2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

CDV-SLAM: Compact Deep Visual SLAM with Unified Semantic and Geometric Perception

Ya Fan and Rongling Lang*

***Abstract*— Robust monocular visual Simultaneous Localization and Mapping (SLAM) serves as a cornerstone for various applications. However, its performance frequently suffers degradation in challenging scenarios including fast motion, dynamic objects, and scale ambiguity. This paper proposes CDV-SLAM, a compact deep visual SLAM framework that unifies geometric and semantic perception through a shared visual foundation model. A tight semantic-geometric fusion network is devised to predict optical flow in fast motion. Semantic features are efficiently reused to obtain segmentation and monocular depth for dynamic objects exclusion and scale acquisition. To further address scale drift, we introduce local scale correction in bundle adjustment. Experimental results demonstrate a 42% decrease in average Absolute Trajectory Error (ATE) on the KITTI dataset over the state-of-the-art. Furthermore, our flow-only visual odometry surpasses geometric-only methods on the TartanAir and EuRoC datasets, with a marginal speed reduction of 6%. Our code is publicly available at** https://github.com/FrankYard/CDV-SLAM**.**

I. INTRODUCTION

Simultaneous Localization and Mapping (SLAM) is a cornerstone technology for various applications, including mobile robotics, autonomous driving, and augmented reality. However, traditional monocular visual SLAM systems [1]–[3] often lack robustness when facing challenging conditions such as fast motion, dynamic objects, and scale ambiguity. With the advancements in deep learning, recent works have sought to overcome some of these challenges by leveraging geometric or semantic features from neural networks. For instance, end-to-end trained geometric features [4]–[7] have demonstrated high accuracy in scenarios with fast motion. Concurrently, methods based on semantic features can enhance stability in the presence of dynamic objects through class information [8], [9] and can also perceive scale information to mitigate scale drift [10].

Nevertheless, each of these methods has difficulty handling all the aforementioned challenges. Efficiently integrating these approaches into a single, unified system to maximize monocular stability remains a significant challenge. The primary obstacle to such a unified system is the difficulty of perceiving semantic and geometric information cohesively within a neural network. A deep SLAM system that unifies semantic and geometric perception should exhibit the following characteristics: 1) Compactness: The semantic and geometric networks should be tightly integrated, resulting in an overall complexity comparable to that of a purely geometric SLAM network. 2) Flexibility: The system should allow for the selective activation of different perception modalities to enhance SLAM, such as segmentation masks or scale. 3) Robustness: The system maintains high trajectory accuracy in challenging scenarios. Existing deep SLAM methods often rely on specialized networks, making it difficult to satisfy these conditions simultaneously.

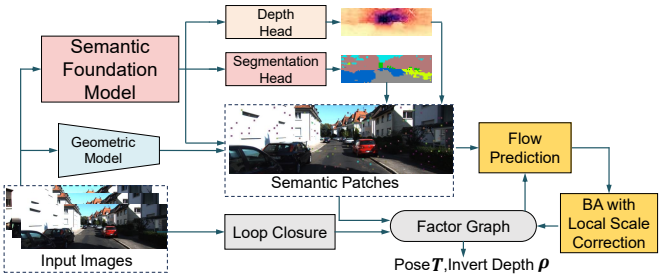

Fig. 1. Overview of our full SLAM system. The system acquires semantic segmentation, depth, and patch features from a unified foundation model, making it compact and efficient.

In this paper, we propose a compact deep monocular SLAM system with unified semantic and geometric perception, as illustrated in Fig. 1. The compactness of our system is achieved in two ways: First, the system utilizes a visual foundation model [11] to provide shared semantic features for optical flow prediction, semantic segmentation, and monocular depth estimation. Second, we implement an efficient fusion of semantic and compact geometric features for iterative flow refinement in fast motion. This design introduces a marginal speed reduction of 6% compared to the geometry-only baseline [7]. The system can flexibly activate single-layer heads for semantic segmentation and depth prediction. Based on the class information from the segmentation head, dynamic objects can be filtered out during the patch extraction stage. The depth head infers scale-aware monocular depth from semantic features. To suppress scale drift, we design a local scale correction method within bundle adjustment. This method leverages scale information from the depth head while preventing inaccurate depth values from disrupting the factor graph optimization. Through end-to-end training and the proposed designs, our SLAM system achieves state-of-the-art (SOTA) average absolute trajectory error (ATE) on KITTI [12] dataset, and our visual odometry (VO) improves accuracy on challenging sequences. Results on TartanAir [13] and EuRoC [14] datasets further demonstrate the robustness of our system in challenging conditions.

The main contributions of this paper are as follows:

*Corresponding author.
School of Electronic Information Engineering, Beihang University, Beijing 100191, China. fanya1502@buaa.edu.cn, ronglinglang@buaa.edu.cn

- We propose CDV-SLAM, a novel and compact deep network for monocular SLAM that unifies semantic and geometric perception by effectively reusing semantic features from a foundation model and fusing semantic-geometric features.
- We introduce a local scale correction method in bundle adjustment, which effectively rectifies the scale drift of monocular SLAM using a single-layer depth prediction head.
- Our method demonstrates high robustness across the KITTI, TartanAir, and EuRoC datasets. The full system achieves state-of-the-art performance on KITTI, while the flow-only configuration demonstrates performance comparable to or exceeding that of the baseline geometric systems [6], [7] on TartanAir and EuRoC. The efficacy of our design choices is further demonstrated on various sequences.

II. RELATED WORK

A. Learned Monocular SLAM

Early pioneering works in learning-based VO like DeepVO [15], demonstrated the feasibility of end-to-end pose estimation using recurrent neural networks. Subsequent methods improved upon this by integrating additional cues. For example, D3VO [16] incorporated depth, pose, and uncertainty estimations into its sparse direct visual odometry framework. A significant trend has been the use of optical flow as an intermediate representation for motion estimation. TartanVO [4] and iSLAM [17] leveraged predicted optical flow and a camera intrinsics layer to enable robust out-of-domain generalization. Others like DiffposeNet [18], predict normal flow for pose estimation via a differentiable cheirality layer, while XVO [19] utilizes auxiliary audio task to achieve generalization across diverse driving scenarios.

More recent systems have tightly integrated end-to-end learning with classical geometric optimization. Among them, DROID-SLAM [5] and Go-SLAM [20] employ the RAFT [21] dense optical flow network within iterative GRU updates, followed by geometric optimization via bundle adjustment. DPVO [6] and DPV-SLAM [7] achieve lightweight performance through sparse image patches.

Development of vision foundation models [11] have also inspired exploration for correspondence-based localization in works like MambaVO++ [22] and DINO-VO [23]. However, these learning-based SLAM systems have largely focused on geometric aspects, without fully exploiting the potential of semantic information. In contrast, our work focuses on the tight fusion of both semantic and geometric cues within a unified framework.

B. Semantic-Augmented Monocular SLAM

Existing works have leveraged semantic information to mitigate drift in monocular SLAM in various ways. A foundational approach is the filtering of dynamic objects. Seminal works like DS-SLAM [24], SaD-SLAM [25], DynaSLAM [8] typically employ semantic detection or segmentation to identify and exclude features belonging to dynamic objects from the pipeline of classical SLAM systems. Object-based SLAM methods take this further by constructing static object-level maps [26], [27] or tracking dynamic objects [9], introducing object shape and pose constraints to suppress drift. Nevertheless, the robustness of these methods is often bottlenecked by the underlying classical SLAM system they are built upon. Distinct from these approaches, we integrate dynamic object filtering directly with a learning-based SLAM system to enhance the robustness.

Another line of work utilizes semantic information to directly infer the metric scale in monocular SLAM, thereby effectively correcting scale drift. The recent SMORE-SLAM [28] achieves scale correction through semantic-based height estimation. More generalizable monocular depth estimation [10], [29], [30] driven by semantic foundation models have demonstrated remarkable scale-aware capabilities. Our work also leverages semantic features to infer monocular depth. However, we don't rely on large and computationally expensive depth estimation models [10], [30]. Instead, in our compact system, even the low-resolution outputs from lightweight linear segmentation and depth heads suffice to achieve SOTA localization performance.

III. COMPACT DEEP VISUAL SLAM

We propose a novel system that tightly couples semantic and geometric information to achieve accurate and robust camera pose estimation. As illustrated in Fig. 1, our framework operates on a monocular image stream to extract semantic and geometric features, concurrently performing semantic patch selection (Sec. III-A). Subsequently, our optical flow network and bundle adjustment (BA) module iteratively refine the camera pose and the inverse depth of these patches. Specifically, the optical flow network employs linear attention to effectively fuse the semantic and geometric features (Sec. III-B). The BA module, in turn, leverages patch-level scale information to mitigate the inherent scale drift of monocular systems (Sec. III-C). Our visual odometry is trained end-to-end and integrated with a loop closure module to form a complete SLAM system (Sec. III-D).

A. Semantic-Geometric Feature Extraction

Our approach utilizes a large-small network architecture for feature extraction. This design enables the large foundation model to focus on high-level semantic information, while the small model specializes in low-level geometric and textural details. For image with the resolution of $H \times W$, we employ DINOv2 [11] ViT-S to produces a semantic feature map $\mathbf{F} \in \mathbb{R}^{H/14 \times W/14 \times 384}$. A shallow ConvNet of XFeat [31] is used to extract a compact geometric feature map $\mathbf{G} \in \mathbb{R}^{H/4 \times W/4 \times 24}$.

The state of each frame i consists of its pose, represented by a 4x4 transformation matrix $\mathbf{T}_i$, and the states of its associated image patches. The state of its k-th image patch is denoted as $\mathbf{P}_{i,k}$, which is a matrix comprising the homogeneous coordinates and inverse depths of its p^2 pixels:

$$\mathbf{P}_{i,k} = [\mathbf{x}, \mathbf{y}, \mathbf{1}, \rho]^T, \mathbf{x}, \mathbf{y}, \rho \in \mathbb{R}^{p^2 \times 1}, \tag{1}$$

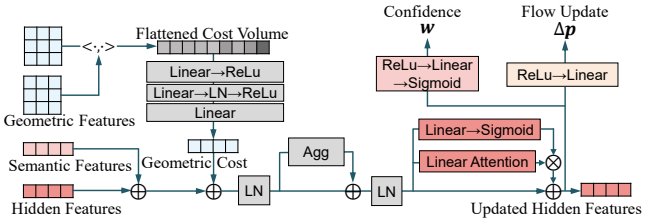

Fig. 2. Flow update network. We use gated linear attention to fuse patch features.

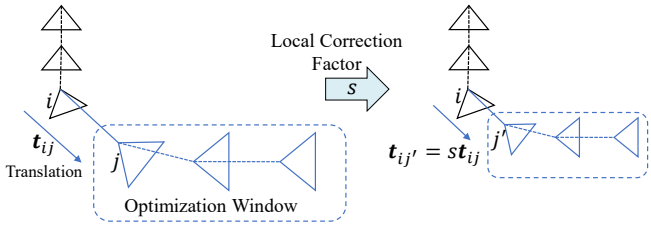

Fig. 3. The process of local scale correction.

where p is the pixel width of the patch. The rows of the state matrix represent the pixel coordinates $(\mathbf{x},\mathbf{y})$, the homogeneous coordinate $\mathbf{1}$, and the inverse depth ρ. To initialize the inverse depth, we use a lightweight depth head to predict scale-aware inverse depth ρ_0 from the semantic feature map $\mathbf{F}$.

To select patches, previous work [6] employs a random sampling strategy. However, patches may be selected from dynamic objects, which violate the static-world assumption and degrade SLAM accuracy. To mitigate this, we design a semantic patch selection mechanism. The image is first divided into multiple 14×14 cells, corresponding to the resolution of the semantic feature map. Then a lightweight segmentation head utilizes the shared $\mathbf{F}$ to identify and mask out cells containing dynamic objects, and patches are randomly sampled from the valid cell centers.

For every patch $\mathbf{P}$, we extract both semantic and geometric features. The semantic feature vector $\mathbf{f}$ is directly extracted from the semantic feature map $\mathbf{F}$ at the patch's center location, avoiding bilinear interpolation. Its geometric features, denoted as $\mathbf{G}[\mathbf{P}]$, are sampled from the geometric feature map $\mathbf{G}$ using the interpolation over the $p\times p$ pixel grid of the patch. We also cache $\mathbf{G}$ for flow prediction.

B. Flow Prediction via Semantic-Geometric Fusion

We predict optical flow through an iterative refinement process that fuses semantic and geometric features. First, a factor graph within a sliding window is constructed as in [7]. Subsequently, for each edge (i,j) in the factor graph, the flow is predicted from each patch $\mathbf{P}_{i,k}$ in the reference frame i to the target frame j, where $k=1,\ldots,K$ and K is the number of sampled patches. The semantic-geometric patch features and the geometric feature map are denoted as $\mathbf{f}_{i,k}$, $\mathbf{G}_i[\mathbf{P}_{i,k}]$, and $\mathbf{G}_j$, respectively.

In each iteration, we construct a geometric cost volume $\mathbf{C}_{i,k,j}\in\mathbb{R}^{p\times p\times q\times q}$ for each triplet $(i,k,j)\in\Omega$, where Ω is the set of indices for reference frames, patches, and target frames in the factor graph. Let Π be the camera projection function. The projection of patch center $\hat{\mathbf{P}}_{i,k}$ onto frame j is given by:

$$\hat{\mathbf{P}}'_{i,k,j}=\Pi\left[\mathbf{T}_j^{-1}\cdot\mathbf{T}_i\cdot\Pi^{-1}\left(\hat{\mathbf{P}}_{i,k}\right)\right]. \tag{2}$$

Then $\mathbf{C}_{i,k,j}$ is computed by correlating $\mathbf{G}_i[\mathbf{P}_{i,k}]$ with $\mathbf{G}_j$ in a $q\times q$ search window centered at $\hat{\mathbf{P}}'_{i,k,j}$. This component is computationally intensive. Compared to the 128-dimensional features used in the baseline method [7], our 24-dimensional compact geometric features reduce the number of multiplications by 81.25%.

To fuse geometric and semantic information, we design a flow update network, as show in Fig. 2. At iteration m, we first compute:

$$\mathbf{h}^{*m}_{i,k,j}=\mathbf{h}^{m-1}_{i,k,j}+\mathbf{f}_{i,k}+\mathrm{Enc}(\mathbf{C}_{i,k,j}) \tag{3}$$

Here, $\mathbf{h}$ denotes the hidden state vector, which is initialized to zeros. The encoder network Enc processes the geometric cost volume to align its features with the semantic feature space, enabling feature-level fusion via addition. To aggregate information across multiple edges and patches, we then perform the following computations:

$$\begin{aligned}\mathbf{h}^{**m}_{i,k,j}&=\mathbf{h}^{*m}_{i,k,j}+\mathrm{Agg}\left(\{\mathbf{h}^{*m}_{i,k,j'}|(i,k,j')\in\Omega\}\right)\\ \mathbf{h}^{m}_{i,k,j}&=\mathbf{h}^{**m}_{i,k,j}+\mathrm{Att}_{\mathrm{self}}\left(\{\mathbf{h}^{**m}_{i,k',j}|(i,k',j)\in\Omega\}\right)\end{aligned} \tag{4}$$

The Agg network aggregates features from the same patch across different target frames, using a weighted average as in [7]. The $\mathrm{Att}_{\mathrm{self}}$ network aggregates features from different patches k within the same reference-target pair (i,j), implemented with gated linear self-attention. The gates are used to stabilize training. From the updated hidden state $\mathbf{h}^m_{i,k,j}$, we finally predict an optical flow update $\Delta\mathbf{p}^m_{i,k,j}$ and a corresponding confidence $\mathbf{w}^m_{i,k,j}$ from separated decoder, as depicted in Fig. 2.

C. Semantic-driven Local Scale Correction

We perform scale correction in bundle adjustment (BA). To estimate the poses and patch states, each patch correspondence is modeled as a Gaussian distribution with mean $\boldsymbol{p}_{i,k,j}=\hat{\mathbf{P}}'_{i,k,j}+\Delta\mathbf{p}_{i,k,j}$ and covariance $\boldsymbol{\Gamma}_{i,k,j}=\mathrm{diag}\left(\mathbf{w}_{i,k,j}\right)^{-1}$. The maximum likelihood estimation over all observations leads to the following optimization problem:

$$\underset{\{\mathbf{T},\rho\}}{\arg\min}\sum_{(i,k,j)\in\Omega}\left\|\Pi\left[\mathbf{T}_j^{-1}\mathbf{T}_i\Pi^{-1}\left(\hat{\mathbf{P}}_{i,k}\right)\right]-\boldsymbol{p}_{i,k,j}\right\|^2_{\boldsymbol{\Gamma}^{-1}_{i,k,j}} \tag{5}$$

The optimization is performed within the sliding window in Sec. III-B using the Levenberg-Marquardt (LM) algorithm, alternating with the flow update module. However, this optimization objective suffers from scale ambiguity, which makes monocular SLAM prone to scale drift. We use the scale-aware depth values from the semantic features and the depth head to correct the drift. If the depth values are directly used to provide depth constraints in BA, the noise in

TABLE I

ATE[M]↓ RESULTS ON THE KITTI ODOMETRY SEQUENCE 0-10. THE TOP ONE IS IN BOLD AND THE SECOND IS UNDERLINED.

	Method	00	01	02	03	04	05	06	07	08	09	10	Avg
SLAM	ORB-SLAM3 [32]	6.77	-	30.50	**1.04**	0.93	5.54	16.61	9.70	60.69	7.89	8.65	-
	LDSO [33]	9.32	11.68	31.98	2.85	1.22	5.1	13.55	2.96	129.02	21.64	17.36	22.42
	DROID-SLAM [5]	92.1	344.6	-	2.38	1.00	118.5	62.47	21.78	161.60	-	118.70	-
	SMORE-SLAM [28]	8.22	324.72	15.93	4.40	2.15	5.54	8.90	2.53	**7.52**	**6.66**	9.10	35.97
	DPV-SLAM [7]	112.80	11.50	123.53	2.50	0.81	57.8	54.86	18.77	110.49	76.66	13.65	53.03
	DPV-SLAM++ [7]	8.30	11.86	39.64	2.50	0.78	5.74	11.6	**1.52**	110.9	76.70	13.70	25.76
	Mamba VO++ [22]	**6.19**	8.04	27.73	1.94	**0.59**	**3.05**	11.79	1.7	105.42	63.24	10.51	21.84
	CDV-SLAM++ (Ours)	8.88	**7.57**	**13.00**	3.40	0.87	10.64	**5.91**	3.58	17.39	19.65	**7.33**	**8.93**
VO	DSO [3]	126.7	165.0	138.7	4.77	1.08	49.5	113.6	27.9	120.2	74.3	16.3	76.3
	DROID-VO [5]	98.43	84.20	108.8	2.58	0.93	59.27	64.4	24.20	64.55	71.8	16.91	54.19
	DPVO [6]	113.21	12.69	123.4	2.09	0.68	58.96	54.78	19.26	115.90	75.10	13.63	53.03
	Mamba VO [22]	112.39	8.16	93.78	1.80	0.66	56.51	57.19	17.9	116.01	73.56	14.37	50.21
	DINO-VO [23]	39.8	24.8	42.6	**1.51**	**0.35**	**15.4**	**3.91**	7.39	21.6	**7.16**	**1.31**	**15.1**
	CDVO++ (Ours)	**24.88**	**6.65**	**28.02**	3.40	0.86	27.67	30.60	**3.69**	**17.39**	19.66	7.15	15.45

these predictions can interfere with the optimization and flow, thereby affecting the accuracy of SLAM. Instead, we propose local scale correction within the optimization window, which aligns the scale of the factor graph with that of the depth head predictions without affecting the BA objective.

The process of local scale correction is shown in Fig. 3. Denote the frame right before the sliding optimization window of the factor graph as frame i. Let the average inverse depth and confidence of the image patches in the factor graph be $\bar{\rho}$ and $\bar{w}$, respectively. Let the average inverse depth predicted by the depth head for these patches be $\bar{\rho}_0$. The local correction factor is then computed by

$$s = 1 + (1 - \bar{w})(\bar{\rho}_0^{-1}\bar{\rho} - 1). \tag{6}$$

Subsequently, we correct all frames within the sliding window by scaling their depths and their translations relative to frame i by the factor s. The rotations of the frames remain unchanged. Furthermore, we observed that scale drift predominantly occurs during turning motions. For efficiency, we perform the local scale correction only when the relative rotation between the latest pair of frames exceeds a predefined threshold.

D. Training and SLAM System

We train the VO component of our system in an end-to-end manner. During training, the semantic foundation model are frozen to preserve the rich semantic information and prevent overfitting. To enhance the robustness of the flow update network against dynamic objects and depth noise, we do not mask out cells during patch selection and employ random depth initialization. Furthermore, to facilitate thorough feature fusion across a wider range of inputs, we design a hybrid sparse-and-dense training procedure: at each training step, either sparse or dense training is selected with equal probability. Under the sparse training mode, image patches are randomly sampled, while in the dense training mode, patches are extracted from every cell. The training is supervised by pose and flow losses [6].

Our SLAM system extends the VO component by incorporating a loop closure module. Loop closure detection is performed using DBoW2 [34] in combination with existing keypoint detectors and matchers [35]–[37], followed by pose graph optimization on $Sim(3)$ to further correct scale drift. For flexibility, we treat the segmentation head, the depth head, and the loop closure module as optional components that can be enabled as needed. In this paper, our system without the segmentation and depth heads is referred to as CDVO. The system incorporating the loop closure module on this basis is termed CDV-SLAM. Extending these two configurations by enabling the segmentation head and the depth head leads to CDVO++ and CDV-SLAM++, respectively.

IV. EXPERIMENTS

We first introduce the setup in Sec. IV-A, followed by a performance comparison of our proposed method against state-of-the-art visual SLAM and VO methods in Sec. IV-B. We then present ablation studies in Sec. IV-C. Finally, we analyze the efficiency of our method in Sec. IV-D.

A. Implementation Details

1) Training Process: Our VO network is trained on TartanAir [13]. We use the pre-trained DINOv2 [11] ViT-S backbone and the first 7 convolutional layers of XFeat [31], keeping the parameters of the ViT backbone frozen. The p and q are set to 3 and 7, respectively. The training process starts with sparse feature training for the first 2 epochs, followed by sparse-dense hybrid training for the next 2 epochs. For sparse training, we set the batch size to 15, K=80, and M=18. For dense training, the batch size is 4, K=1530, and M=10.

2) Evaluation Setup: We adopt DPVO [6] and DPV-SLAM [7] as baseline methods, keeping the number of image patches, the sliding window length, and the loop closure settings consistent with their default settings. Operating at a resolution of 1/14 of the original image, the segmentation and depth heads use the single-layer linear networks from DINOv2 without fine-tuning. The rotation threshold for local scale correction is 0.04 radians. We evaluate the Absolute Trajectory Error (ATE) of our method on the KITTI [12], TartanAir MH [13], and EuRoC [14] datasets. For each sequence, we perform 5 repeated experiments and report the median of their results. The trajectories are aligned following the protocol in [4].

TABLE II

ATE[M]↓ RESULTS ON THE EUROC. THE TOP ONE IS IN BOLD AND THE SECOND IS UNDERLINED.

	Method	MH01	MH02	MH03	MH04	MH05	V101	V102	V103	V201	V202	V203	Avg
SLAM	ORB-SLAM3 [32]	0.016	0.027	0.028	0.138	0.072	**0.033**	0.015	0.033	0.023	0.029	-	-
	LDSO [33]	0.046	0.035	0.175	1.954	0.385	0.093	0.085	-	0.043	0.405	-	-
	GO-SLAM [20]	0.016	**0.014**	0.023	0.045	0.045	0.037	0.011	0.023	**0.016**	**0.010**	0.022	0.024
	DROID-SLAM [5]	**0.013**	**0.014**	**0.022**	**0.043**	0.043	0.037	0.012	0.02	0.017	0.013	**0.014**	**0.022**
	DPV-SLAM [7]	**0.013**	0.016	**0.022**	**0.043**	0.041	0.035	**0.008**	**0.015**	0.020	0.011	0.040	0.024
	CDV-SLAM (Ours)	**0.013**	**0.014**	**0.022**	0.044	**0.039**	0.034	0.009	**0.015**	0.017	0.011	0.046	0.024
VO	SVO [2]	0.10	0.12	0.41	0.43	0.30	0.07	0.21	-	0.11	0.11	1.08	-
	DeepV2D [38]	1.614	1.492	1.635	1.775	1.013	0.717	0.695	1.483	0.839	1.052	0.591	1.173
	TartanVO [4]	0.639	0.325	0.550	1.153	1.021	0.447	0.389	0.622	0.433	0.749	1.152	0.680
	DROID-VO [5]	0.163	0.121	0.242	0.399	0.270	0.103	0.165	0.158	0.102	0.115	**0.204**	0.186
	DPVO [6]	0.087	0.055	0.158	**0.137**	0.114	**0.050**	0.140	0.086	**0.057**	**0.049**	0.211	0.105
	MAC-VO (Stereo) [39]	0.240	0.256	0.260	0.496	0.560	0.197	**0.096**	0.198	0.145	0.270	0.425	0.286
	DINO-VO [23]	0.150	0.118	0.254	0.982	0.454	0.298	0.260	0.559	0.334	0.278	0.763	0.404
	CDVO (Ours)	**0.067**	**0.049**	**0.109**	0.144	**0.099**	**0.050**	0.150	**0.066**	0.066	0.071	0.248	**0.102**

TABLE III

ATE[M]↓ RESULTS ON THE MH SEQUENCES OF TARTANAIR DATASET.

Methods	000	001	002	003	004	005	006	007	Avg
SVO [2]	14.42	1.17	4.97	6.88	X	19.2	11.27	17.68	X
DSO [3]	9.65	0.35	7.96	3.46	X	12.58	8.42	7.50	X
DeepV2D [38]	6.15	2.12	4.54	3.89	2.71	11.55	5.53	3.76	5.03
TartanVO [4]	4.88	0.26	2.00	0.94	1.07	3.19	1.00	2.04	1.92
DROID-VO [5]	0.32	0.13	0.08	0.09	1.52	0.69	0.39	0.97	0.52
DPVO [6]	**0.21**	**0.04**	**0.04**	0.08	0.58	**0.17**	**0.11**	0.15	0.17
DINO-VO [23]	0.87	0.19	0.28	0.25	1.19	0.92	0.23	0.73	0.58
CDVO (Ours)	0.25	0.05	0.09	**0.04**	**0.28**	0.27	0.12	**0.12**	**0.15**

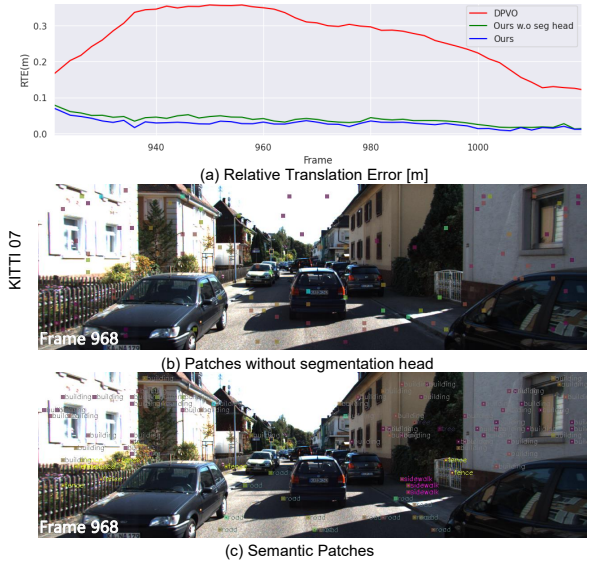

Fig. 4. Effect of semantic patch selection.

B. Monocular SLAM and VO Results

On the KITTI dataset, we enable the segmentation and depth heads to mitigate the effects of dynamic objects and scale drift. As shown in Table I, our CDV-SLAM achieves SOTA monocular SLAM performance, reducing the ATE by 59% compared to MambaVO++ and by 42% compared to DINO-VO. Sequences 01, 02, 06, and 10 are particularly challenging. Sequence 01 is a highway scenario with fast motion, while 01, 06, and 10 contain significant dynamic objects. Furthermore, all these sequences include turns that are prone to causing scale drift. Our SLAM method achieves the lowest ATE on these challenging sequences. Additionally, while the average ATE of our VO is comparable to DINO-VO, our method demonstrates superior stability, being the only one to achieve an ATE below 35 meters on all sequences.

We also evaluate the performance of our flow-only method on the TartanAir and EuRoC datasets. Because both of these datasets primarily consist of static scenes and their frequent short-term loops prevent scale drift, we do not enable the segmentation and depth heads on these datasets. Table II shows the results of our method on the EuRoC dataset, which features various lighting conditions and aggressive drone movements. Our VO outperforms DPVO and stereo MAC-VO, highlighting the superior generalization capability of our approach. Table III presents the performance of our method on the TartanAir MH sequences. This dataset is challenging for SLAM due to illumination changes, dynamic objects, low-texture areas, and significant viewpoint variations. As a method also trained on TartanAir, our VO achieves a lower average ATE than DPVO, demonstrating that our feature-level semantic-geometric fusion approach provides a better fit for these challenging scenarios.

C. Ablation Studies

We conduct ablation studies on both the network design and the SLAM components.

The ablation study on our network design is shown in Table IV. We first replace the backbone of the baseline method with the large-small network composed of the semantic foundation model and the compact geometric model. We then replace the GRU with our semantic-geometric fusion network. Finally, we add training on dense features. The results show that our semantic-geometric fusion and training design improve the Area Under Curve (AUC) on the TartanAir validation set.

We then conduct ablation study to discern the impact of each optional component on the overall SLAM performance. As shown in Table V, employing a segmentation head with semantic-guided patch selection enhanced the average

TABLE IV

ABLATION STUDY ON NETWORK DESIGNS. WE REPORT AUC↑ RESULTS.

Configuration	TartanAir Val AUC
baseline [6]	0.80
+ Large-small backbone	0.80
+ Semantic-geometric fusion	0.82
+ Training with dense features	0.83

TABLE V

ABLATION STUDY ON SLAM COMPONENTS. THE TOP ONE IS IN BOLD AND THE SECOND IS UNDERLINED.

L	S	D	01	02	08	10	Avg
✓			9.71	50.30	132.25	16.42	28.89
✓	✓		7.70	49.18	126.08	17.08	28.04
	✓	✓	**6.65**	28.02	<u>17.39</u>	**7.15**	15.45
✓		✓	10.67	<u>15.36</u>	**16.62**	7.64	<u>10.30</u>
✓	✓	✓	<u>7.57</u>	**13.00**	<u>17.39</u>	<u>7.33</u>	**8.93**

- L: loop closure. S: segmentation head. D: depth head.
- Average ATE is computed over KITTI odometry sequence 0-10.

TABLE VI

AVERAGE EXECUTION TIME [MS/FRAME]

Method	Feature Extraction	Graph Maintenance	Flow Update	BA	Total
DPVO	5.2	8.5	25.8	1.6	41.0
CDVO	11.4	8.4	22.2	1.6	43.6
CDVO++	12.3	8.3	22.2	2.0	44.8
DPV-SLAM	5.3	7.2	27.7	11.1	51.3
CDV-SLAM	11.5	7.3	23.8	10.5	53.1

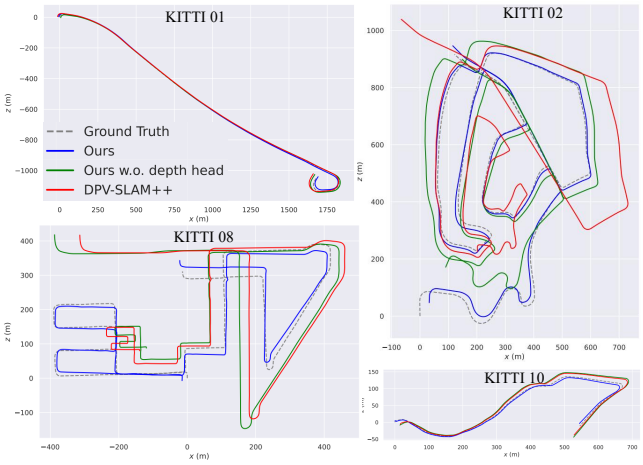

Fig. 5. Trajectories of ours full SLAM (blue), our SLAM without depth head (green), DPV-SLAM++ [7] (red), and ground truth on KITTI.

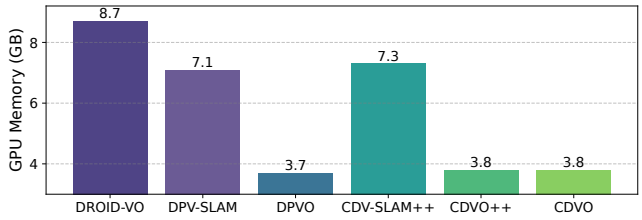

Fig. 6. GPU memory usage of our method compared with other methods.

trajectory accuracy. We find that utilizing the depth head for scale correction leads to a significant improvement in average trajectory accuracy, even in the absence of loop closure. The highest average accuracy is achieved when the loop closure, the segmentation head, and the depth head are used concurrently.

Fig. 4 illustrates the effect of semantic patch selection on the KITTI 07 sequence. The outer color of each patch denotes the flow confidences, and the inner color (if present) represents the semantic class. Without the segmentation head, patches located on dynamic objects may have high confidence, thus interfering with SLAM. In contrast, semantic patches can exclude these objects and enhance the relative translation accuracy. In Fig. 5, we provide a further qualitative comparison to demonstrate the importance of the depth head to suppress scale drift.

D. Time and Memory Analysis

We compare the runtime of our methods with DPVO and DPV-SLAM on the EuRoC dataset using a single NVIDIA RTX 3090 GPU and an AMD EPYC 7642 CPU, while maintaining an identical data pipeline. As shown in Table VI, enabling the segmentation and depth heads introduces only a small amount of additional overhead. For the flow update, which includes the correlation and the update network, our compact geometric features reduce the inference time. Using local scale correction adds a small amount of extra time to the BA process. The total processing time of our CDVO is 6% higher than that of DPVO. Our CDVO++ simultaneously obtain semantic segmentation and monocular depth estimation with an additional overhead of around 1ms, achieving a real-time performance of 22Hz on EuRoC. Our CDV-SLAM incurs about 3% additional inference time compared to DPV-SLAM. A comparison of the GPU memory usage between different configurations of our method and other methods is presented in Fig 6. It can be seen that our CDVO++ consumes only 0.1 GB more memory than DPVO, while our complete CDV-SLAM++ requires an additional 0.2 GB compared to DPV-SLAM. These results highlight the compactness of our approach, which is a benefit of our unified semantic-geometric perception framework.

V. CONCLUSIONS

In this work, we propose CDV-SLAM, a compact, flexible and robust deep visual SLAM system that effectively unifies the geometric and semantic perception. By leveraging a shared visual foundation model, we design an efficient fusion of semantic and geometric features for robust flow prediction in fast motion. Furthermore, we reuse the semantic features for dynamic object exclusion and introduce local scale correction in bundle adjustment to address monocular scale drift. Our full monocular SLAM system achieves state-of-the-art performance on KITTI. Moreover, the system's flow-only configuration surpassed baselines on TartanAir and EuRoC, confirming the framework's efficiency and flexibility with a marginal computational overhead.

REFERENCES

[1] R. Mur-Artal and J. D. Tardós, "ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras," *IEEE Transactions on Robotics*, vol. 33, no. 5, pp. 1255–1262, Oct. 2017.

[2] C. Forster, Z. Zhang, M. Gassner, M. Werlberger, and D. Scaramuzza, "SVO: Semidirect Visual Odometry for Monocular and Multicamera Systems," *IEEE Transactions on Robotics*, vol. 33, no. 2, pp. 249–265, Apr. 2017.

[3] J. Engel, V. Koltun, and D. Cremers, "Direct Sparse Odometry," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 3, pp. 611–625, Mar. 2018.

[4] W. Wang, Y. Hu, and S. Scherer, "TartanVO: A Generalizable Learning-based VO," in *Proceedings of the 2020 Conference on Robot Learning*. PMLR, Oct. 2021, pp. 1761–1772.

[5] Z. Teed and J. Deng, "DROID-SLAM: Deep Visual SLAM for Monocular, Stereo, and RGB-D Cameras," in *Advances in Neural Information Processing Systems*, vol. 34. Curran Associates, Inc., 2021, pp. 16 558–16 569.

[6] Z. Teed, L. Lipson, and J. Deng, "Deep patch visual odometry," *Advances in Neural Information Processing Systems*, vol. 36, pp. 39 033–39 051, 2023.

[7] L. Lipson, Z. Teed, and J. Deng, "Deep Patch Visual SLAM," in *Computer Vision – ECCV 2024*, A. Leonardis, E. Ricci, S. Roth, O. Russakovsky, T. Sattler, and G. Varol, Eds. Cham: Springer Nature Switzerland, 2025, vol. 15060, pp. 424–440.

[8] B. Bescos, J. M. Fácil, J. Civera, and J. Neira, "DynaSLAM: Tracking, Mapping, and Inpainting in Dynamic Scenes," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 4076–4083, Oct. 2018.

[9] B. Bescos, C. Campos, J. D. Tardós, and J. Neira, "DynaSLAM II: Tightly-Coupled Multi-Object Tracking and SLAM," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 5191–5198, Jul. 2021.

[10] M. Hu, W. Yin, C. Zhang, Z. Cai, X. Long, H. Chen, K. Wang, G. Yu, C. Shen, and S. 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*, vol. 46, no. 12, pp. 10 579–10 596, Dec. 2024.

[11] M. Oquab, T. Darcet, T. Moutakanni, H. V. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby, M. Assran, N. Ballas, W. Galuba, R. Howes, P.-Y. Huang, S.-W. Li, I. Misra, M. Rabbat, V. Sharma, G. Synnaeve, H. Xu, H. Jegou, J. Mairal, P. Labatut, A. Joulin, and P. Bojanowski, "DINOv2: Learning Robust Visual Features without Supervision," *Transactions on Machine Learning Research*, Jul. 2023.

[12] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, Jun. 2012, pp. 3354–3361.

[13] W. Wang, D. Zhu, X. Wang, Y. Hu, Y. Qiu, C. Wang, Y. Hu, A. Kapoor, and S. Scherer, "TartanAir: A Dataset to Push the Limits of Visual SLAM," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct. 2020, pp. 4909–4916.

[14] M. Burri, J. Nikolic, P. Gohl, T. Schneider, J. Rehder, S. Omari, M. W. Achtelik, and R. Siegwart, "The EuRoC micro aerial vehicle datasets," *The International Journal of Robotics Research*, vol. 35, no. 10, pp. 1157–1163, Sep. 2016.

[15] S. Wang, R. Clark, H. Wen, and N. Trigoni, "DeepVO: Towards end-to-end visual odometry with deep Recurrent Convolutional Neural Networks," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, May 2017, pp. 2043–2050.

[16] N. Yang, L. von Stumberg, R. Wang, and D. Cremers, "D3VO: Deep Depth, Deep Pose and Deep Uncertainty for Monocular Visual Odometry," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2020, pp. 1278–1289.

[17] T. Fu, S. Su, Y. Lu, and C. Wang, "Islam: Imperative slam," *IEEE Robotics and Automation Letters*, 2024.

[18] C. M. Parameshwara, G. Hari, C. Fermüller, N. J. Sanket, and Y. Aloimonos, "DiffPoseNet: Direct Differentiable Camera Pose Estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 6845–6854.

[19] L. Lai, Z. Shangguan, J. Zhang, and E. Ohn-Bar, "XVO: Generalized visual odometry via cross-modal self-training," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 10 094–10 105.

[20] Y. Zhang, F. Tosi, S. Mattoccia, and M. Poggi, "Go-slam: Global optimization for consistent 3d instant reconstruction," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 3727–3737.

[21] Z. Teed and J. Deng, "RAFT: Recurrent All-Pairs Field Transforms for Optical Flow," in *Computer Vision – ECCV 2020*, ser. Lecture Notes in Computer Science, A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, Eds. Cham: Springer International Publishing, 2020, pp. 402–419.

[22] S. Wang, W. Li, Y. Wang, Z. Fan, Z. Huang, X. Cai, J. Zhao, and D. Li, "MambaVO: Deep Visual Odometry Based on Sequential Matching Refinement and Training Smoothing," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025, pp. 1252–1262.

[23] M. B. Azhari and D. H. Shim, "DINO-VO: A Feature-Based Visual Odometry Leveraging a Visual Foundation Model," *IEEE Robotics and Automation Letters*, vol. 10, no. 9, pp. 9152–9159, Sep. 2025.

[24] C. Yu, Z. Liu, X.-J. Liu, F. Xie, Y. Yang, Q. Wei, and Q. Fei, "DS-SLAM: A semantic visual SLAM towards dynamic environments," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 1168–1174.

[25] X. Yuan and S. Chen, "Sad-slam: A visual slam based on semantic and depth information," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 4930–4935.

[26] S. Yang and S. Scherer, "CubeSLAM: Monocular 3-D Object SLAM," *IEEE Transactions on Robotics*, vol. 35, no. 4, pp. 925–938, Aug. 2019.

[27] J. Wang, M. Rünz, and L. Agapito, "DSP-SLAM: Object Oriented SLAM with Deep Shape Priors," in *2021 International Conference on 3D Vision (3DV)*, Dec. 2021, pp. 1362–1371.

[28] Y. Chen, F. Zhao, Y. Zhuge, J. Liu, J. Yan, and H. Luo, "SMORE-SLAM: Semantic Monocular SLAM with Scale Correction and Reverse Loop Utilization in Outdoor Environments," in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct. 2024, pp. 7870–7877.

[29] X. Dong, M. A. Garratt, S. G. Anavatti, and H. A. Abbass, "Towards Real-Time Monocular Depth Estimation for Robotics: A Survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 10, pp. 16 940–16 961, Oct. 2022.

[30] S. F. Bhat, R. Birkl, D. Wofk, P. Wonka, and M. Müller, "ZoeDepth: Zero-shot Transfer by Combining Relative and Metric Depth," Feb. 2023.

[31] G. Potje, F. Cadar, A. Araujo, R. Martins, and E. R. Nascimento, "XFeat: Accelerated Features for Lightweight Image Matching," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 2682–2691.

[32] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. M. Montiel, and J. D. Tardós, "ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual–Inertial, and Multimap SLAM," *IEEE Transactions on Robotics*, vol. 37, no. 6, pp. 1874–1890, Dec. 2021.

[33] X. Gao, R. Wang, N. Demmel, and D. Cremers, "LDSO: Direct Sparse Odometry with Loop Closure," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct. 2018, pp. 2198–2204.

[34] D. Galvez-López and J. D. Tardos, "Bags of Binary Words for Fast Place Recognition in Image Sequences," *IEEE Transactions on Robotics*, vol. 28, no. 5, pp. 1188–1197, Oct. 2012.

[35] E. Rublee, V. Rabaud, K. Konolige, and G. R. Bradski, "ORB: An efficient alternative to SIFT or SURF," in *IEEE International Conference on Computer Vision, ICCV 2011, Barcelona, Spain, November 6-13, 2011*, 2011.

[36] M. Tyszkiewicz, P. Fua, and E. Trulls, "DISK: Learning local features with policy gradient," *Advances in Neural Information Processing Systems*, vol. 33, pp. 14 254–14 265, 2020.

[37] P. Lindenberger, P.-E. Sarlin, and M. Pollefeys, "Lightglue: Local feature matching at light speed," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 17 627–17 638.

[38] Z. Teed and J. Deng, "DeepV2D: Video to Depth with Differentiable Structure from Motion," in *International Conference on Learning Representations*, Feb. 2022.

[39] Y. Qiu, Y. Chen, Z. Zhang, W. Wang, and S. Scherer, "MAC-VO: Metrics-aware Covariance for Learning-based Stereo Visual Odometry," Mar. 2025.