OpenVO: Open-World Visual Odometry with Temporal Dynamics Awareness

Supplementary Material

In this supplementary material, we first provide a more detailed implementation of our Differentiable 2D-Guided 3D Flow in Sec. 1. We then present additional training and evaluation analysis in Sec. 2. Furthermore, we outline potential applications enabled by OpenVO in Sec. 3. Finally, we include additional qualitative results in Sec. 4, demonstrating the robustness and superior trajectory reconstruction capabilities of OpenVO across diverse scenarios.

1. Differentiable 2D-Guided 3D Flow

Given two consecutive frames with metric depth and a dense optical flow field, our differentiable 2D-guided 3D flow layer computes a per-pixel 3D flow in the coordinate system of the first camera. Let $\mathbf{D}_1, \mathbf{D}_2 \in \mathbb{R}^{H\times W}$ denote the metric depth maps at times t_0 and t_1, respectively, expressed in meters. The camera intrinsics are parameterized as:

$$K = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \tag{1}$$

and the optical flow from $t_0 \to t_1$ is given by:

$$\mathbf{F}(u,v) = \begin{bmatrix} \Delta u(u,v) \\ \Delta v(u,v) \end{bmatrix}, \qquad \mathbf{F} \in \mathbb{R}^{H\times W\times 2} \tag{2}$$

where (u, v) indexes pixel coordinates. We back-project every pixel (u, v) by constructing the homogeneous image coordinate $\mathbf{p}$ into 3D using the depth $\mathbf{D}_1(u, v)$

$$\mathbf{P}_1(u,v) = \mathbf{D}_1(u,v)\, K^{-1}\mathbf{p} = \begin{bmatrix} X_1(u,v) \\ Y_1(u,v) \\ Z_1(u,v) \end{bmatrix} \in \mathbb{R}^3 \tag{3}$$

In coordinates, this is equivalent to:

$$\begin{bmatrix} X_1(u,v) \\ Y_1(u,v) \\ Z_1(u,v) \end{bmatrix} = \begin{bmatrix} \dfrac{(u-c_x)\,\mathbf{D}_1(u,v)}{f_x} \\ \dfrac{(v-c_y)\,\mathbf{D}_1(u,v)}{f_y} \\ \mathbf{D}_1(u,v) \end{bmatrix} \tag{4}$$

To establish dense correspondences between the two consecutive frames, we warp each pixel (u, v) from time t_0 to its estimated sub-pixel location at time t_1 using the optical flow field. The warped pixel coordinates are computed as:

$$u' = u + \Delta u(u,v), \qquad v' = v + \Delta v(u,v) \tag{5}$$

This operation allows the model to reason about motion at a finer level than integer pixel shifts, which is crucial for handling small object motions, rolling-shutter distortions, and variations in frame rates. Since the predicted coordinates (u', v') are generally non-integer and may lie between pixel centers, we perform differentiable sampling of the target depth map $\mathbf{D}_2$. We define the final sampling grid:

$$\mathbf{g}(u,v) = \begin{bmatrix} u'(u,v) \\ v'(u,v) \end{bmatrix} \tag{6}$$

We declare an operation $\sigma(\cdot)$ which takes metric depth $\mathbf{D}_2$ and sampling grid $\mathbf{g}$ as input and outputs sampled metric depth $\tilde{\mathbf{D}}_2(u, v)$ as:

$$\mathbf{D}_2(u',v') = \sum_{i\in\{0,1\}} \sum_{j\in\{0,1\}} w_{ij}\, \mathbf{D}_2(u_i, v_j) \tag{7}$$

where pixel neighbors are:

$$u_i = \lfloor u' \rfloor + i, \quad v_j = \lfloor v' \rfloor + j, \quad i,j \in \{0,1\} \tag{8}$$

and bilinear interpolation weights are:

$$\gamma = u' - u_0, \qquad \psi = v' - v_0, \tag{9}$$

$$w = \begin{bmatrix} (1-\gamma)(1-\psi) & (1-\gamma)\psi \\ \gamma(1-\psi) & \gamma\psi \end{bmatrix} \tag{10}$$

We obtain $\tilde{\mathbf{D}}_2 = \sigma(\mathbf{D}_2, \mathbf{g})$, then following Eq. (3), we back-project $\tilde{\mathbf{D}}_2$ into 3D at time t_1 to obtain $\mathbf{P}_2(u, v)$. Finally, we compute a dense 3D flow for every pixel:

$$\mathbf{S}(u,v) = (\mathbf{P}_2 - \mathbf{P}_1) \tag{11}$$

The entire pipeline used to obtain the warped depth $\tilde{\mathbf{D}}_2(u, v)$ is fully differentiable with respect to the depth map, the warped coordinates (u', v'), and the predicted optical flow $(\Delta u, \Delta v)$. Since the warped position satisfies $u' = u + \Delta u$ and $v' = v + \Delta v$, gradients propagate through the sampling process via the chain rule. For example:

$$\frac{\partial \tilde{\mathbf{D}}_2}{\partial \Delta u} = \frac{\partial \tilde{\mathbf{D}}_2}{\partial u'} \cdot \frac{\partial u'}{\partial \Delta u} = \frac{\partial \tilde{\mathbf{D}}_2}{\partial u'} \tag{12}$$

This smooth interpolation ensures stable gradient flow across sub-pixel displacements, enabling reliable end-to-end optimization for Time-Aware Flow Encoder block.

Method	# params (M)	Runtime (s)
WildCamera [11]	270.447	41.906
Metric3Dv2 [4]	411.941	180.752

Table 1. **Expected latency of foundation priors.**

Method	# params (M)	Runtime (s)
Flow Estimator	20.655	3.089
Time Condition Layers	0.003673	0.0101
Diff. 2D-Guided 3D Flow	-	0.0204
Self-Attn	174.695	0.0339
Context Encoder	56.702	0.0129
Egomotion Decoder	29.988	0.4368

Table 2. **Expected latency of our OpenVO.** Red rows indicate Time-Aware Flow Encoder block. Green row indicates Geometry-Aware Context Encoder block. Purple row indicates Egomotion decoder block.

Setting	KITTI 00–10 (10Hz)				nuScenes (12Hz)			
	t_{err}	r_{err}	ATE	s_{err}	t_{err}	r_{err}	ATE	s_{err}
Ours–Oracle	7.22	2.75	90.85	0.05	6.41	3.44	4.24	0.09

Table 3. **Oracle testing.** We equip OpenVO with ground-truth intrinsic parameters to isolate errors from calibration, revealing the upper-bound performance under perfect camera parameters estimation.

To ensure that only geometrically valid correspondences contribute to the loss, we construct a binary validity mask

$$\mathbf{m}(u,v) = \mathbf{1}\big[\,0 \le u' \le W-1,\;\; 0 \le v' \le H-1,\\ \mathbf{D}_1(u,v) > 0,\;\; \tilde{\mathbf{D}}_2(u,v) > 0\big]. \tag{13}$$

where $\mathbf{1}$ is an indicator function and apply it element-wise to the 3D flow:

$$\tilde{\mathbf{S}}(u,v) = \mathbf{m}(u,v)\mathbf{S}(u,v) \tag{14}$$

The mask $\mathbf{m}$ is a non-learned, piecewise-constant tensor computed from geometric constraints; during backpropagation we do not differentiate through the indicator function, so the operation remains differentiable w.r.t. $\mathbf{S}$, and gradients are simply zeroed out at invalid pixels.

2. Analysis

Motivation: Δt is a conditioning signal indicating the temporal stride of the input pair, serving as a proxy for providing temporal context so the network can produce stable pose increments across different sampling intervals as shown in Tab. 4 (main paper). The purpose of conditioning is not to deterministically scale the predicted pose or to "force" a particular motion magnitude, but to *adapt to a wide range of motion magnitudes* – a behavior that prior methods cannot achieve. Without TCL, the model must self-discover motion patterns without any explicit temporal modeling, leading to suboptimal learning as the single flow encoder backbone is overloaded with multiple conflicting objectives (differentiating small, medium, and large motion magnitudes). Our TCL with the proposed P.E. provides the necessary temporal bridge, stabilizing and improving learning. Unlike prior work, our Context Encoder jointly encodes image–depth inputs and camera parameters (L.361), enabling the model to be *camera-aware* and generalize effectively to **any unseen** camera settings during testing.

Time-Conditioning Layers	P.E.	t_{err}	r_{err}	ATE
Cross-Attn (PE(Δt) in Eq.1)	✓	12.74	4.22	155.30
Cross-Attn (single scalar Δt)	✗	14.74	4.78	180.16

Table 4. **Time-Conditioning Layers design.** Using Cross-Attention layers as alternatives to the design of Time-Conditioning Layers and the use of our proposed positional encoding (P.E.).

Ablation	P.E.	Flow	Geometry	t_{err}	r_{err}	ATE
OpenVO	✓	✓	✗	17.12	4.42	158.67
OpenVO	✓	✗	✓	-	-	-
OpenVO	✗	✓	✓	9.59	3.54	117.50

Table 5. **Time-Conditioning Layers design.** Using Cross-Attention layers as alternatives to the design of Time-Conditioning Layers and the use of our proposed positional encoding (P.E.).

Methods	t_{err}	r_{err}	ATE
PnP+Metric3Dv2+MaskFlow	54.38	2.81	302.74
DroidCalib [3]: no Sim(3) (fair)	69.16	0.625	382.65
DroidCalib [3]: w/ Sim(3) (unfair)	11.01	0.625	73.18
ViPE [5]: no Sim(3) (fair)	13.29	0.721	137.15
ViPE [5]: w/ Sim(3) (unfair)	10.15	0.721	72.73

Table 6. **Geometry-based and optimization-based methods.**

Latency: We report the number of parameters and the total running time for each component of our OpenVO. For the internal calibrator WildCamera [11] and metric depth estimator Metric3Dv2 [4], we report the expected runtime for a *single process on 320 image frames* of Argoverse 2 [9] in Tab. 1. For OpenVO, we present the number of parameters and runtime for each components for a single forward for batch of 16 pairs of images in Tab. 2. Because the flow encoder performs convolution operations over full-resolution image grids, while the self-attention blocks operate on patch-level representations, the two modules exhibit distinct runtime behaviors.

Oracle Test: Given that accurate camera intrinsics are pivotal to our pipeline, we further examine an oracle scenario where the method is supplied with ground-truth in-

trinsic parameters. We conduct these oracle evaluations on nuScenes [1] and KITTI [2] to quantify the potential performance gains obtainable under perfect calibration in Tab. 3.

Ablation on Time-Conditioning Layers: We further replace Cross-Attention as the Time-Conditioning Layer (TCL) and ablate our positional encoding in Tab. 4. When disregard our proposed P.E., we use a single scalar Δt as a temporal condition. The results validate the effectiveness of our design, potentially motivating the 3D/4D reconstruction communities to leverage temporal dynamics for computer vision problems.

Ablation on the proposed components: We provide an ablation on the feature stream of OpenVO in Tab. 5. Without the flow branch, we cannot estimate the geometry difference between two consecutive frames. OpenVO without explicit geometry (depth) modeling and camera information suffers from generalization with t_{err} of 17.12.

Comparison with geometric-based and optimization-based methods: Traditional closed-world monocular VO/SLAM approaches with known intrinsics, backend optimization & loop closure, are evaluated with known GT poses alignment. Our OpenVO is evaluated under a strictly harder setting: unknown intrinsics, no GT alignment, no loop closure, and ***zero-shot cross-dataset generalization***. We showcase some state-of-the-art (2026-March) and traditional geometry-based methods on visual odometry in Tab. 6. Our OpenVO still achieves state-of-the-art results on reconstructing real-world scale egomotion without the need of alignment.

3. Global High-Definition (HD) Semantic Maps Reconstruction

HD maps play a critical role in autonomous driving and 3D scene understanding, providing precise lane geometry, road topology, traffic elements, and other structural cues that are necessary for enhanced situational awareness and safer control and navigation [7, 8, 10]. Such maps enable downstream tasks—including planning, motion prediction, navigation, and safety validation—to operate with spatial awareness and robust scene priors. However, HD maps are expensive to produce, typically requiring a combination of LiDAR sensors, human annotation, and specialized mapping vehicles. This motivates reconstructing HD maps directly from onboard sensors like cameras as a more scalable alternative. Yet this approach faces substantial technical challenges, including heavy dependency on ego-motion, occlusions, sensor noise, and complex 3D scene structure, which together can lead to compounding errors in the reconstructed maps over the horizon, especially under uncalibrated monocular observation under varying frame rates. Integrating OpenVO into the mapping pipeline helps mitigate these issues by providing accurate frame-to-frame poses that register local map predictions into a coherent global representation in world coordinates. This combination compensates for camera motion and enables consistent multi-view fusion, allowing scalable, camera-only reconstruction of high-quality HD semantic maps.

Metrics	**Full (6) cams + Lidar**	**Front cam only**
mAP @ 0.5 (↑)	0.187	0.102
mAP @ 1.0 (↑)	0.479	0.318
mAP @ 1.5 (↑)	0.646	0.472
Overall mAP (↑)	**0.4375**	**0.297**
mAP ped @ 1.5 (↑)	0.644	0.469
mAP div @ 1.5 (↑)	0.642	0.509
mAP cont @ 1.5 (↑)	0.653	0.437
mAP @ 1.5 (↑)	**0.646**	**0.472**

Table 7. Comparison between the full-sensor configuration and our monocular front-camera–only configuration for HDMap reconstruction. Green cells highlight our configuration. *ped* denotes pedestrian crossings, *div* denotes dividers, and *cont* denotes contours. Performance is reported in terms of mean Average Precision (mAP).

Following VectorMapNet [7], we reconstruct vectorized local HD maps directly from front-view imagery in Fig. 1. We disregard the LiDAR branch and train the network using only a monocular front-view stream. Ground-truth camera intrinsics and poses are used during training, whereas at inference we rely on the VO estimated by OpenVO and the camera parameters predicted by WildCamera [11]. For reference purpose, we present our quantitative results on local HDMap reconstruction in Tab. 7 and qualitative result in Fig. 3. Since the network predicts local map fragments independently, we apply post-processing heuristics to spatially align and merge these fragments into a coherent global map. Combining the modified architecture with our proposed OpenVO, we can obtain the final global map as illustrated in Fig. 2

4. Qualitative Results

Stereo VO: We showcase the performance of our OpenVO on the challenging stereo benchmark Argoverse 2 [9] in Fig. 4. In Argoverse 2, the stereo image pairs produce loose and noisy metric depth due to limited baseline, challenging lighting, and frequent low-texture or distant regions. This makes the recovered depth unstable across frames, which in turn causes conventional VO systems to become highly sensitive to depth noise and scale fluctuations. As a result, even small stereo-depth errors can propagate and lead to inconsistent trajectory estimates. In con-

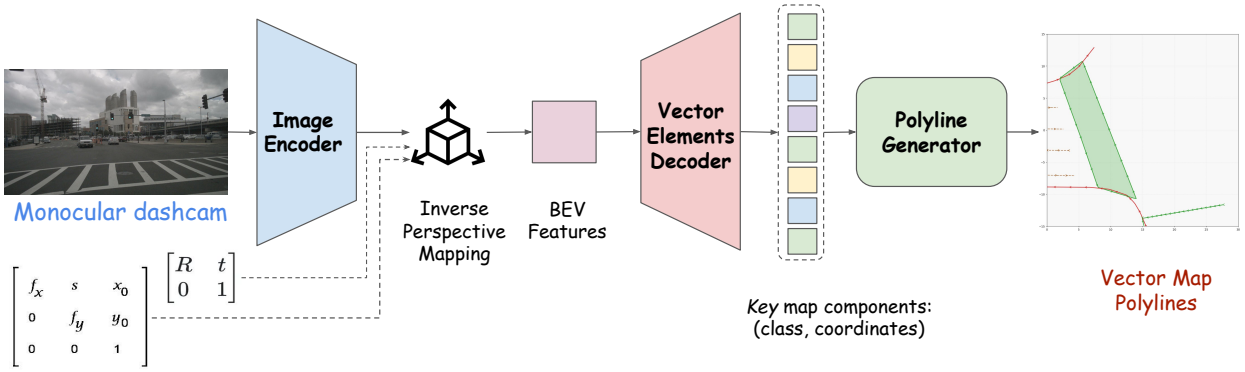

Figure 1. **Modified VectorMapNet [7].** A front-view input image is first processed by an image encoder to extract semantic and geometric features. These features are then lifted into a bird's-eye-view (BEV) representation using inverse perspective mapping, which leverages the camera's intrinsic and extrinsic parameters from **OpenVO** to geometrically project image features onto the ground plane. The resulting BEV feature map is fed into the Vector Map Decoder, which predicts structured map elements in an intermediate representation consisting of key components such as polyline classes, control points, and geometric attributes. Finally, a polyline generator converts these decoded components into continuous vectorized map elements, such as lane boundaries, road dividers, and crosswalks – yielding a high-resolution, topologically meaningful HD map suitable for downstream driving tasks.

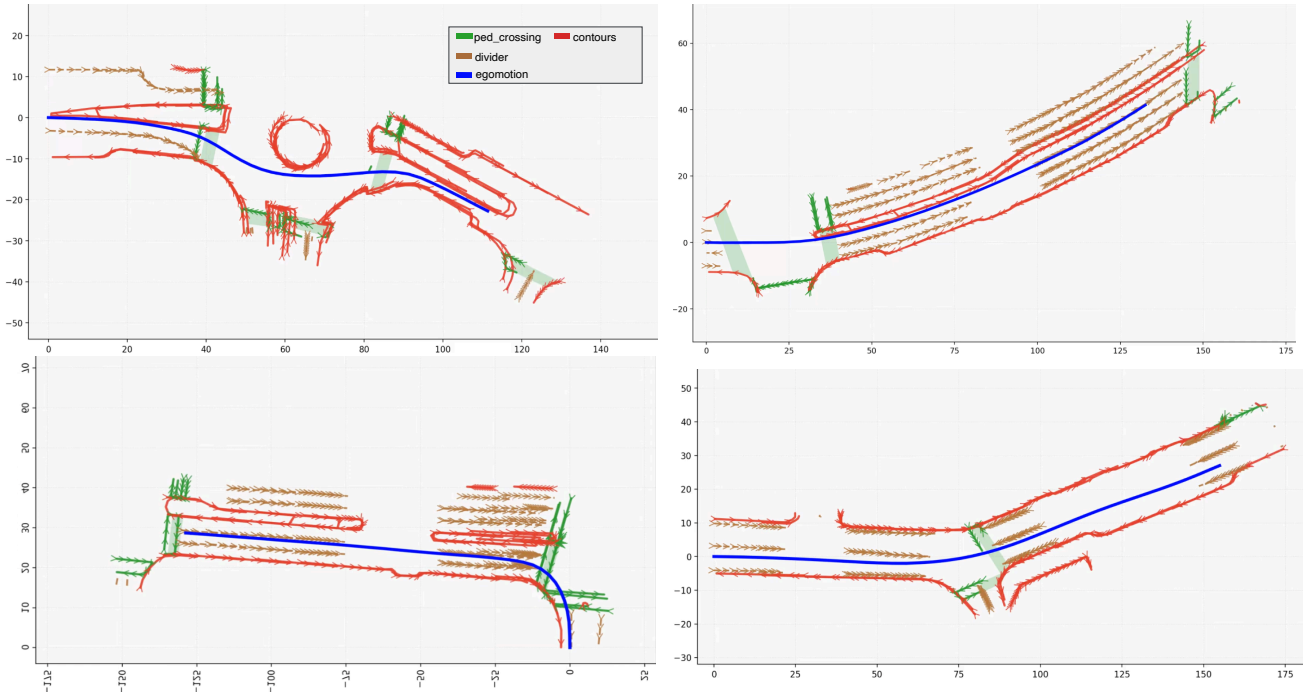

Figure 2. Qualitative results of Global HDMap reconstruction results produced by **OpenVO** + modified monocular VectorMapNet [7]. Local mapping outputs are gradually fused through OpenVO's ego-to-world pose estimates, producing a coherent global HD-map reconstruction of the full scenario. We would like to refer to **our supplementary videos** for further details of the OpenVO-enabled monocular-based global map reconstruction.

trast, our OpenVO approach remains robust under these conditions and delivers stable, consistent results despite the imperfections of the stereo-derived metric depth.

Long-Range VO: We show the performance of our OpenVO on the challenging long-range KITTI [2] benchmark in Fig. 5. The KITTI odometry dataset contains many long-range highway and suburban scenes where most structures lie far from the camera. In such settings, monocular cues become weak: distant objects provide very small pixel motion, depth becomes highly ambiguous, and small errors in these regions can accumulate into noticeable scale drift. As a result, conventional monocular VO systems tend to be

Figure 3. Qualitative HD Map reconstruction results produced by modified monocular VectorMapNet [7]. The leftmost column displays the input RGB frames; the second column shows the ground-truth HDMaps; the third column presents the results from our modified VectorMapNet using only a single front-camera image; and the last column shows the outputs under the six-camera + LiDAR configuration. We highlight the differences in each example using dashed pink regions.

sensitive on KITTI, especially over long trajectories where scale inconsistency quickly compounds. Despite these challenges, OpenVO produces stable and consistent results on KITTI, showing that our time-aware and geometry-aware design remains reliable even in long-range, low-parallax environments.

Real-World VO: We further assess the generalizability of OpenVO using videos from real-world environments, as illustrated in Fig. 6. Real-world driving scenes are heavily populated with vehicles, pedestrians, and cyclists, and are often dominated by occlusions and rapid appearance changes. Such dynamic factors degrade monocular geometric cues and introduce unstable depth signals, making conventional VO pipelines prone to drift and inconsistency. Despite such challenges, OpenVO produces stable and coherent trajectories, highlighting its robustness to noise, clutter, and complex dynamic activity commonly encountered in real-world driving.

Application beyond VO: OpenVO-enabled Monocular Global Map Reconstruction: Reconstructing real-world long-tail scenarios is crucial for safety-critical domains such as autonomous driving, where realistic simulations of rare events provide insights into failure modes, vehicle dynamics, and hazardous scene geometry. However, collecting such data directly is extremely challenging due to economic constraints, safety risks, liability concerns, and legal prohibitions. In contrast, dashcam videos offer an abundant source of real-world long-tail footage, but are typically monocular, uncalibrated, and can be captured at different frame rates, which together pose great challenges for accurate 3D reconstruction tasks like mapping. Prior online mapping approaches [6, 7, 10] address local mapping using calibrated multi-camera or LiDAR setups with fixed observation rates, which effectively enhance online situational awareness but cannot reconstruct full HD map trajectories or scene evolution required for simulating rare events. In Fig. 3 and Fig. 2 and and our corresponding supplementary video, we show that **OpenVO can bridge the global mapping gap** by generating accurate world-consistent poses and fusing them with local mapping to reconstruct the full observable scene, yielding coherent geometric and dynamic information about rare events. This facilitates scalable, monocular-based reconstruction of complex real-world scenarios from dashcam footage, offering deeper insights and enabling more comprehensive training and validation of autonomous driving algorithms in long-tail settings.

References

[1] 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 (CVPR)*, pages 11621–11631, 2020. 3

[2] 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. 3, 4, 8

[3] Annika Hagemann, Moritz Knorr, and Christoph Stiller. Deep geometry-aware camera self-calibration from video. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3438–3448, 2023. 2

[4] Mu Hu, Wei Yin, Chi Zhang, Zhipeng Cai, Xiaoxiao Long, Hao Chen, Kaixuan Wang, Gang Yu, Chunhua Shen, and Shaojie Shen. Metric3d v2: A versatile monocular geometric foundation model for zero-shot metric depth and surface normal estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024. 2

[5] Jiahui Huang, Qunjie Zhou, Hesam Rabeti, Aleksandr Korovko, Huan Ling, Xuanchi Ren, Tianchang Shen, Jun Gao, Dmitry Slepichev, Chen-Hsuan Lin, Jiawei Ren, Kevin Xie, Joydeep Biswas, Laura Leal-Taixe, and Sanja Fidler. Vipe: Video pose engine for 3d geometric perception. In *NVIDIA Research Whitepapers arXiv:2508.10934*, 2025. 2

[6] Qi Li, Yue Wang, Yilun Wang, and Hang Zhao. Hdmapnet: An online hd map construction and evaluation framework. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 4628–4634. IEEE, 2022. 6

[7] Yicheng Liu, Tianyuan Yuan, Yue Wang, Yilun Wang, and Hang Zhao. Vectormapnet: End-to-end vectorized hd map learning. In *International Conference on Machine Learning*, pages 22352–22369. PMLR, 2023. 3, 4, 5, 6

[8] Anqi Shi, Yuze Cai, Xiangyu Chen, Jian Pu, Zeyu Fu, and Hong Lu. Globalmapnet: An online framework for vectorized global hd map construction. *arXiv preprint arXiv:2409.10063*, 2024. 3

[9] Benjamin Wilson, William Qi, Tanmay Agarwal, John Lambert, Jagjeet Singh, Siddhesh Khandelwal, Bowen Pan, Ratnesh Kumar, Andrew Hartnett, Jhony Kaesemodel Pontes, Deva Ramanan, Peter Carr, and James Hays. Argoverse 2: Next generation datasets for self-driving perception and forecasting. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (NeurIPS D&B)*, 2021. 2, 3, 7

[10] Tianyuan Yuan, Yicheng Liu, Yue Wang, Yilun Wang, and Hang Zhao. Streammapnet: Streaming mapping network for vectorized online hd map construction. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 7356–7365, 2024. 3, 6

[11] Shengjie Zhu, Abhinav Kumar, Masa Hu, and Xiaoming Liu. Tame a wild camera: In-the-wild monocular camera calibration. *Advances in Neural Information Processing Systems*, 36:45137–45149, 2023. 2, 3

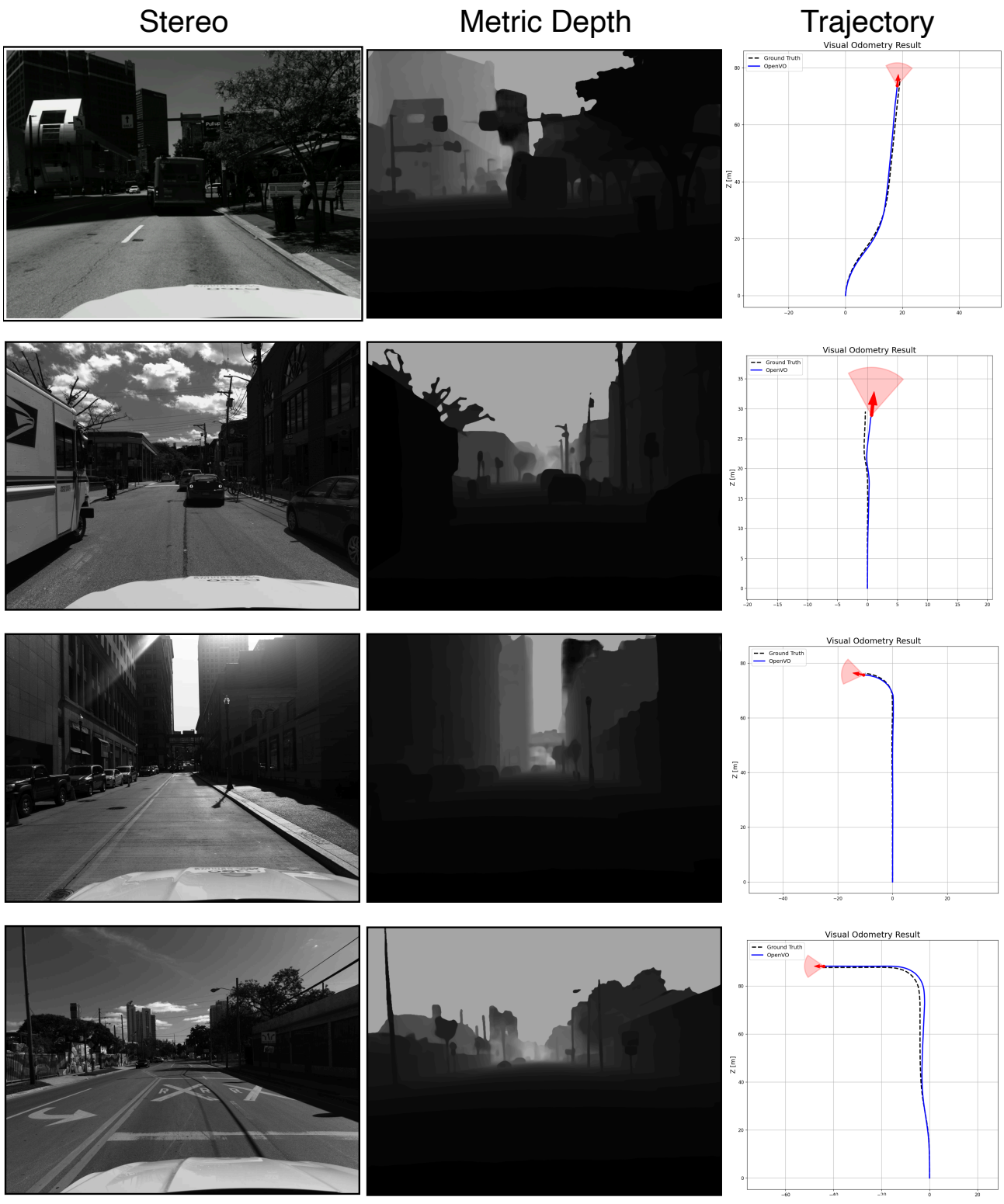

Figure 4. **Qualitative of Stereo benchmark on Argoverse 2 [9].** Each row shows one example, including the input stereo image and the reference metric depth. The stereo images in Argoverse 2 often provides low-quality or weakly constrained metric depth due to limited disparity in long-range regions and visually challenging street scenes. This degradation leads to information loss and introduces uncertainty into downstream VO estimation.

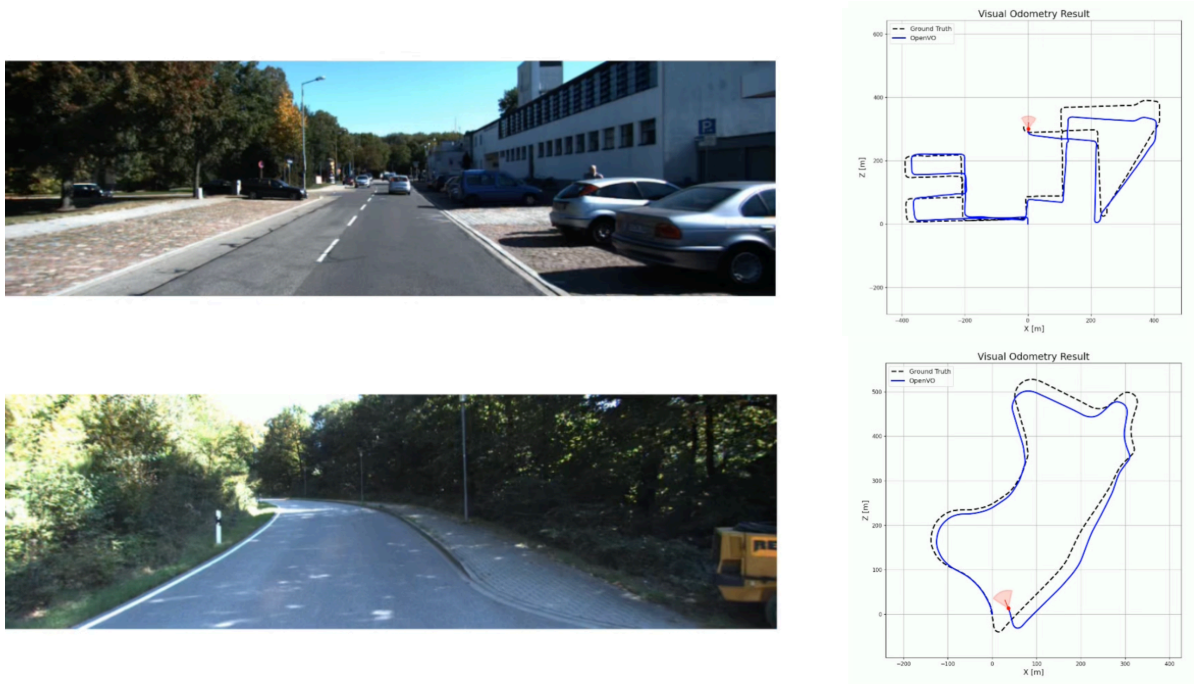

Figure 5. **Qualitative results on KITTI [2] benchmark.** Each row presents one example. The KITTI camera provides a wider field of view than most datasets, allowing it to capture a richer set of dynamic objects while still preserving its long-range odometry characteristics.

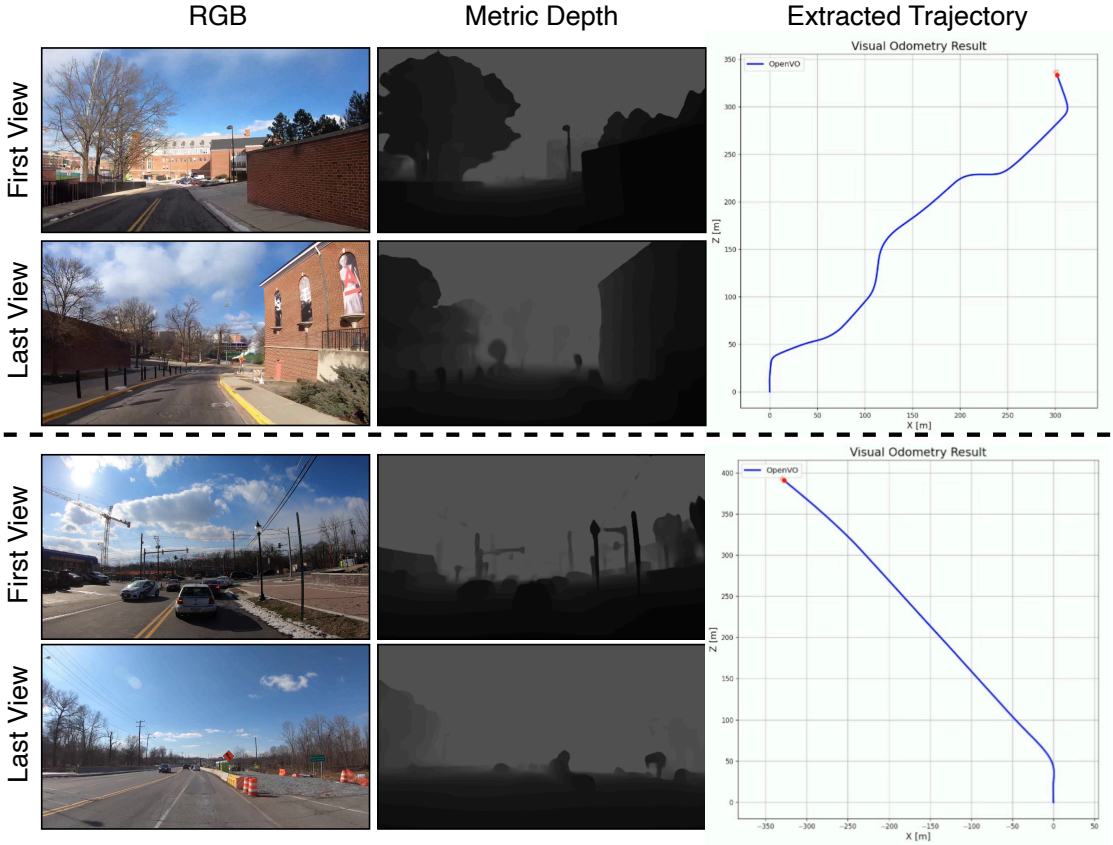

Figure 6. **Qualitative results on real-world captured videos.** We present two examples, each accompanied by the corresponding RGB frames and reference metric-depth images. Real-world videos commonly exhibit numerous environmental artifacts—such as noise, clutter, and dynamic elements—which pose significant challenges for generalizability and real-world performance assessment.