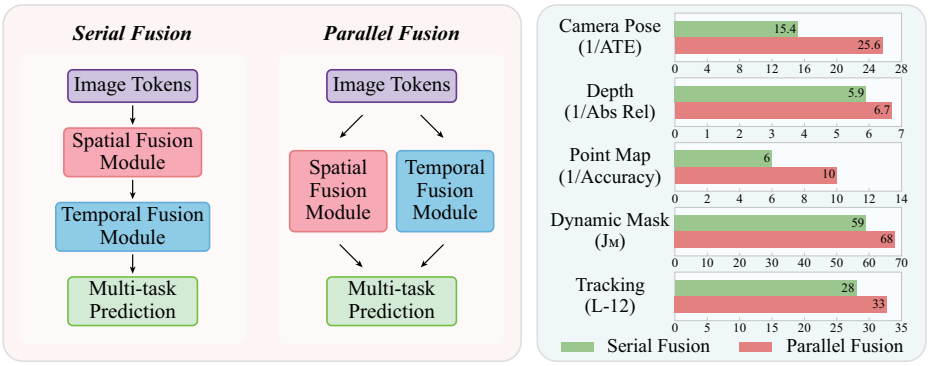
5.3 Discussion

Effectiveness of Spatiotemporal Representation Strategy. A key difference between 4D-VGGT and existing 4D geometry estimation methods lies in the parallel fusion strategy for spatiotemporal representation, while prior methods adopt the serial fusion strategy. To verify its effectiveness, we evaluate the performance across multiple geometry estimation tasks. As shown in Fig. 8, the parallel fusion strategy yields notably better performance, particularly on tasks sensitive to dynamic patterns, *e.g.*, camera pose and point map.

Impact of Attention Masks. We evaluate the effect of attention masks in the spatiotemporal representation module through ablation on two mask types: the spatial mask and the temporal mask. Tab. 10 shows that both masks improve point map estimation performance, while the spatial and temporal masks notably enhance camera pose estimation and point tracking, respectively.

Effect of Hyperparameters. To determine the optimal hyperparameters for the spatiotemporal representation module, we conduct additional quantitative experiments. As shown in Tab. 8, increasing the number of layers L in CVGF leads to performance improvements but higher inference time. Therefore, we set $L = 16$ to achieve the best trade-off between accuracy and efficiency. For CTLF, we adopt the same principle and set the window size $S = 5$.

Generalization on Camera Settings. To verify the generalization to different camera settings, we conduct experiments on the same scenes under multi-view static (Multi-S), monocular static (Mono-S), and monocular dynamic (Mono-D) settings. As shown in Tab. 12, our model maintains consistent performance across all tasks, demonstrating strong generalization to diverse camera settings.



(a) Spatiotemporal Representation Strategy (b) Performance Comparison

Fig. 8: Analysis on spatiotemporal representation strategy. To validate the effectiveness of our parallel fusion strategy, we compare it with serial fusion across multiple geometry tasks. As shown in the bar chart, our strategy consistently yields superior performance in all tasks.

Limitations. While our model effectively captures dynamic scene geometry from various camera settings, it still faces challenges. Due to the fixed window size in the CTLF, the motion patterns of fast-moving objects may not be fully captured. In future work, we aim to integrate event cameras [107, 108] to provide visual signals with higher temporal resolution and better capture rapid dynamics.

6 Conclusion

In this work, we propose 4D-VGGT, a general spatiotemporal foundation model for dynamic scene geometry estimation. We design a multi-setting input component, where we introduce an adaptive visual grid to support inputs from arbitrary views and time steps. We propose a multi-level representation component, which achieves a divide-and-conquer spatiotemporal representation through attention masks. We construct a multi-task prediction component, which builds multiple task-specific heads to learn the dynamic scene geometry via multi-task optimization. The three components jointly promote the feature discriminability and application universality for dynamic scenes. We further integrate diverse datasets to train our model and validate the effectiveness of our method through extensive experiments. We believe that this work provides a new paradigm for the research community on multi-task prediction in dynamic scenes.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China under Grant U24B20139 and the Hubei Provincial Natural Science Foundation of China under Grant 2026AFA040. The computations were performed on the HPC Platform of Huazhong University of Science and Technology. We also thank the reviewers for their valuable comments and suggestions.

References

1. Zachary Teed, Lahav Lipson, and Jia Deng. Deep patch visual odometry. *Advances in Neural Information Processing Systems*, 36:39033–39051, 2023. 1, 8, 12
2. Weirong Chen, Le Chen, Rui Wang, and Marc Pollefeys. Leap-vo: Long-term effective any point tracking for visual odometry. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19844–19853, 2024. 1, 8, 11
3. Yiran Wang, Min Shi, Jiaqi Li, Zihao Huang, Zhiguo Cao, Jianming Zhang, Ke Xian, and Guosheng Lin. Neural video depth stabilizer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9466–9476, 2023. 1, 9, 12
4. Lihe Yang, Bingyi Kang, Zilong Huang, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything: Unleashing the power of large-scale unlabeled data. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10371–10381, 2024. 1
5. Sili Chen, Hengkai Guo, Shengnan Zhu, Feihu Zhang, Zilong Huang, Jiashi Feng, and Bingyi Kang. Video depth anything: Consistent depth estimation for super-long videos. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 22831–22840, 2025. 1
6. Wenbo Hu, Xiangjun Gao, Xiaoyu Li, Sijie Zhao, Xiaodong Cun, Yong Zhang, Long Quan, and Ying Shan. Depthcrafter: Generating consistent long depth sequences for open-world videos. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 2005–2015, 2025. 1, 9, 12
7. Carl Doersch, Yi Yang, Mel Vecerik, Dilara Gokay, Ankush Gupta, Yusuf Aytar, Joao Carreira, and Andrew Zisserman. Tapir: Tracking any point with per-frame initialization and temporal refinement. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10061–10072, 2023. 1, 8, 11, 13
8. Yuxi Xiao, Qianqian Wang, Shangzhan Zhang, Nan Xue, Sida Peng, Yujun Shen, and Xiaowei Zhou. Spatialtracker: Tracking any 2d pixels in 3d space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20406–20417, 2024. 1, 11, 13
9. Nikita Karaev, Iurii Makarov, Jianyuan Wang, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. Cotracker3: Simpler and better point tracking by pseudo-labelling real videos. *arXiv preprint arXiv:2410.11831*, 2024. 1, 8, 11, 13
10. Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 5294–5306, 2025. 1, 2, 3, 8, 9, 10, 12, 13
11. Yifan Wang, Jianjun Zhou, Haoyi Zhu, Wenzheng Chang, Yang Zhou, Zizun Li, Junyi Chen, Jiangmiao Pang, Chunhua Shen, and Tong He. pi3: Scalable permutation-equivariant visual geometry learning. *arXiv preprint arXiv:2507.13347*, 2025. 1, 3, 8, 9, 10, 12
12. 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. *arXiv preprint arXiv:2410.03825*, 2024. 2, 3, 8, 9, 10, 11, 12
13. 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, 3, 8, 9, 10, 12
14. Dong Zhuo, Wenzhao Zheng, Jiahe Guo, Yuqi Wu, Jie Zhou, and Jiwen Lu. Streaming 4d visual geometry transformer. *arXiv preprint arXiv:2507.11539*, 2025. 2, 4, 8, 9, 10, 12
15. Sameer Agarwal, Yasutaka Furukawa, Noah Snavely, Ian Simon, Brian Curless, Steven M Seitz, and Richard Szeliski. Building rome in a day. *Communications of the ACM*, 54(10):105–112, 2011. 3
16. Jan-Michael Frahm, Pierre Fite-Georgel, David Gallup, Tim Johnson, Rahul Raguram, Changchang Wu, Yi-Hung Jen, Enrique Dunn, Brian Clipp, Svetlana Lazebnik, et al. Building rome on a cloudless day. In *European conference on computer vision*, pages 368–381. Springer, 2010. 3
17. Shaohui Liu, Yidan Gao, Tianyi Zhang, Rémi Pautrat, Johannes L Schönberger, Viktor Larsson, and Marc Pollefeys. Robust incremental structure-from-motion with hybrid features. In *European Conference on Computer Vision*, pages 249–269. Springer, 2024. 3
18. Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4104–4113, 2016. 3
19. Changchang Wu. Towards linear-time incremental structure from motion. In *2013 International Conference on 3D Vision-3DV 2013*, pages 127–134. IEEE, 2013. 3
20. Yasutaka Furukawa, Carlos Hernández, et al. Multi-view stereo: A tutorial. *Foundations and trends® in Computer Graphics and Vision*, 9(1-2):1–148, 2015. 3
21. Silvano Galliani, Katrin Lasinger, and Konrad Schindler. Massively parallel multiview stereopsis by surface normal diffusion. In *Proceedings of the IEEE international conference on computer vision*, pages 873–881, 2015. 3
22. Johannes L Schönberger, Enliang Zheng, Jan-Michael Frahm, and Marc Pollefeys. Pixelwise view selection for unstructured multi-view stereo. In *European conference on computer vision*, pages 501–518. Springer, 2016. 3
23. Yuesong Wang, Zhaojie Zeng, Tao Guan, Wei Yang, Zhuo Chen, Wenkai Liu, Luoyuan Xu, and Yawei Luo. Adaptive patch deformation for textureless-resilient multi-view stereo. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1621–1630, 2023. 3
24. Zachary Teed and Jia Deng. Droid-slam: Deep visual slam for monocular, stereo, and rgb-d cameras. *Advances in neural information processing systems*, 34:16558–16569, 2021. 3
25. Raul Mur-Artal, Jose Maria Martinez Montiel, and Juan D Tardos. Orb-slam: A versatile and accurate monocular slam system. *IEEE transactions on robotics*, 31(5):1147–1163, 2015. 3
26. Raul Mur-Artal and Juan D Tardós. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE transactions on robotics*, 33(5):1255–1262, 2017. 3
27. Richard A Newcombe, Steven J Lovegrove, and Andrew J Davison. Dtam: Dense tracking and mapping in real-time. In *2011 international conference on computer vision*, pages 2320–2327. IEEE, 2011. 3
28. Carlos Campos, Richard Elvira, Juan J Gómez Rodríguez, José MM Montiel, and Juan D Tardós. Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam. *IEEE transactions on robotics*, 37(6):1874–1890, 2021. 3

29. Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4938–4947, 2020. 3
30. Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. Loftr: Detector-free local feature matching with transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8922–8931, 2021. 3
31. Xingkui Wei, Yinda Zhang, Zhuwen Li, Yanwei Fu, and Xiangyang Xue. Deepsfm: Structure from motion via deep bundle adjustment. In *European conference on computer vision*, pages 230–247. Springer, 2020. 3
32. Jianyuan Wang, Nikita Karaev, Christian Rupprecht, and David Novotny. Vggsfm: Visual geometry grounded deep structure from motion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 21686–21697, 2024. 3
33. Hanyu Zhou, Haonan Wang, Haoyue Liu, Yuxing Duan, Luxin Yan, and Gim Hee Lee. Std-gs: Exploring frame-event interaction for spatiotemporal-disentangled gaussian splatting to reconstruct high-dynamic scene. *arXiv preprint arXiv:2506.23157*, 2025. 3
34. Vincent Leroy, Yohann Cabon, and Jérôme Revaud. Grounding image matching in 3d with mast3r. In *European Conference on Computer Vision*, pages 71–91. Springer, 2024. 3, 8, 9, 12
35. Hengyi Wang and Lourdes Agapito. 3d reconstruction with spatial memory. *arXiv preprint arXiv:2408.16061*, 2024. 3, 10, 12
36. Jianing Yang, Alexander Sax, Kevin J Liang, Mikael Henaff, Hao Tang, Ang Cao, Joyce Chai, Franziska Meier, and Matt Feiszli. Fast3r: Towards 3d reconstruction of 1000+ images in one forward pass. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 21924–21935, 2025. 3
37. Zhenggang Tang, Yuchen Fan, Dilin Wang, Hongyu Xu, Rakesh Ranjan, Alexander Schwing, and Zhicheng Yan. Mv-dust3r+: Single-stage scene reconstruction from sparse views in 2 seconds. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 5283–5293, 2025. 3
38. Shangzhan Zhang, Jianyuan Wang, Yinghao Xu, Nan Xue, Christian Rupprecht, Xiaowei Zhou, Yujun Shen, and Gordon Wetzstein. Flare: Feed-forward geometry, appearance and camera estimation from uncalibrated sparse views. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 21936–21947, 2025. 3
39. Sven Elflein, Qunjie Zhou, and Laura Leal-Taixé. Light3r-sfm: Towards feed-forward structure-from-motion. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 16774–16784, 2025. 3
40. Yuzheng Liu, Siyan Dong, Shuzhe Wang, Yingda Yin, Yanchao Yang, Qingnan Fan, and Baoquan Chen. Slam3r: Real-time dense scene reconstruction from monocular rgb videos. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 16651–16662, 2025. 3
41. 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. 3
42. Haotong Lin, Sili Chen, Junhao Liew, Donny Y Chen, Zhenyu Li, Guang Shi, Jiashi Feng, and Bingyi Kang. Depth anything 3: Recovering the visual space from any views. *arXiv preprint arXiv:2511.10647*, 2025. 3, 8, 9, 10

43. Linyi Jin, Richard Tucker, Zhengqi Li, David Fouhey, Noah Snavely, and Aleksander Holynski. Stereo4d: Learning how things move in 3d from internet stereo videos. *arXiv preprint arXiv:2412.09621*, 2024. 3
44. Jisang Han, Honggyu An, Jaewoo Jung, Takuya Narihira, Junyoung Seo, Kazumi Fukuda, Chaehyun Kim, Sunghwan Hong, Yuki Mitsufuji, and Seungryong Kim. Dˆ 2ust3r: Enhancing 3d reconstruction with 4d pointmaps for dynamic scenes. *arXiv preprint arXiv:2504.06264*, 2025. 3, 11, 12
45. Jinjie Mai, Wenxuan Zhu, Haozhe Liu, Bing Li, Cheng Zheng, Jürgen Schmidhuber, and Bernard Ghanem. Can video diffusion model reconstruct 4d geometry? *arXiv preprint arXiv:2503.21082*, 2025. 3
46. Edgar Sucar, Zihang Lai, Eldar Insafutdinov, and Andrea Vedaldi. Dynamic point maps: A versatile representation for dynamic 3d reconstruction. *arXiv preprint arXiv:2503.16318*, 2025. 3
47. Zeren Jiang, Chuanxia Zheng, Iro Laina, Diane Larlus, and Andrea Vedaldi. Geo4d: Leveraging video generators for geometric 4d scene reconstruction. *arXiv preprint arXiv:2504.07961*, 2025. 3
48. Haiwen Feng, Junyi Zhang, Qianqian Wang, Yufei Ye, Pengcheng Yu, Michael J Black, Trevor Darrell, and Angjoo Kanazawa. St4rtrack: Simultaneous 4d reconstruction and tracking in the world. *arXiv preprint arXiv:2504.13152*, 2025. 3, 11, 13
49. Songyan Zhang, Yongtao Ge, Jinyuan Tian, Guangkai Xu, Hao Chen, Chen Lv, and Chunhua Shen. Pomato: Marrying pointmap matching with temporal motion for dynamic 3d reconstruction. *arXiv preprint arXiv:2504.05692*, 2025. 3, 8, 9, 11, 12
50. Xingyu Chen, Yue Chen, Yuliang Xiu, Andreas Geiger, and Anpei Chen. Easi3r: Estimating disentangled motion from dust3r without training. *arXiv preprint arXiv:2503.24391*, 2025. 3, 11, 12
51. Qianqian Wang, Yifei Zhang, Aleksander Holynski, Alexei A Efros, and Angjoo Kanazawa. Continuous 3d perception model with persistent state. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 10510–10522, 2025. 4, 10, 12
52. Yohann Cabon, Lucas Stoffl, Leonid Antsfeld, Gabriela Csurka, Boris Chidlovskii, Jerome Revaud, and Vincent Leroy. Must3r: Multi-view network for stereo 3d reconstruction. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 1050–1060, 2025. 4
53. Yuqi Wu, Wenzhao Zheng, Jie Zhou, and Jiwen Lu. Point3r: Streaming 3d reconstruction with explicit spatial pointer memory. *arXiv preprint arXiv:2507.02863*, 2025. 4
54. Xudong Cai, Shuo Wang, Peng Wang, Yongcai Wang, Zhaoxin Fan, Wanting Li, Tianbao Zhang, Jianrong Tao, Yeying Jin, and Deying Li. Mem4d: Decoupling static and dynamic memory for dynamic scene reconstruction. *arXiv preprint arXiv:2508.07908*, 2025. 4, 10, 12
55. Chuhan Zhang, Guillaume Le Moing, Skanda Koppula, Ignacio Rocco, Liliane Momeni, Junyu Xie, Shuyang Sun, Rahul Sukthankar, Joëlle K Barral, Raia Hadsell, et al. Efficiently reconstructing dynamic scenes one d4rt at a time. *arXiv preprint arXiv:2512.08924*, 2025. 4
56. Shizun Wang, Zhenxiang Jiang, Xingyi Yang, and Xinchao Wang. C4d: 4d made from 3d through dual correspondences. 2025. 4
57. Xiaoxue Chen, Ziyi Xiong, Yuantao Chen, Gen Li, Nan Wang, Hongcheng Luo, Long Chen, Haiyang Sun, Bing Wang, Guang Chen, et al. Dggt: Feedforward 4d

reconstruction of dynamic driving scenes using unposed images. *arXiv preprint arXiv:2512.03004*, 2025. 4

58. Yu Hu, Chong Cheng, Sicheng Yu, Xiaoyang Guo, and Hao Wang. Vggt4d: Mining motion cues in visual geometry transformers for 4d scene reconstruction. *arXiv preprint arXiv:2511.19971*, 2025. 4, 10
59. Kaichen Zhou, Yuhan Wang, Grace Chen, Xinhai Chang, Gaspard Beaudouin, Fangneng Zhan, Paul Pu Liang, and Mengyu Wang. Page-4d: Disentangled pose and geometry estimation for vggt-4d perception. *arXiv preprint arXiv:2510.17568*, 2025. 4, 8, 9, 10
60. Amir R Zamir, Alexander Sax, William Shen, Leonidas J Guibas, Jitendra Malik, and Silvio Savarese. Taskonomy: Disentangling task transfer learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3712–3722, 2018. 4
61. Bin Xiao, Haiping Wu, Weijian Xu, Xiyang Dai, Houdong Hu, Yumao Lu, Michael Zeng, Ce Liu, and Lu Yuan. Florence-2: Advancing a unified representation for a variety of vision tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4818–4829, 2024. 4
62. Siddharth Srivastava and Gaurav Sharma. Omnivec2-a novel transformer based network for large scale multimodal and multitask learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 27412–27424, 2024. 4
63. Hanrong Ye and Dan Xu. Joint 2d-3d multi-task learning on cityscapes-3d: 3d detection, segmentation, and depth estimation. *arXiv preprint arXiv:2304.00971*, 2023. 4
64. Hanyu Zhou and Gim Hee Lee. Uni4d-llm: A unified spatiotemporal-aware vlm for 4d understanding and generation. *arXiv preprint arXiv:2509.23828*, 2025. 4
65. Hanyu Zhou and Gim Hee Lee. Llava-4d: Embedding spatiotemporal prompt into lmms for 4d scene understanding. *arXiv preprint arXiv:2505.12253*, 2025. 4
66. Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023. 5, 6
67. Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018. 5
68. Jeremy Reizenstein, Roman Shapovalov, Philipp Henzler, Luca Sbordone, Patrick Labatut, and David Novotny. Common objects in 3d: Large-scale learning and evaluation of real-life 3d category reconstruction. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10901–10911, 2021. 7, 10
69. Hongchi Xia, Yang Fu, Sifei Liu, and Xiaolong Wang. Rgbd objects in the wild: Scaling real-world 3d object learning from rgb-d videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22378–22389, 2024. 7, 10
70. Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5828–5839, 2017. 7
71. Mike Roberts, Jason Ramapuram, Anurag Ranjan, Atulit Kumar, Miguel Angel Bautista, Nathan Paczan, Russ Webb, and Joshua M Susskind. Hypersim: A

photorealistic synthetic dataset for holistic indoor scene understanding. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10912–10922, 2021. 7

72. Santhosh K Ramakrishnan, Aaron Gokaslan, Erik Wijmans, Oleksandr Maksymets, Alex Clegg, John Turner, Eric Undersander, Wojciech Galuba, Andrew Westbury, Angel X Chang, et al. Habitat-matterport 3d dataset (hm3d): 1000 large-scale 3d environments for embodied ai. *arXiv preprint arXiv:2109.08238*, 2021. 7
73. Julian Straub, Thomas Whelan, Lingni Ma, Yufan Chen, Erik Wijmans, Simon Green, Jakob J Engel, Raul Mur-Artal, Carl Ren, Shobhit Verma, et al. The replica dataset: A digital replica of indoor spaces. *arXiv preprint arXiv:1906.05797*, 2019. 7
74. Yao Yao, Zixin Luo, Shiwei Li, Jingyang Zhang, Yufan Ren, Lei Zhou, Tian Fang, and Long Quan. Blendedmvs: A large-scale dataset for generalized multi-view stereo networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1790–1799, 2020. 7, 10
75. Lu Ling, Yichen Sheng, Zhi Tu, Wentian Zhao, Cheng Xin, Kun Wan, Lantao Yu, Qianyu Guo, Zixun Yu, Yawen Lu, et al. Dl3dv-10k: A large-scale scene dataset for deep learning-based 3d vision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22160–22169, 2024. 7, 10
76. Zhengqi Li and Noah Snavely. Megadepth: Learning single-view depth prediction from internet photos. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2041–2050, 2018. 7, 10
77. Klaus Greff, Francois Belletti, Lucas Beyer, Carl Doersch, Yilun Du, Daniel Duckworth, David J Fleet, Dan Gnanapragasam, Florian Golemo, Charles Herrmann, et al. Kubric: A scalable dataset generator. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3749–3761, 2022. 7, 10
78. Po-Han Huang, Kevin Matzen, Johannes Kopf, Narendra Ahuja, and Jia-Bin Huang. Deepmvs: Learning multi-view stereopsis. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2821–2830, 2018. 7
79. Yohann Cabon, Naila Murray, and Martin Humenberger. Virtual kitti 2. *arXiv preprint arXiv:2001.10773*, 2020. 7
80. Yunze Liu, Yun Liu, Che Jiang, Kangbo Lyu, Weikang Wan, Hao Shen, Boqiang Liang, Zhoujie Fu, He Wang, and Li Yi. Hoi4d: A 4d egocentric dataset for category-level human-object interaction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21013–21022, 2022. 7
81. Nikita Karaev, Ignacio Rocco, Benjamin Graham, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. Dynamicstereo: Consistent dynamic depth from stereo videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13229–13239, 2023. 7
82. Carl Doersch, Ankush Gupta, Larisa Markeeva, Adria Recasens, Lucas Smaira, Yusuf Aytar, Joao Carreira, Andrew Zisserman, and Yi Yang. Tap-vid: A benchmark for tracking any point in a video. *Advances in Neural Information Processing Systems*, 35:13610–13626, 2022. 7, 10
83. 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. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4909–4916. IEEE, 2020. 7, 10
84. Lukas Mehl, Jenny Schmalfuss, Azin Jahedi, Yaroslava Nalivayko, and Andrés Bruhn. Spring: A high-resolution high-detail dataset and benchmark for scene

flow, optical flow and stereo. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4981–4991, 2023. 7, 10
85. Yang Zhou, Yifan Wang, Jianjun Zhou, Wenzheng Chang, Haoyu Guo, Zizun Li, Kaijing Ma, Xinyue Li, Yating Wang, Haoyi Zhu, et al. Omniworld: A multi-domain and multi-modal dataset for 4d world modeling. *arXiv preprint arXiv:2509.12201*, 2025. 7
86. Jiahao Wang, Yufeng Yuan, Rujie Zheng, Youtian Lin, Jian Gao, Lin-Zhuo Chen, Yajie Bao, Yi Zhang, Chang Zeng, Yanxi Zhou, et al. Spatialvid: A large-scale video dataset with spatial annotations. *arXiv preprint arXiv:2509.09676*, 2025. 7
87. Daniel J Butler, Jonas Wulff, Garrett B Stanley, and Michael J Black. A naturalistic open source movie for optical flow evaluation. In *European conference on computer vision*, pages 611–625. Springer, 2012. 8, 9, 11
88. Jürgen Sturm, Nikolas Engelhard, Felix Endres, Wolfram Burgard, and Daniel Cremers. A benchmark for the evaluation of rgb-d slam systems. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, pages 573–580. IEEE, 2012. 8, 11
89. Emanuele Palazzolo, Jens Behley, Philipp Lottes, Philippe Giguere, and Cyrill Stachniss. Refusion: 3d reconstruction in dynamic environments for rgb-d cameras exploiting residuals. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7855–7862. IEEE, 2019. 8, 9, 11
90. Johannes Kopf, Xuejian Rong, and Jia-Bin Huang. Robust consistent video depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1611–1621, 2021. 8, 9, 12
91. Zhoutong Zhang, Forrester Cole, Zhengqi Li, Michael Rubinstein, Noah Snavely, and William T Freeman. Structure and motion from casual videos. In *European Conference on Computer Vision*, pages 20–37. Springer, 2022. 8, 9, 12
92. René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 12179–12188, 2021. 8
93. Haoqiang Fan, Hao Su, and Leonidas J Guibas. A point set generation network for 3d object reconstruction from a single image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 605–613, 2017. 9
94. Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The international journal of robotics research*, 32(11):1231–1237, 2013. 9, 12
95. Jamie Shotton, Ben Glocker, Christopher Zach, Shahram Izadi, Antonio Criminisi, and Andrew Fitzgibbon. Scene coordinate regression forests for camera relocalization in rgb-d images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2930–2937, 2013. 10, 12
96. Dejan Azinović, Ricardo Martin-Brualla, Dan B Goldman, Matthias Nießner, and Justus Thies. Neural rgb-d surface reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6290–6301, 2022. 10, 12
97. Thomas Schops, Johannes L Schonberger, Silvano Galliani, Torsten Sattler, Konrad Schindler, Marc Pollefeys, and Andreas Geiger. A multi-view stereo benchmark with high-resolution images and multi-camera videos. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3260–3269, 2017. 10, 12
98. Federico Perazzi, Jordi Pont-Tuset, Brian McWilliams, Luc Van Gool, Markus Gross, and Alexander Sorkine-Hornung. A benchmark dataset and evaluation

methodology for video object segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 724–732, 2016. 11, 12

99. Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbeláez, Alex Sorkine-Hornung, and Luc Van Gool. The 2017 davis challenge on video object segmentation. *arXiv preprint arXiv:1704.00675*, 2017. 11, 12
100. Fuxin Li, Taeyoung Kim, Ahmad Humayun, David Tsai, and James M Rehg. Video segmentation by tracking many figure-ground segments. In *Proceedings of the IEEE international conference on computer vision*, pages 2192–2199, 2013. 11, 12
101. Junyu Xie, Weidi Xie, and Andrew Zisserman. Segmenting moving objects via an object-centric layered representation. *Advances in neural information processing systems*, 35:28023–28036, 2022. 11, 12
102. Junyu Xie, Weidi Xie, and Andrew Zisserman. Appearance-based refinement for object-centric motion segmentation. In *European Conference on Computer Vision*, pages 238–256. Springer, 2024. 11, 12
103. Yang Zheng, Adam W Harley, Bokui Shen, Gordon Wetzstein, and Leonidas J Guibas. Pointodyssey: A large-scale synthetic dataset for long-term point tracking. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19855–19865, 2023. 11, 13
104. Xiaqing Pan, Nicholas Charron, Yongqian Yang, Scott Peters, Thomas Whelan, Chen Kong, Omkar Parkhi, Richard Newcombe, and Yuheng Carl Ren. Aria digital twin: A new benchmark dataset for egocentric 3d machine perception. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 20133–20143, 2023. 11, 13
105. Hanbyul Joo, Hao Liu, Lei Tan, Lin Gui, Bart Nabbe, Iain Matthews, Takeo Kanade, Shohei Nobuhara, and Yaser Sheikh. Panoptic studio: A massively multiview system for social motion capture. In *Proceedings of the IEEE international conference on computer vision*, pages 3334–3342, 2015. 11, 13
106. Shinji Umeyama. Least-squares estimation of transformation parameters between two point patterns. *IEEE Transactions on pattern analysis and machine intelligence*, 13(4):376–380, 2002. 12
107. Hanyu Zhou, Haonan Wang, Haoyue Liu, Yuxing Duan, Yi Chang, and Luxin Yan. Bridge frame and event: Common spatiotemporal fusion for high-dynamic scene optical flow. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 27904–27913, 2025. 15
108. Haonan Wang, Hanyu Zhou, Haoyue Liu, and Luxin Yan. Injecting frame-event complementary fusion into diffusion for optical flow in challenging scenes. *arXiv preprint arXiv:2510.10577*, 2025. 15