GEOMETRY MEETS STRUCTURE: LEARNING ROBUST MONOCULAR VISUAL ODOMETRY WITH SCALE-PRESERVING CONSTRAINTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Self-supervised monocular visual odometry learns motion from appearance but lacks explicit geometric consistency at inference time. Two-view geometry provides complementary motion constraints, yet its translation is scale-ambiguous and its estimates can become unreliable under weak parallax or noisy correspondences. We investigate how learned motion predictions and explicit geometry can be selectively combined, and present *GeoStruct-VO*. Its Structure-Aware Feature Enhancement (SAFE) module augments shallow appearance features with spatially distributed responses for pose estimation. At inference time, Scale-Preserving Geometric Refinement (SPGR) assesses geometric reliability based on correspondence support, cheirality, epipolar consistency, and agreement with the learned prediction. Reliable estimates refine rotation and translation direction while retaining the network-predicted translation magnitude; otherwise, the learned pose is preserved. A Multi-frame Rotation Graph (MRG) further incorporates quality-filtered long-gap rotational constraints to reduce accumulated drift. Experiments on KITTI and cross-domain evaluations on vKITTI2 show that *GeoStruct-VO* improves translational and rotational accuracy over the evaluated learning-based methods. Component analyses further demonstrate the complementary effects of feature enhancement, selective geometric correction, and multi-frame rotation refinement. The real-time configuration processes each frame in 20.9 ms, showing that explicit geometry can improve learned odometry without online parameter adaptation.

1 INTRODUCTION

Visual odometry (VO) estimates six-degree-of-freedom camera motion from image sequences and underpins autonomous driving, mobile robotics, and augmented reality Cadena et al. (2016); Placed et al. (2023). Classical VO methods Mur-Artal & Tardós (2017); Campos et al. (2021); Engel et al. (2018) recover motion by enforcing multi-view consistency through feature correspondences or photometric alignment, but their accuracy depends on the reliability of geometric observations. Learning-based methods Wang et al. (2017); Zhou et al. (2017); Wang et al. (2021); Song et al. (2025) instead estimate camera pose from learned representations and data priors, but their generalization remains sensitive to shifts in scene appearance and motion patterns. Although advances in network architectures and temporal modeling Wu & Zhu (2024); Xu et al. (2025); Dou et al. (2026) have strengthened motion representations, they neither guarantee the preservation of motion-relevant structural cues during feature encoding nor explicitly enforce geometric consistency of predicted poses at inference time.

Scene boundaries, widely used to delineate objects and regions in segmentation Li et al. (2020) and remote sensing Wang et al. (2026), can also provide useful structural cues for motion estimation. When spatially well distributed, these boundaries and local intensity transitions complement high-level appearance features by providing motion-relevant information across different image regions. However, during early downsampling, the simple local convolutional stacks used in many pose encoders Zhou et al. (2017); Bian et al. (2019); Zhang et al. (2023) can produce responses dominated by high-contrast regions, leaving subtle yet motion-informative structural cues underrepresented. This motivates a learnable mechanism that preserves these less prominent cues and adaptively ag-

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

gregates structural responses according to their relevance to pose estimation. Such a mechanism can be jointly optimized with the pose network without requiring predefined edge operators.

A richer pose representation, however, does not ensure that the predicted motion satisfies the geometric relations observed at inference time. Although view reconstruction provides geometric supervision during training, direct pose regression does not explicitly verify individual predictions against inter-frame correspondences. Two-view geometry provides complementary constraints and can recover relative rotation accurately when reliable correspondences are available. Its translation, however, is observable only up to scale and can become unstable under noisy correspondences, weak parallax, or degenerate motion. The learned pose and geometric estimate therefore contribute complementary information: the network provides a translation magnitude learned from data, whereas calibrated two-view geometry provides rotation and translation direction. This complementarity motivates a selective refinement rule that applies geometric correction only when the estimate is sufficiently reliable.

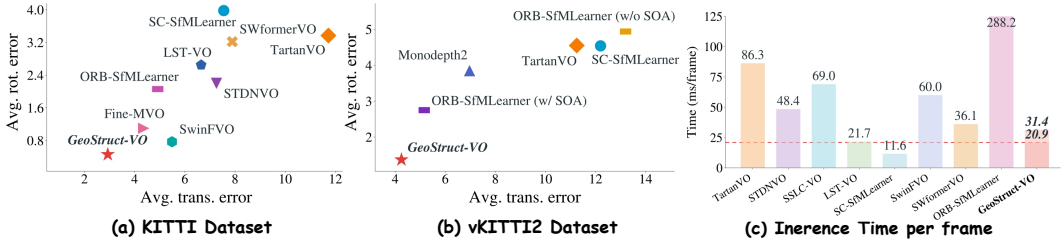


Figure 1: Accuracy–efficiency comparison with representative learning-based VO methods. *GeoStruct-VO* consistently achieves the lowest average translational and rotational errors on both KITTI and vKITTI2 while maintaining competitive inference speed, demonstrating a favorable trade-off between estimation accuracy and computational efficiency.

Motivated by these observations, we propose *GeoStruct-VO*, a self-supervised monocular VO framework that integrates representation learning with selective geometric optimization. Structure-Aware Feature Enhancement (SAFE) enriches shallow pose features through learnable structural feature extraction and adaptive spatial–channel aggregation, preserving subtle responses that are informative for motion estimation. Scale-Preserving Geometric Refinement (SPGR) then formulates inference-time fusion as a reliability-gated correction process. It uses two-view geometry that satisfies geometric reliability and prediction-agreement checks to refine rotation and translation direction while retaining the network-predicted translation magnitude; otherwise, the learned pose is preserved. Finally, Multi-frame Rotation Graph (MRG) incorporates quality-filtered cross-frame rotational constraints through a decoupled back end. Together, these components separate online frame-to-frame estimation from trajectory-level correction while explicitly distinguishing the information contributed by learning from that provided by geometry. Figure 1 compares *GeoStruct-VO* with representative learning-based methods in pose accuracy on KITTI and vKITTI2 and in per-frame inference time. Our contributions are summarized as follows:

- We formulate learned monocular motion estimation and two-view geometry as a selective correction problem, exploiting their complementary roles: learning provides metric translation magnitude, while geometry constrains rotation and translation direction. This formulation enables geometry to intervene only when its evidence is sufficiently reliable.
- We develop *GeoStruct-VO* as a unified learning–geometry framework operating at representation, frame-to-frame, and trajectory levels. SAFE strengthens motion-relevant structural representations, SPGR performs reliability-gated refinement of rotation and translation direction while preserving the learned translation magnitude, and MRG incorporates quality-filtered long-gap rotational constraints to further reduce accumulated rotational drift.
- Extensive evaluations on KITTI and cross-domain vKITTI2 show that *GeoStruct-VO* achieves the lowest average translational and rotational errors among the compared learning-based methods while maintaining practical inference efficiency. Component-wise analyses further validate the complementary contributions of structural enhancement, selective geometric correction, and multi-frame refinement.

2 RELATED WORK

Classical geometry-based VO. Feature-based systems such as ORB-SLAM3 Campos et al. (2021) use feature matching, landmark reconstruction, and multi-view optimization to estimate camera motion. These systems are geometrically interpretable but depend on reliable data association, sufficient parallax, and spatially well-distributed features. Direct methods such as DSO Engel et al. (2018) instead optimize photometric residuals and avoid keypoint matching, but their brightness-constancy assumption can be violated by illumination or exposure changes, motion blur, and dynamic content.

Edge-based systems further exploit boundaries and contours as complementary structural cues Zhao et al. (2023; 2025). Handcrafted operators are efficient and interpretable, but their fixed responses are not optimized for the motion-estimation objective. GeoStruct-VO instead learns local response filters jointly with the pose network. However, improving the representation alone does not address the inherent failure modes of monocular geometry; accordingly, the inference stage separately determines whether a geometric estimate is sufficiently reliable to refine the learned pose.

Learning-based and hybrid VO. End-to-end methods use architectures such as CRNNs and Vision Transformers to learn spatial and temporal motion representations directly from image sequences Wang et al. (2017); Wu & Zhu (2024); Xu et al. (2025); Song et al. (2025); Zhou et al. (2017); Bian et al. (2019); Godard et al. (2019). Despite differences in temporal modeling, many of these approaches rely on conventional convolutional stems and infer motion primarily from learned appearance representations. Hybrid systems retain explicit geometric optimization while learning selected components: DROID-SLAM Teed & Deng (2021) and DPVO Teed et al. (2023) combine learned correspondences with differentiable bundle adjustment, whereas DF-VO Zhan et al. (2020) incorporates learned optical flow and uncertainty into geometric estimation. These methods demonstrate the value of combining learning with geometry, but monocular translation remains scale-ambiguous and geometric estimates remain sensitive to degenerate conditions. Our focus is different: rather than jointly optimizing all quantities, we selectively inject two-view geometric correction only when it is sufficiently reliable, while preserving the learned translation magnitude.

Explicit point features have also been used to guide self-supervised pose learning Jin et al. (2025), but they capture scene structure only at sparse locations. In other vision tasks, learned boundary representations and multi-scale structural fusion have been used to preserve spatial discontinuities and improve discrimination Wang et al. (2026); Li et al. (2020). Motivated by these observations, SAFE learns complementary local responses within the pose encoder rather than imposing a fixed edge map. GeoStruct-VO then combines this representation-level enhancement with reliability-gated geometric correction, separating feature learning from inference-time constraint injection.

3 PROPOSED METHOD: GEOSTRUCT-VO

Figure 2 summarizes GeoStruct-VO. Given two consecutive monocular frames, the depth network predicts their disparity maps, and the pose network encodes the RGB and disparity pairs in separate branches. SAFE augments the RGB branch through adaptive local filtering and spatial–channel aggregation, while the disparity branch extracts complementary depth-derived cues. Following MLF-VO Jiang et al. (2022), channel exchange facilitates intermediate interaction between the branches before the fused representation is decoded into relative motion. The depth and pose networks are trained jointly using the self-supervised view-reconstruction objective detailed in Appendix A.2. At inference time, SPGR recovers a geometric motion estimate from sparse correspondences and applies it only when all reliability criteria are satisfied. MRG subsequently optimizes frame-wise rotational corrections within a sliding window using sequential smoothness and filtered long-gap constraints. As a decoupled back end, MRG does not affect the reported frame-to-frame latency, although it introduces additional asynchronous computation.

3.1 STRUCTURE-AWARE FEATURE ENHANCEMENT

Early downsampling can attenuate fine-scale responses that are informative for camera-motion estimation. SAFE addresses this issue through two stages: a learnable local feature stem that retains local-response information during resolution reduction, and a structure-aware aggregation block that integrates complementary spatial responses and recalibrates their channels. We use "structure-

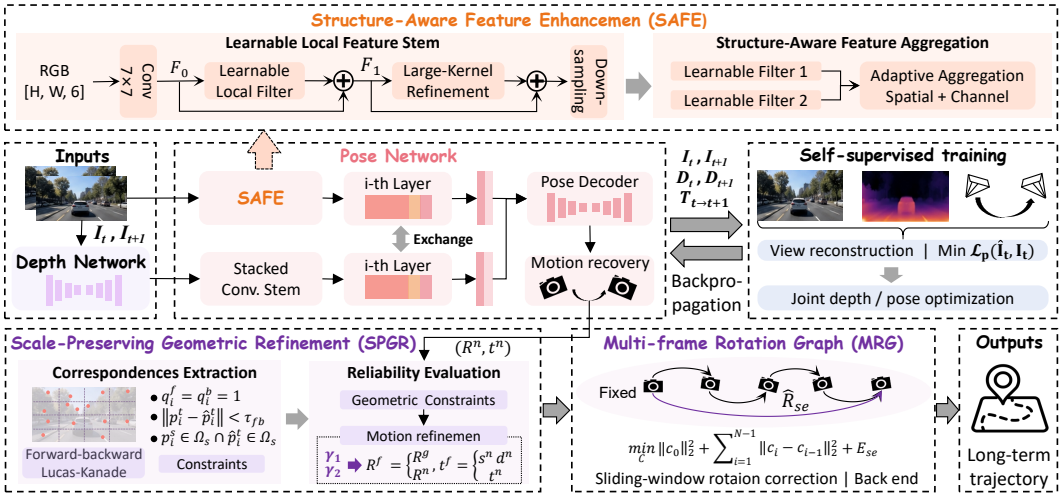


Figure 2: GeoStruct-VO comprises dual-stream pose estimation with SAFE, reliability-gated SPGR for local pose refinement, and a decoupled MRG for multi-frame orientation consistency.

aware" to describe this architectural bias toward spatially differentiated local responses; SAFE does not rely on explicit edge labels or structural supervision.

Learnable Local Feature Stem. Given the concatenated images, a standard 7×7 convolution first produces the base feature F_0. A learnable depthwise filter is then applied through residual fusion:

$$F_1 = \mathrm{BN}\left[F_0 + \mathrm{ReLU}\left(\mathrm{BN}\left(\mathrm{DWConv}_{7\times7}(F_0)\right)\right)\right]. \tag{1}$$

Because the convolution is depthwise, it learns a separate local filter for each channel, while residual fusion preserves the base representation. The feature is then downsampled and refined with learnable large-kernel filtering:

$$F_2 = \mathrm{BN}\left[F_d + \mathcal{L}(F_d)\right], \qquad F_d = \mathcal{D}(F_1), \tag{2}$$

where $\mathcal{D}$ performs channel projection and spatial downsampling, and $\mathcal{L}$ captures a wider spatial context. To retain complementary responses during further resolution reduction, F_2 is processed by a dual-branch residual block:

$$F_3 = \mathcal{R}(F_2), F_4 = \mathrm{Conv}_{1\times1}\left([\mathcal{B}_c(F_3), \mathcal{B}_p(F_3)]\right), \tag{3}$$

where $\mathcal{R}$ performs channel expansion and residual refinement. The convolutional branch $\mathcal{B}_c$ learns the downsampling transformation, whereas the max-pooling branch $\mathcal{B}_p$ preserves locally prominent responses. Their concatenation and projection produce F_4 at one-quarter of the spatial resolution of F_1.

Structure-Aware Feature Aggregation. The aggregation stage combines complementary local responses while preserving the original appearance representation. Two independently parameterized depthwise filters first process F_4:

$$G_i = \mathrm{DWConv}^{(i)}_{3\times3}(F_4),\ i \in \{1,2\}, \quad G = \mathrm{ReLU}\left(\mathrm{BN}\left(\sqrt{G_1^2 + G_2^2 + \epsilon}\right)\right) \tag{4}$$

Their squared responses are combined channel-wise before normalization and activation, yielding a nonnegative response magnitude G. This magnitude is integrated with F_4 through a lightweight transformation and then modulates the spatial representation through residual interaction:

$$F_5 = \Phi_r(F_4 + G), \quad F_{\mathrm{attn}} = \Phi_s(F_4 \odot F_5 + F_4). \tag{5}$$

where Φ_r comprises successive 1×1, 3×3, and 1×1 convolutions, and Φ_s is a 3×3 spatial aggregation layer. The residual terms retain the input feature while allowing the learned responses to modulate spatial locations.

Channel recalibration then summarizes F_{attn} through global average pooling and produces weights w through a one-dimensional convolution and sigmoid function. The weighted response is fused residually with F_4:

$$w = \sigma\left(\psi_c\left(\mathrm{GAP}(F_{attn})\right)\right), \quad F_g = \mathrm{BN}\left(F_4 + w \odot F_{attn}\right), \tag{6}$$

where GAP denotes global average pooling, ψ_c is the one-dimensional channel interaction, and σ is the sigmoid function. The resulting F_g is passed to the subsequent pose-encoder layers.

3.2 Scale-Preserving Geometric Refinement

Calibrated two-view geometry constrains relative rotation and translation direction, but the magnitude of monocular translation is unobservable. SPGR therefore treats the network prediction and geometric estimate as complementary rather than interchangeable. An essential matrix estimated from 2D–2D correspondences is decomposed into a geometric rotation R^g and unit translation direction d^g. This construction does not require predicted depth to form 3D points, thereby avoiding dependence on predicted depth during geometric recovery. Let (R^n, t^n) denote the network-predicted pose and $s^n \triangleq \|t^n\|_2$ its translation magnitude. SPGR defines the refined pose as

$$R^f = \begin{cases} R^g, & \gamma_1 = 1, \\ R^n, & \gamma_1 = 0, \end{cases} \qquad t^f = \begin{cases} s^n d^g, & \gamma_1 = \gamma_2 = 1, \\ t^n, & \text{otherwise.} \end{cases} \tag{7}$$

where $\gamma_1, \gamma_2 \in \{0, 1\}$ indicate geometric and translation-direction reliability, respectively, and $|d^g|_2 = 1$. The sparse-correspondence and geometric evaluation below determine γ_1. Since passing these checks does not ensure a reliable translation direction, γ_2 further evaluates directional agreement and temporal consistency, as detailed in Appendix A.4.

Sparse Correspondence Extraction. Shi–Tomasi corners detected in the target frame are tracked to the source frame using pyramidal Lucas–Kanade optical flow. Each forward track $p_i^t \to p_i^s$ is subsequently tracked backward to obtain $\hat{p}_i^t$. We validate tracks using the forward–backward error, the binary tracking-success indicators q_i^f and q_i^b, and image-domain constraints:

$$e_i^{fb} = \left\| p_i^t - \hat{p}_i^t \right\|_2, \quad \mathcal{V} = \left\{ i \;\middle|\; q_i^f = q_i^b = 1, p_i^s \in \Omega_s, \hat{p}_i^t \in \Omega_t, e_i^{fb} < \tau_{fb} \right\}, \tag{8}$$

where Ω_t and Ω_s denote the valid image domains and τ_{fb} is the maximum forward–backward error in pixels. We divide the target image into a regular grid to prevent a small number of highly textured regions from dominating geometric estimation. Within each cell, valid tracks are ranked by e_i^{fb}, and at most K tracks are retained. The resulting spatially distributed set $\mathcal{M}$ is used to estimate the essential matrix.

Geometric Reliability Evaluation. Geometric estimation is attempted only when the valid tracks provide sufficient support and displacement:

$$|\mathcal{M}| \geq N_{\min}, \quad \text{median}_{i \in \mathcal{M}} \left\| p_i^s - p_i^t \right\|_2 \geq \tau_p. \tag{9}$$

We then estimate the essential matrix using RANSAC and decompose it to recover (R^g, d^g) detailed in Appendix A.3. Let $\mathcal{I}_r \subseteq \mathcal{M}$ denote the RANSAC inliers and $\mathcal{I}_g \subseteq \mathcal{I}_r$ the subset whose triangulated points have positive depth in both views. We require both sufficient support among all valid tracks and a sufficient cheirality-consistent fraction among the RANSAC inliers:

$$\frac{|\mathcal{I}_g|}{|\mathcal{M}|} \geq \tau_i, \qquad \frac{|\mathcal{I}_g|}{\max(1, |\mathcal{I}_r|)} \geq \tau_c. \tag{10}$$

Because a geometrically feasible solution can still be inconsistent with the learned motion estimate, we bound the relative rotation discrepancy and require agreement between the translation directions:

$$\text{ang}\big(R^g (R^n)^\top\big) \leq \tau_R, \qquad \frac{(d^g)^\top t^n}{\|d^g\|_2 \|t^n\|_2} \geq \tau_t. \tag{11}$$

where $\text{ang}(\cdot)$ returns the rotation angle. This agreement test serves only to reject large discrepancies and does not treat the network prediction as ground truth.

Finally, we assess the epipolar consistency of the geometric candidate using $E^g = [d^g]_\times R^g$. Let x_i^t and x_i^s denote normalized homogeneous coordinates, and let d_S denote the square root of the Sampson error Zhang & Ye (2021). We require the mean residual over $\mathcal{I}_g$ to satisfy

$$\frac{1}{|\mathcal{I}_g|} \sum_{i \in \mathcal{I}_g} d_S\big(x_i^t, x_i^s; E^g\big) \leq \tau_e. \tag{12}$$

3.3 MULTI-FRAME ROTATION GRAPH

SPGR corrects individual frame pairs but does not explicitly distribute residual rotational error over longer temporal intervals. MRG addresses this limitation by estimating frame-wise rotation corrections from reliable cross-frame constraints within a sliding window.

For each window of N frames, MRG evaluates the endpoint pair (s, e) using the applicable geometry-only validity checks adopted in SPGR. If the pair is accepted, two-view geometry provides the relative rotation $\hat{R}_{se}$. We compare this estimate with the relative rotation obtained by composing the locally refined poses and map their discrepancy to the world-frame tangent space:

$$d_{se} = R_s^f \left[\log \left(\hat{R}_{se} \left((R_s^f)^T R_e^f \right)^T \right) \right]^\vee . \tag{13}$$

Here, $\log(\cdot)$ maps the rotational discrepancy from $\mathrm{SO}(3)$ to $\mathfrak{so}(3)$, and $(\cdot)^\vee$ converts it into the vector $d_{se} \in \mathbb{R}^3$. We discard non-finite residuals and endpoint pairs with $\|d_{se}\|_2 > 10^\circ$; the remaining pairs define the accepted cross-frame constraints.

Let $c_i \in \mathbb{R}^3$ denote the world-frame small-angle correction for frame i, and let $C \in \mathbb{R}^{N\times 3}$ stack these corrections row-wise. For a window with an accepted endpoint constraint, we estimate C using an anchor term, temporal smoothness, and the weighted endpoint residual:

$$\min_C \ \|c_0\|_2^2 + \sum_{i=1}^{N-1} \|c_i - c_{i-1}\|_2^2 + E_{se}, \quad E_{se} = w_{se} \left\| (c_e - c_s) - d_{se} \right\|_2^2 , \tag{14}$$

where w_{se} is the confidence weight assigned to the endpoint constraint. The first term fixes the gauge by anchoring the initial correction, the second distributes corrections smoothly across adjacent frames, and E_{se} enforces agreement with the accepted long-gap rotation. The resulting quadratic least-squares problem is solved as a sparse linear system. The i-th row of C provides the estimated correction, which is applied to the current rotation using the update rule detailed in Appendix A.5.

4 EXPERIMENTS AND RESULTS

4.1 EXPERIMENTAL SETUP

Datasets and protocols. We evaluate GeoStruct-VO on KITTI Geiger et al. (2013) and its cross-domain generalization on Virtual KITTI2 (vKITTI2) Cabon et al. (2020). Following SwinFVO Song et al. (2025), we train on KITTI sequences 00–08 and test on 09–10. For vKITTI2, we follow ORB-SfMLearner Jin et al. (2025): the model is trained on the KITTI Eigen split Eigen et al. (2014) and tested without fine-tuning on Scenes 01 and 20 under Clone (C), Fog (F), and Morning (M), covering nominal appearance, reduced visibility, and altered illumination. Additional KITTI results under the Eigen-split protocol are given in Appendix A.7.

Baselines. For the KITTI benchmark, we compare established and recent methods across three VO paradigms. The geometric references are DSO Engel et al. (2018) and ORB-SLAM3 Campos et al. (2021), with loop closure disabled for the latter. Supervised baselines include TartanVO Wang et al. (2021), SWformer-VO Wu & Zhu (2024), STDNVO Xu et al. (2025), SSLC-VO Li & He (2026), and LST-VO Dou et al. (2026). Self-supervised baselines include SfMLearner Zhou et al. (2017), Depth-VO-Feat Zhan et al. (2018), Monodepth2 Godard et al. (2019), SC-SfMLearner Bian et al. (2019), Sun et al. Sun et al. (2022), CM-VO Liu et al. (2022), GraphAVO Song et al. (2024), KD-VO Li et al. (2022), Fine-MVO Wei et al. (2024b), Lite-Mono Zhang et al. (2023), Lite-SVO Wei et al. (2024a), SwinFVO Song et al. (2025), and ORB-SfMLearner Jin et al. (2025). For vKITTI2, we select baselines with publicly available implementations and evaluate them on the same scenes and conditions to assess cross-domain transfer. These include the supervised methods TartanVO and LST-VO, the self-supervised methods SfMLearner, SC-SfMLearner, and Monodepth2, and ORB-SfMLearner with and without selective online adaptation (SOA). The two ORB-SfMLearner configurations isolate the effect of test-time adaptation.

Implementation Details. We implement GeoStruct-VO in PyTorch 2.4 and train it on a single RTX 4090D GPU using three-frame samples at a resolution of 192×640. Both branches use ImageNet-pretrained ResNet-18 backbones. The model is optimized with Adam for 40 epochs using a batch size of 12, with the learning rate alternating between 10^{-4} and 5×10^{-5} every 10 epochs.

SPGR tracks up to 500 corners on KITTI and 1,000 on vKITTI2 at 0.8 image scale. Tracked correspondences are distributed across a 4×8 spatial grid, with at most $K = 50$ retained per cell. We set $N_{\min} = 50$, $\tau_{fb} = \tau_p = 0.5$, $\tau_i = \tau_c = 0.05$, $\tau_R = 20°$, $\tau_t = -0.1$, and $\tau_e = 0.30$. RANSAC uses a threshold of 0.5, a confidence of 0.99, a maximum of 25 iterations, and a minimum inlier ratio of 0.70. The SPGR hyperparameters are selected through sensitivity experiments and kept fixed across both datasets. MRG uses temporal offsets of three and five frames.

Evaluation Metrics. We use the same evaluation tool employed in LST-VO Dou et al. (2026) to align trajectories and report the standard translational error t_{rel} and rotational error r_{rel}, computed over trajectory segments ranging from 100 to 800 m. We additionally report Absolute Trajectory Error (ATE) to quantify global trajectory consistency.

4.2 Results

Results on KITTI. Table 1 reports results on test Seq. 09 and 10. GeoStruct-VO achieves the lowest average t_{rel} among all compared methods and the lowest average r_{rel} among learning-based methods. DSO and ORB-SLAM3, both of which exploit explicit multi-frame geometric optimization, report lower average r_{rel} of $0.22°/100$ m and $0.38°/100$ m, respectively, compared with $0.46°/100$ m for GeoStruct-VO. In contrast, GeoStruct-VO reduces their average translational errors by 88.8% and 25.4%, respectively. This complementary behavior is consistent with the design of SPGR, which uses geometric rotation and translation direction while retaining the network-predicted translation magnitude. Compared with the recent Mamba-based dual-stream supervised LST-VO, GeoStruct-VO reduces average t_{rel} and r_{rel} by 55.9% and 82.6%, respectively. Fine-MVO enhances fine-grained features in dynamic scenes, yet GeoStruct-VO lowers its two errors by 32.8% and 58.2%. Relative to SwinFVO, which combines a Swin Transformer with optical flow, the reductions are 46.6% and 40.3%. Even against ORB-SfMLearner with ORB-guided features and selective online adaptation (SOA), they reach 40.4% and 77.7%. These results are consistent with GeoStruct-VO combining learned motion and geometry refinement. The real-time GeoStruct-VO† achieves average t_{rel} and r_{rel} of 3.12% and $0.50°/100$ m; adding MRG further reduces them by 6.1% and 8.0%.

Table 1: Performance comparison on KITTI. Best learning-based results are **bold**; second-best are underlined.

Type	Method	Seq. 09		Seq. 10		Average	
		t_{rel}	r_{rel}	t_{rel}	r_{rel}	t_{rel}	r_{rel}
Geo.	DSO	28.10	0.21	24.00	0.22	26.05	0.22
	ORB-SLAM3	3.02	0.43	4.84	0.33	3.93	0.38
Sup.	TartanVO	10.03	2.21	13.41	4.53	11.72	3.37
	SWformer-VO	8.71	2.94	7.07	3.49	7.89	3.22
	STDNVO	7.39	2.49	7.13	1.90	7.26	2.20
	SSLC-VO	7.65	2.56	5.81	3.63	6.73	3.10
	LST-VO	6.84	2.23	6.46	3.07	6.65	2.65
Self-Sup.	SfMLearner	16.28	4.23	9.97	5.02	13.13	4.63
	Depth-VO-Feat	11.93	3.91	12.45	3.46	12.19	3.69
	Monodepth2	14.84	3.30	7.73	3.40	11.29	3.35
	SC-SfMLearner	7.31	3.05	7.79	4.90	7.55	3.98
	Sun *et al.*	7.14	2.30	7.72	2.30	7.43	2.30
	CM-VO	9.69	3.40	7.05	2.90	8.37	3.15
	GraphAVO	5.87	1.53	9.07	3.63	7.47	2.58
	KD-VO	6.33	1.88	10.75	2.61	8.54	2.25
	Lite-Mono	9.14	2.00	8.92	3.20	9.03	2.60
	Lite-SVO	7.20	1.50	8.50	2.40	7.85	1.95
	Fine-MVO	3.37	1.00	5.34	1.20	4.36	1.10
	SwinFVO	4.63	0.72	6.34	0.81	5.49	0.77
	ORB-SfMLearner	3.85	1.67	5.98	2.44	4.92	2.06
	GeoStruct-VO†	3.14	0.51	3.09	0.49	3.12	0.50
	GeoStruct-VO	**2.81**	**0.45**	**3.04**	**0.46**	**2.93**	**0.46**

† denotes the real-time configuration (w/o MRG).

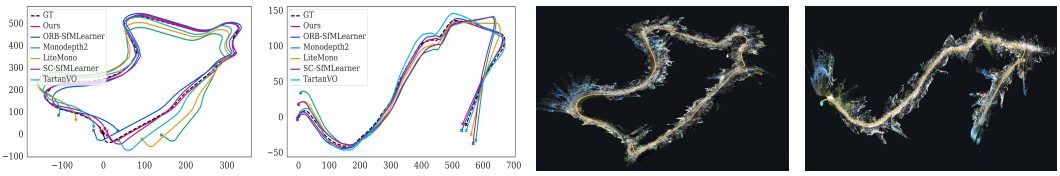


Figure 3: Trajectory comparisons on KITTI sequences 09 and 10.

Figure 3 qualitatively compares trajectories from representative open-source methods on Seq. 09 and 10. In the displayed examples, GeoStruct-VO follows the reference trajectory more closely over several trajectory segments, in agreement with the lower aggregate errors reported in Table 1. The final two panels provide qualitative dense-mapping visualizations only and are not used to support a quantitative mapping claim.

Generalization Evaluation on vKITTI2. Table 2 evaluates transfer from KITTI to the selected vKITTI2 scenes without fine-tuning. GeoStruct-VO achieves the lowest average ATE, t_{rel} and r_{rel} among the compared methods, at 5.36, 4.25, and 1.37, respectively. Compared with LST-VO, GeoStruct-VO reduces average ATE, t_{rel}, and r_{rel} by 37.3%, 46.4%, and 57.1%, respectively. Relative to Monodepth2, the corresponding reductions are 27.5%, 39.0%, and 64.4%. It also achieves the lowest r_{rel} in all six evaluated conditions, with the largest gains occurring in rotation. ORB-SfMLearner with Selective Online Adaptation (SOA) is the strongest competing baseline because it explicitly addresses domain shift through test-time adaptation. In contrast, GeoStruct-VO keeps its network parameters fixed at inference time and relies on reliability-gated geometric refinement. Even so, the full model reduces the SOA-enabled ORB-SfMLearner's average ATE, t_{rel}, and r_{rel} by 7.7%, 17.6%, and 50.2%, respectively. Adding MRG further improves GeoStruct-VO† from 6.41, 4.55, and 1.49 to 5.36, 4.25, and 1.37, supporting the benefit of multi-frame refinement under cross-domain transfer.

Table 2: Performance comparison on vKITTI2. Best learning-based results are **bold**; second-best are underlined.

Method	Metric	Scene 01			Scene 20			Avg.
		C	F	M	C	F	M	
TartanVO	ATE	6.15	6.59	6.36	16.25	22.37	17.56	12.55
	t_{rel}	8.83	8.28	9.44	9.84	18.04	13.07	11.25
	r_{rel}	5.37	5.54	4.84	4.29	3.81	3.46	4.55
LST-VO	ATE	2.50	2.15	3.59	13.33	19.80	9.92	8.55
	t_{rel}	3.63	3.80	6.12	8.68	15.21	10.16	7.93
	r_{rel}	2.60	2.08	2.85	3.41	5.59	2.58	3.19
SfMLearner	ATE	35.77	24.32	35.15	16.40	17.64	15.33	24.10
	t_{rel}	54.16	36.89	52.22	11.75	13.95	9.68	29.78
	r_{rel}	35.99	22.82	34.87	4.58	5.26	3.96	17.91
SC-SfMLearner	ATE	4.58	7.59	5.58	13.51	51.74	12.72	15.95
	t_{rel}	5.27	9.63	6.38	9.20	33.52	9.18	12.20
	r_{rel}	4.09	6.46	5.13	1.95	7.97	1.66	4.54
Monodepth2	ATE	2.48	3.92	3.54	8.07	12.32	14.00	7.39
	t_{rel}	3.02	4.62	3.15	6.79	11.11	13.10	6.97
	r_{rel}	2.76	5.89	3.46	2.64	3.92	4.41	3.85
ORB-SfMLearner (w/o SOA)	ATE	2.58	5.58	6.47	12.04	44.13	15.63	14.41
	t_{rel}	4.85	8.17	10.21	9.13	37.15	9.68	13.20
	r_{rel}	3.20	2.76	3.05	3.59	13.39	3.64	4.94
ORB-SfMLearner (w/ SOA)	ATE	1.96	**1.18**	**1.48**	8.98	**10.61**	10.63	5.81
	t_{rel}	2.99	**2.42**	2.83	7.47	**6.63**	8.59	5.16
	r_{rel}	2.63	2.67	3.11	2.60	2.64	2.82	2.75
GeoStruct-VO†	ATE	1.34	3.57	2.81	**4.61**	16.43	9.68	6.41
	t_{rel}	**1.72**	4.82	**2.29**	**3.09**	10.75	4.64	4.55
	r_{rel}	0.78	1.93	1.93	0.37	2.50	1.40	1.49
GeoStruct-VO	ATE	**1.32**	3.35	2.64	4.72	11.03	**9.12**	**5.36**
	t_{rel}	1.86	4.74	2.46	**3.09**	8.93	**4.39**	**4.25**
	r_{rel}	**0.77**	**1.63**	**1.79**	**0.34**	**2.44**	**1.22**	**1.37**

† denotes the real-time configuration (w/o MRG).

Accuracy–Efficiency Analysis. Table 3 reports the accuracy–efficiency comparison. GeoStruct-VO† requires 20.9 ms/frame for online pose estimation while achieving average t_{rel} and r_{rel} of 3.12% and $0.50°/100$ m, respectively. Several baselines report lower inference times, including SfMLearner, Monodepth2, SC-SfMLearner, whereas GeoStruct-VO† provides substantially lower odometry errors. We therefore interpret the comparison as an accuracy–efficiency trade-off rather than claiming uniformly lower runtime. GeoStruct-VO with MRG introduces an additional 10.5 ms/frame of asynchronous back-end computation and reduces average t_{rel} and r_{rel} by a further 6.1% and 8.0%, respectively. Because MRG is decoupled from frame-to-frame pose estimation, the online pose-estimation latency remains 20.9 ms/frame. The component-wise computational cost is detailed in Appendix A.9.

Table 3: Comparison of average error and inference time on KITTI seq. 09 and 10.

Method	Time (ms/frame) ↓	t_{rel} ↓	r_{rel} ↓
TartanVO	86.3	11.72	3.37
SSLC-VO	69.0	6.73	3.10
STDNVO	48.4	7.26	2.20
LST-VO	21.7	6.65	2.65
SfMMearner	**3.63**	13.13	4.63
SC-SfMLearner	11.6	7.55	3.98
Monodepth2	7.05	11.29	3.35
SwinFVO	$\geq$ 60.0	5.49	0.77
ORB-SfMLearner	288.17	4.92	2.06
GeoStruct-VO†	20.9	3.12	0.50
GeoStruct-VO	31.4	**2.93**	**0.46**

† denotes the real-time configuration (w/o MRG)

4.3 Ablation Study

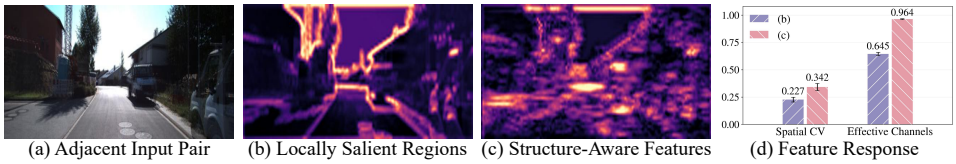
We evaluate the contributions of SAFE, SPGR, and MRG using t_{rel} and r_{rel} averaged over KITTI Seq. 09 and 10. All trainable variants use the same backbone and training protocol, with only the indicated component removed.

Component Ablation. Table 4 evaluates the contributions of SAFE, SPGR, and MRG on KITTI Seq. 09 and 10. The full GeoStruct-VO reduces the baseline t_{rel} from 5.26% to 2.93% and r_{rel} from $1.97°/100$ m to $0.46°/100$ m, corresponding to reductions of 44.3% and 76.6%. SPGR provides the largest marginal improvement: removing it increases t_{rel} to 4.24% and r_{rel} to $1.21°/100$ m. Restoring SPGR reduces these errors by 30.9% and 62.0%, highlighting its dominant contribution to rotational accuracy. SAFE further reduces t_{rel} from 3.60% to 2.93%

Table 4: Component ablation of GeoStruct-VO.

Configuration	Time (ms/frame) ↓	t_{rel} ↓	r_{rel} ↓
Baseline	8.9	5.26	1.97
w/o SAFE	27.9	3.60	0.46
w/o SPGR	24.5	4.24	1.21
w/o MRG	20.9	3.12	0.50
GeoStruct-VO	31.4	2.93	0.46

with an additional 3.5 ms/frame, while r_{rel} remains unchanged at the reported precision, suggesting that SAFE mainly benefits translation once geometric refinement is active. MRG further improves t_{rel} from 3.12% to 2.93% and r_{rel} from $0.50°/100$ m to $0.46°/100$ m at a back-end cost of 10.5 ms/frame. Overall, SAFE strengthens the learned representation, SPGR provides the largest frame-to-frame accuracy gain, and MRG adds complementary multi-frame refinement. Figure 4 further illustrates the effect of SAFE on the learned representation.The spatial CV increases from 0.227 to 0.342 and the normalized effective number of channels from 0.645 to 0.964, indicating more spatially differentiated responses and broader channel participation. These observations complement the quantitative ablation by showing how SAFE changes the feature-response distribution.



(a) Adjacent Input Pair (b) Locally Salient Regions (c) Structure-Aware Features (d) Feature Response

Figure 4: Feature responses before and after SAFE. (a) Input image pair. (b) Baseline responses. (c) SAFE-enhanced responses. (d) Mean response statistics over 200 samples.

Pose Refinement Ablation in SPGR. With MRG disabled, we isolate the contributions of geometric rotation, translation direction, and the learned translation magnitude in Figure 5. The Pose Network (PN) and Pure Geometry (PG) exhibit complementary error profiles: PN achieves substantially lower t_{rel} than PG (4.39% vs. 10.28%), whereas PG reduces r_{rel} from $1.34°/100$ m to $0.50°/100$ m. This motivates SPGR to retain the learned translation magnitude while introducing geometric rotation and direction. Rotation-only refinement (R_g) reduces r_{rel} to $0.50°/100$ m, while direction-only refinement (d_g) lowers t_{rel} to 3.36%. Joint refinement achieves the lowest t_{rel} of 3.12%, improving over R_g and d_g by 16.0% and 7.1%, respectively, while retaining the lowest rotational error. These results support the scale-preserving design of SPGR. Additionally, the contribution of SPGR's reliability gates is evaluated in Appendix A.8.

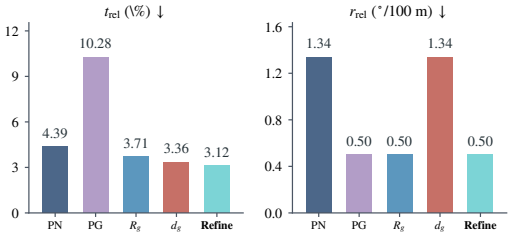


Figure 5: Ablation study of geometric refinement.

Temporal Window Analysis in MRG.

Figure 6 studies the effect of the temporal window size W with all other settings fixed. Increasing W from 1 to 5 reduces t_{rel} from 3.12% to 2.93% and r_{rel} from $0.50°/100$ m to $0.46°/100$ m, corresponding to improvements of 6.1% and 8.0%, respectively. Although $W = 7$ achieves the lowest rotational error of $0.44°/100$ m, its translational error increases to 2.98%. We therefore adopt $W = 5$, which yields the lowest translational error while maintaining low rotational error. The non-monotonic behavior for larger windows further indicates that increasing the temporal span does not consistently improve estimation accuracy.

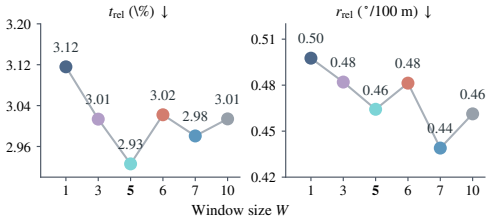


Figure 6: Effect of the window size in MRG.

5 CONCLUSION

We presented GeoStruct-VO, a self-supervised monocular VO framework that selectively combines learned motion estimation with explicit geometry. SAFE enhances shallow pose features, SPGR refines rotation and translation direction only when geometric estimates are reliable while preserving the learned translation magnitude, and MRG adds quality-filtered long-gap rotational constraints. On KITTI sequences 09–10, GeoStruct-VO achieves average translational and rotational errors of 2.93% and $0.46°/100$ m, while the real-time configuration runs at 20.9 ms/frame. Cross-domain evaluation on vKITTI2 further shows improved accuracy over the evaluated learning-based methods without test-time parameter adaptation. These results support selective geometric correction as an effective complement to learned monocular motion estimation.

AI USE STATEMENT

We used generative AI tools solely for manuscript language editing, including grammar checking, sentence refinement, and improving clarity and readability. These tools were not used for method development, algorithm implementation, data generation, or experimental analysis. We reviewed the AI-assisted revisions to ensure that they preserved the intended technical meaning and did not introduce unsupported claims. We take full responsibility for the final content of this work.

ETHICS STATEMENT

This work investigates self-supervised monocular visual odometry using the publicly available KITTI and vKITTI2 benchmarks. It does not involve human-subject experiments or the collection of new personal data. We have not identified additional ethical concerns specific to the methodology and experiments presented in this study.

REPRODUCIBILITY STATEMENT

The experimental setup and Appendix A.6 describe the hardware and software environment, backbone initialization, input configuration, training schedule, and geometric refinement settings. Additional details of the training objective, SPGR reliability checks, and MRG updates are provided in the appendix. Appendix A.9 specifies the timing protocol and component measurements. These details are provided to facilitate reproduction of the reported results.

REFERENCES

Jiawang Bian, Zhichao Li, Naiyan Wang, Huangying Zhan, Chunhua Shen, Ming-Ming Cheng, and Ian Reid. Unsupervised scale-consistent depth and ego-motion learning from monocular video. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

Yohann Cabon, Naila Murray, and Martin Humenberger. Virtual kitti 2, 2020.

Cesar Cadena, Luca Carlone, Henry Carrillo, Yasir Latif, Davide Scaramuzza, José Neira, Ian Reid, and John J. Leonard. Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Transactions on Robotics*, 32(6):1309–1332, 2016. doi: 10.1109/TRO.2016.2624754.

Carlos Campos, Richard Elvira, Juan J. Gómez Rodríguez, José M. M. 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. doi: 10.1109/TRO.2021.3075644.

Jiarui Dou, Xianlei Long, Huidong Liu, Hui Zhao, Fangming Guo, Boshan Liu, Chao Chen, and Fuqiang Gu. Linear dual-branch spatiotemporal reasoning network for end-to-end visual odometry. *IEEE Transactions on Circuits and Systems for Video Technology*, pp. 1–1, 2026. doi: 10.1109/TCSVT.2026.3726441.

David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Weinberger (eds.), *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014.

Jakob Engel, Vladlen Koltun, and Daniel Cremers. Direct sparse odometry. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(3):611–625, 2018. doi: 10.1109/TPAMI.2017.2658577.

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.

Clement Godard, Oisin Mac Aodha, Michael Firman, and Gabriel Brostow. Digging into self-supervised monocular depth estimation. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 3827–3837, 2019. doi: 10.1109/ICCV.2019.00393.

Zijie Jiang, Hajime Taira, Naoyuki Miyashita, and Masatoshi Okutomi. Self-supervised ego-motion estimation based on multi-layer fusion of rgb and inferred depth. In *2022 International Conference on Robotics and Automation (ICRA)*, pp. 7605–7611, 2022. doi: 10.1109/ICRA46639.2022.9811842.

Yanlin Jin, Rui-Yang Ju, Haojun Liu, and Yuzhong Zhong. Orb-sfmlearner: Orb-guided self-supervised visual odometry with selective online adaptation. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1046–1052, 2025. doi: 10.1109/ICRA55743.2025.11127848.

Bin Li, Shuling Wang, Haifeng Ye, Xiaojin Gong, and Zhiyu Xiang. Cross-modal knowledge distillation for depth privileged monocular visual odometry. *IEEE Robotics and Automation Letters*, 7 (3):6171–6178, 2022. doi: 10.1109/LRA.2022.3166457.

Shuaiyong Li and Zhongfei He. A novel visual odometry network based on short sequence images with local constraint loss function. *IEEE Transactions on Instrumentation and Measurement*, 75: 1–12, 2026. doi: 10.1109/TIM.2026.3660434.

Xiangtai Li, Xia Li, Li Zhang, Guangliang Cheng, Jianping Shi, Zhouchen Lin, Shaohua Tan, and Yunhai Tong. Improving semantic segmentation via decoupled body and edge supervision. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm (eds.), *Computer Vision – ECCV 2020*, pp. 435–452, Cham, 2020. Springer International Publishing. ISBN 978-3-030-58520-4.

Yiling Liu, Hesheng Wang, Jingchuan Wang, and Xinlei Wang. Unsupervised monocular visual odometry based on confidence evaluation. *IEEE Transactions on Intelligent Transportation Systems*, 23(6):5387–5396, 2022. doi: 10.1109/TITS.2021.3053412.

Raúl 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. doi: 10.1109/TRO.2017.2705103.

Julio A. Placed, Jared Strader, Henry Carrillo, Nikolay Atanasov, Vadim Indelman, Luca Carlone, and José A. Castellanos. A survey on active simultaneous localization and mapping: State of the art and new frontiers. *IEEE Transactions on Robotics*, 39(3):1686–1705, 2023. doi: 10.1109/TRO.2023.3248510.

Rujun Song, Jiaqi Liu, Zhuoling Xiao, and Bo Yan. Graphavo: Self-supervised visual odometry based on graph-assisted geometric consistency. *IEEE Transactions on Intelligent Transportation Systems*, 25(12):20673–20682, 2024. doi: 10.1109/TITS.2024.3462596.

Rujun Song, Ruoqi Li, Zhuoling Xiao, and Bo Yan. Swinfvo: Self-supervised visual odometry with enhanced global spatiotemporal perception. *IEEE Transactions on Circuits and Systems for Video Technology*, pp. 1–1, 2025. doi: 10.1109/TCSVT.2025.3637304.

Qiyu Sun, Yang Tang, Chongzhen Zhang, Chaoqiang Zhao, Feng Qian, and Jürgen Kurths. Unsupervised estimation of monocular depth and vo in dynamic environments via hybrid masks. *IEEE Transactions on Neural Networks and Learning Systems*, 33(5):2023–2033, 2022. doi: 10.1109/TNNLS.2021.3100895.

Zachary Teed and Jia Deng. Droid-slam: Deep visual slam for monocular, stereo, and rgb-d cameras. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 16558–16569. Curran Associates, Inc., 2021.

Zachary Teed, Lahav Lipson, and Jia Deng. Deep patch visual odometry. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA, 2023. Curran Associates Inc.

Chengjun Wang, Zhihui Cao, Jingyin Wang, Yonghua Jiang, Mingjun Deng, Lijing Bu, and Guohui Deng. Hedge-mamyolo: High-frequency edge information-facilitated cross-domain feature interaction mamba for drone images small-object detection. *IEEE Transactions on Geoscience and Remote Sensing*, 64:1–16, 2026. doi: 10.1109/TGRS.2026.3686228.

Sen Wang, Ronald Clark, Hongkai Wen, and Niki Trigoni. Deepvo: Towards end-to-end visual odometry with deep recurrent convolutional neural networks. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2043–2050, 2017. doi: 10.1109/ICRA.2017.7989236.

Wenshan Wang, Yaoyu Hu, and Sebastian Scherer. Tartanvo: A generalizable learning-based vo. In Jens Kober, Fabio Ramos, and Claire Tomlin (eds.), *Proceedings of the 2020 Conference on Robot Learning*, volume 155 of *Proceedings of Machine Learning Research*, pp. 1761–1772. PMLR, 16–18 Nov 2021.

Wenhui Wei, Jiantao Li, Kaizhu Huang, Jiadong Li, Xin Liu, and Yangfan Zhou. Lite-svo: Towards a lightweight self-supervised semantic visual odometry exploiting multi-feature sharing architecture. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 10305–10311, 2024a. doi: 10.1109/ICRA57147.2024.10610019.

Wenhui Wei, Yang Ping, Jiadong Li, Xin Liu, and Yangfan Zhou. Fine-mvo: Toward fine-grained feature enhancement for self-supervised monocular visual odometry in dynamic environments. *IEEE Transactions on Intelligent Transportation Systems*, 25(10):13947–13960, 2024b. doi: 10.1109/TITS.2024.3404924.

Zhigang Wu and Yaohui Zhu. Swformer-vo: A monocular visual odometry model based on swin transformer. *IEEE Robotics and Automation Letters*, 9(5):4766–4773, 2024. doi: 10.1109/LRA.2024.3384911.

Chang Xu, Taiping Zeng, Yifan Luo, Fei Song, and Bailu Si. Spatiotemporal dual-stream network for visual odometry. *IEEE Robotics and Automation Letters*, 10(4):3867–3874, 2025. doi: 10.1109/LRA.2025.3544521.

Huangying Zhan, Ravi Garg, Chamara Saroj Weerasekera, Kejie Li, Harsh Agarwal, and Ian M. Reid. Unsupervised learning of monocular depth estimation and visual odometry with deep feature reconstruction. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 340–349, 2018. doi: 10.1109/CVPR.2018.00043.

Huangying Zhan, Chamara Saroj Weerasekera, Jia-Wang Bian, and Ian Reid. Visual odometry revisited: What should be learnt? In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4203–4210, 2020. doi: 10.1109/ICRA40945.2020.9197374.

He Zhang and Cang Ye. Sampson distance: A new approach to improving visual-inertial odometry's accuracy. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 9184–9189, 2021. doi: 10.1109/IROS51168.2021.9636115.

Ning Zhang, Francesco Nex, George Vosselman, and Norman Kerle. Lite-mono: A lightweight cnn and transformer architecture for self-supervised monocular depth estimation. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18537–18546, 2023. doi: 10.1109/CVPR52729.2023.01778.

Hui Zhao, Jianga Shang, Kai Liu, Chao Chen, and Fuqiang Gu. Edgevo: An efficient and accurate edge-based visual odometry. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 10630–10636, 2023. doi: 10.1109/ICRA48891.2023.10160754.

Hui Zhao, Fuqiang Gu, Jianga Shang, Xianlei Long, Jiarui Dou, Chao Chen, Huayan Pu, and Jun Luo. Toward accurate, efficient, and robust rgb-d simultaneous localization and mapping in challenging environments. *IEEE Transactions on Robotics*, 41:5720–5739, 2025. doi: 10.1109/TRO.2025.3610173.

Tinghui Zhou, Matthew Brown, Noah Snavely, and David G. Lowe. Unsupervised learning of depth and ego-motion from video. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6612–6619, 2017. doi: 10.1109/CVPR.2017.700.

A APPENDIX

A.1 OVERVIEW

This appendix provides supplementary methodological details and experimental analyses. It describes the self-supervised training objective (Appendix A.2), essential-matrix pose recovery for SPGR (Appendix A.3), translation-direction reliability checks (Appendix A.4), and the MRG optimization and trajectory updates (Appendix A.5). It also presents implementation and evaluation details (Appendix A.6), additional KITTI results under the Eigen-split training protocol Eigen et al. (2014) (Appendix A.7), SPGR reliability-gate ablations (Appendix A.8), and the inference-time measurement protocol and component breakdown (Appendix A.9). Each detailed section is explicitly referenced in the main paper.

A.2 SELF-SUPERVISED TRAINING VIA VIEW SYNTHESIS

Following established self-supervised VO methods Zhou et al. (2017); Bian et al. (2019), GeoStruct-VO jointly trains the depth and pose networks using differentiable view synthesis, without requiring ground-truth depth or camera poses. Given a target image I_t and adjacent source images I_s, the depth network estimates the corresponding depth maps D_t and D_s, while the pose network estimates the relative transformation $T_{t\to s} = [R_{t\to s} \mid t_{t\to s}]$ from the target camera to each source camera. These predictions are used to synthesize the target view from the source images. The complete loss function $\mathcal{L}$ consists of an image reconstruction loss $\mathcal{L}_{\text{rec}}$, a geometry consistency loss $\mathcal{L}_{\text{geo}}$, an edge-aware smoothness loss $\mathcal{L}_{\text{sm}}$, and a channel regularization loss $\mathcal{L}_{\text{reg}}$:

$$\mathcal{L} = \mathcal{L}_{\text{rec}} + \lambda_{\text{geo}}\mathcal{L}_{\text{geo}} + \lambda_{\text{sm}}\mathcal{L}_{\text{sm}} + \lambda_{\text{reg}}\mathcal{L}_{\text{reg}}, \tag{15}$$

where $\lambda_{\text{geo}} = 10^{-2}$, $\lambda_{\text{sm}} = 10^{-3}$, and $\lambda_{\text{reg}} = 2 \times 10^{-5}$.

Image reconstruction loss. A target pixel p with homogeneous coordinate $\widetilde{p}$ is projected onto the source view as

$$\widetilde{p}_s \sim KR_{t\to s}\left(D_t(p)K^{-1}\widetilde{p}\right) + Kt_{t\to s}, \qquad \widehat{I}_{s\to t}(p) = I_s(\pi(\widetilde{p}_s))\,, \tag{16}$$

where K is the camera intrinsic matrix, $\pi(\cdot)$ performs perspective division, and $\widehat{I}_{s\to t}$ is obtained using bilinear sampling. The photometric discrepancy between the target and synthesized images is

$$\mathcal{L}\left(I_t, \widehat{I}_t\right) = \alpha\,\frac{1 - \text{SSIM}\left(I_t, \widehat{I}_{s\to t}\right)}{2} + (1-\alpha)\left\|I_t - \widehat{I}_{s\to t}\right\|_1. \tag{17}$$

where $\alpha = 0.85$. For the two adjacent source frames, the smaller reprojection error is retained at each target pixel:

$$\mathcal{L}\left(I_t, \widehat{I}_t\right) = \min_{s\in\{t-1,t+1\}} \mathcal{L}\left(I_t, \widehat{I}_t\right). \tag{18}$$

Following Godard et al. (2019), an auto-mask removes pixels for which view synthesis does not improve over the unwarped source image:

$$\mu = \left[\min_{s\in\{t-1,t+1\}} \mathcal{L}\left(I_t, I_s\right) < \min_{s\in\{t-1,t+1\}} \mathcal{L}\left(I_t, \widehat{I}_t\right)\right]. \tag{19}$$

Here, $[\cdot]$ equals one when the condition is true and zero otherwise. Therefore, the image reconstruction loss is defined as:

$$\mathcal{L}_{rec}\left(I_t, I_s\right) = \mu \cdot \mathcal{L}\left(I_t, \widehat{I}_t\right), \tag{20}$$

Geometry consistency loss. The geometry consistency loss encourages the consistency of depth predictions across adjacent views. Specifically, the depth of a target point after transformation to the source camera is

$$D_{t\to s}(p) = \left[R_{t\to s}\left(D_t(p)K^{-1}\widetilde{p}\right) + t_{t\to s}\right]_z, \tag{21}$$

where $[\cdot]_z$ denotes the third coordinate of the transformed 3D point. The predicted source depth is sampled at the corresponding source location:

$$\widehat{D}_s(p) = D_s(\pi(\widetilde{p}_s))\,. \tag{22}$$

The normalized depth inconsistency between the transformed target depth and the sampled source depth is defined as

$$\Delta_s(p) = \frac{|D_{t\to s}(p) - \widehat{D}_s(p)|}{D_{t\to s}(p) + \widehat{D}_s(p) + \epsilon}. \tag{23}$$

Following the minimum reprojection strategy, the smaller inconsistency from the two adjacent source frames is retained. The geometry consistency loss is

$$\mathcal{L}_{\text{geo}} = \frac{1}{|\mathcal{V}|} \sum_{p\in\mathcal{V}} \min_{s\in\{t-1,t+1\}} \Delta_s(p), \tag{24}$$

where $\mathcal{V}$ denotes the set of pixels with valid source-view projections.

Edge-aware smoothness loss. The edge-aware smoothness loss encourages locally smooth depth predictions while preserving depth discontinuities near image boundaries. Let $d_t = 1/(D_t + \epsilon)$ denote the inverse depth and $\bar{d}_t = d_t/(\text{mean}(d_t) + \epsilon)$ its mean-normalized form. The smoothness loss is defined as

$$\mathcal{L}_{\text{sm}} = \frac{1}{N} \sum_p \left(|\partial_x \bar{d}_t(p)| e^{-|\partial_x I_t(p)|} + |\partial_y \bar{d}_t(p)| e^{-|\partial_y I_t(p)|} \right), \tag{25}$$

where N is the number of pixels, and ∂_x and ∂_y denote spatial gradients along the horizontal and vertical directions, respectively. The image-gradient weights reduce the smoothness penalty at object boundaries, allowing sharp depth discontinuities to be retained.

Channel regularization loss. The channel-exchange operation uses the scaling factors of batch-normalization layers to identify less informative channels. Directly minimizing these scaling factors, however, may produce a trivial solution in which all exchangeable channels approach zero. Therefore, the channel regularization loss encourages channel sparsity while maintaining a clear separation between informative and uninformative channels:

$$\mathcal{L}_{\text{reg}} = \sum_m \sum_l \sum_i \left(|\gamma_{m,l,i}| - \eta |\gamma_{m,l,i} - \bar{\gamma}_l| \right), \tag{26}$$

where $\gamma_{m,l,i}$ denotes the scaling factor of channel i in layer l for modality m, and $\bar{\gamma}_l$ is the mean of the exchangeable scaling factors in layer l. The first term suppresses less informative channels, while the second term encourages the scaling factors to separate from their mean and prevents all channels from collapsing to zero. We set $\eta = 10^{-1}$.

A.3 Essential-Matrix Pose Recovery for SPGR

This stage provides SPGR with a geometry-based relative rotation and translation direction. Given the camera intrinsic matrix K and selected 2D–2D correspondences between the target and source images, we estimate the essential matrix and decompose it into a geometric pose (R^g, d^g). Here, R^g is the relative rotation and d^g is the unit translation direction. The translation magnitude is not observable from two-view geometry and is therefore provided by the pose network.

Essential matrix estimation. Let $\mathcal{M} = \{(p_i^t, p_i^s)\}_{i=1}^N$ denote the selected correspondences. The matched pixels are converted into normalized camera coordinates:

$$x_i^t = K^{-1}\widetilde{p}_i^t, \qquad x_i^s = K^{-1}\widetilde{p}_i^s, \tag{27}$$

where $\widetilde{p}_i^t$ and $\widetilde{p}_i^s$ are homogeneous pixel coordinates.

Let z_i^t and z_i^s denote the depths of the corresponding 3D point in the target and source cameras. Under the target-to-source relative pose (R, t), the two viewing rays satisfy

$$z_i^s x_i^s = R\left(z_i^t x_i^t\right) + t. \tag{28}$$

Multiplying this equation from the left by $(x_i^s)^\top [t]_\times$ eliminates the unknown depths because $(x_i^s)^\top [t]_\times x_i^s = 0$ and $[t]_\times t = 0$. This gives the epipolar constraint

$$(x_i^s)^\top [t]_\times R x_i^t = 0. \tag{29}$$

Therefore, the essential matrix and the relative pose are related by

$$E = [t]_\times R. \tag{30}$$

With $t = sd$ 7, we have $E = s[d]_\times R$. Thus, the essential matrix determines the rotation and translation direction, but not the translation magnitude.

Pose recovery from the essential matrix. For each valid essential-matrix solution, we apply singular value decomposition:

$$E = U\Sigma V^\top. \tag{31}$$

An ideal essential matrix has two equal nonzero singular values and one zero singular value. Since the overall scale of the essential matrix is irrelevant, we use the normalized form $\Sigma = \mathrm{diag}(1, 1, 0)$. The signs of the singular vectors are adjusted when necessary so that the recovered rotation candidates belong to $\mathrm{SO}(3)$. Because $E = [d]_\times R$ implies $E^\top d = 0$, the translation direction is given by the third column u_3 of U, up to sign. Let

$$W = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{32}$$

The rotation and translation-direction candidates are

$$R_1 = UWV^\top, \qquad R_2 = UW^\top V^\top, \tag{33}$$

$$d_1 = u_3, \qquad d_2 = -u_3. \tag{34}$$

Their combinations produce four pose hypotheses: (R_1, d_1), (R_1, d_2), (R_2, d_1), and (R_2, d_2).

The RANSAC inliers are triangulated under each hypothesis, and the solution producing the largest number of points with positive depth in both views is retained. If multiple essential-matrix solutions are obtained, the same criterion is used to select the best one. The resulting geometric pose is denoted by

$$(R^g, d^g), \qquad \|d^g\|_2 = 1. \tag{35}$$

SPGR retains the magnitude of the network translation:

$$s^n = \|t^n\|_2, \qquad T^c_{t\to s} = [R^g \mid s^n d^g]. \tag{36}$$

The candidate pose is used only when the geometric estimate passes the reliability checks; otherwise, SPGR falls back to the original network pose. Thus, geometry provides the rotation and translation direction, while the pose network provides the translation magnitude.

A.4 Reliability of Essential-Matrix Translation

An essential-matrix estimate can satisfy the geometric checks while yielding an unreliable translation direction. We therefore assess its agreement with the network prediction and the temporal consistency of the resulting correction.

Let t^n_i be the network translation and d^g_i the unit direction recovered from the essential matrix. With $\hat{t}^n_i = t^n_i / \|t^n_i\|_2$, define $c_i = (\hat{t}^n_i)^\top d^g_i$. This is the cosine of the angle between the two directions; candidates with $c_i < 0$ are rejected. For the remaining candidates, the rotation vector aligning the network direction with d^g_i is

$$v_i = \frac{\arccos(c_i)}{\|\hat{t}^n_i \times d^g_i\|_2}(\hat{t}^n_i \times d^g_i). \tag{37}$$

Let $\mathcal{H}_i$ contain up to ten preceding correction vectors whose estimates passed the geometric checks and satisfied $c_j \geq 0$. We use their componentwise median $\bar{v}_i$ as the temporal reference and define

$$\gamma_2 = \begin{cases} 1, & c_i \geq 0, \quad |\mathcal{H}_i| \geq 2, \quad \|v_i - \bar{v}_i\|_2 \leq \pi/36, \\ 0, & \text{otherwise.} \end{cases} \tag{38}$$

When $\gamma_2 = 1$, SPGR uses the geometric direction with the network-predicted magnitude, $\|t^n_i\|_2 d^g_i$; otherwise, it retains t^n_i.

A.5 Multi-frame Rotation Graph

The rotational residual and optimization objective are defined in Eqs. 13 and 14. Here, we give the explicit normal equations and describe how the optimized corrections are used to reconstruct the trajectory.

The correction matrix C has size $N \times 3$, with its i-th row given by $c_i^\top$. Accordingly, the i-th row of the solution $C^\star$ contains the optimized correction $(c_i^\star)^\top$. Taking the derivative of Eq. 14 with respect to C gives the following linear system:

$$\left(\begin{bmatrix} 2 & -1 & 0 & \cdots & 0 \\ -1 & 2 & -1 & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & -1 & 2 & -1 \\ 0 & \cdots & 0 & -1 & 1 \end{bmatrix} + \sum_{(s,e)\in\mathcal{E}} w_{se}(e_e - e_s)(e_e - e_s)^\top \right) C^\star = \sum_{(s,e)\in\mathcal{E}} w_{se}(e_e - e_s) d_{se}^\top. \tag{39}$$

The tridiagonal matrix is produced by the anchor and temporal-smoothness terms, while each accepted edge contributes a constraint between c_s and c_e.

The optimized rotation of frame i is

$$R_i^\star = \mathrm{Exp}([c_i^\star]_\times) R_i^f. \tag{40}$$

To update the positions consistently, the original displacement between adjacent frames is expressed in the local camera frame:

$$\Delta p_i^{\mathrm{loc}} = (R_i^f)^\top (p_{i+1}^f - p_i^f). \tag{41}$$

The corrected positions are then reconstructed as

$$p_0^\star = p_0^f, \qquad p_{i+1}^\star = p_i^\star + R_i^\star \Delta p_i^{\mathrm{loc}}. \tag{42}$$

Therefore, $(R_i^\star, p_i^\star)$ directly forms the corrected camera-to-world trajectory. If no cross-frame constraint is accepted, Eq. 39 gives $C^\star = 0$, and the locally refined trajectory remains unchanged.

A.6 Implementation and Evaluation Details

GeoStruct-VO is implemented in PyTorch 2.4.0 and trained on a single NVIDIA RTX 4090 D GPU. The experiments use an AMD EPYC 7K62 CPU with 48 cores and 251 GiB of host memory. The software environment consists of Python 3.9.25, CUDA 12.1, cuDNN 9.1.0, TorchVision 0.19.0, and OpenCV 4.6.0, together with NumPy, SciPy, TensorBoardX, and EasyDict. Training uses three-frame samples resized to 192×640, ImageNet-pretrained ResNet-18 encoders, the Adam optimizer, a batch size of 12, and 40 epochs. The learning rate alternates between 10^{-4} and 5×10^{-5} every ten epochs. All reported variants use identical data preprocessing, training schedules, checkpoint-selection rules, and trajectory evaluators.

At inference time, SPGR detects 500 corners on KITTI and 1,000 on vKITTI2 at an image scale of 0.8 and tracks them using pyramidal Lucas–Kanade optical flow. The target image is partitioned into a 4×8 grid, with at most $K = 50$ valid tracks retained per cell. We set $N_{\min} = 50$, $\tau_{fb} = \tau_p = 0.5$, $\tau_i = \tau_c = 0.05$, $\tau_R = 20^\circ$, $\tau_t = -0.1$, and $\tau_e = 0.30$. RANSAC uses a threshold of 0.5, a confidence of 0.99, at most 25 iterations, and a minimum inlier ratio of 0.70. MRG uses temporal offsets of three and five frames and rejects non-finite cross-frame residuals or residuals larger than 10°. The full GeoStruct-VO model applies MRG, whereas GeoStruct-VO† denotes the real-time configuration without MRG.

A.7 Additional KITTI Results

The main experiments use KITTI odometry sequences 00–08 for training and sequences 09–10 for testing. For an additional comparison with prior monocular VO methods, we separately train on the KITTI-Raw Eigen split Eigen et al. (2014) and report results on odometry sequences 00–10. The Eigen-split training data overlap odometry sequences 01, 02, 06, 08, 09, and 10 Jin et al. (2025). Consequently, the 11-sequence averages include training-overlapping sequences and are reported

Table 5: Comparison on the KITTI odometry split. The table reports t_{rel}, r_{rel}, and ATE without units in the metric labels. Lower values are better. The average is computed over all 11 sequences (Seq. 00–10). Best and second-best results are shown in bold and underlined, respectively.

Methods	Metrics	Seq. 00	Seq. 01	Seq. 02	Seq. 03	Seq. 04	Seq. 05	Seq. 06	Seq. 07	Seq. 08	Seq. 09	Seq. 10	Avg.
ORB-SLAM2 (w/o loop closure)	t_{rel}	11.43	107.56	10.34	**0.97**	**1.30**	9.04	14.56	9.77	11.46	9.30	2.57	17.12
	r_{rel}	0.58	0.89	**0.26**	**0.19**	**0.27**	**0.26**	0.26	0.36	**0.28**	**0.26**	0.32	**0.36**
	ATE	40.65	502.20	47.82	**0.94**	1.30	29.95	40.82	16.04	43.09	38.77	5.42	69.73
VISO2	t_{rel}	12.66	41.93	9.47	3.93	2.50	15.10	6.80	10.80	14.82	3.69	21.01	12.97
	r_{rel}	2.73	7.68	1.19	2.21	1.78	3.65	1.93	4.67	2.52	1.25	3.26	2.99
	ATE	78.40	127.59	49.96	2.35	1.50	65.73	21.23	17.88	53.57	10.18	46.62	43.18
SfMLearner	t_{rel}	21.32	22.41	24.10	12.56	4.32	12.99	15.55	12.61	10.66	11.32	15.26	14.83
	r_{rel}	6.19	2.79	4.18	4.52	3.28	4.66	5.58	6.31	3.75	4.07	4.06	4.49
	ATE	104.87	109.61	185.43	8.42	3.10	60.89	52.19	20.12	30.97	26.93	24.09	56.96
SC-SFM	t_{rel}	11.01	27.09	6.74	9.22	4.22	6.70	5.36	8.29	8.11	7.64	10.74	9.56
	r_{rel}	3.39	1.31	1.96	4.93	2.01	2.38	1.65	4.53	2.61	2.19	4.58	2.87
	ATE	93.04	85.91	70.37	10.21	2.97	40.56	12.56	21.01	56.15	15.02	20.19	38.91
Monodepth2	t_{rel}	17.65	8.78	11.97	11.88	4.06	8.88	4.86	8.15	11.58	12.42	10.66	10.08
	r_{rel}	3.79	1.16	2.16	5.99	1.49	3.95	1.56	5.14	4.00	2.72	4.53	3.32
	ATE	133.58	32.99	135.47	12.72	1.56	59.40	11.13	18.01	93.74	57.13	18.34	52.19
TartanVO	t_{rel}	9.94	18.87	9.87	7.16	8.08	9.40	9.50	9.78	11.43	10.03	13.41	10.68
	r_{rel}	3.59	1.93	3.37	2.73	4.47	3.24	2.51	4.96	3.17	3.18	3.21	3.31
	ATE	63.45	70.79	67.64	7.72	2.90	54.07	24.62	19.56	59.36	31.97	25.08	38.83
Depth-VO-Feat	t_{rel}	8.45	28.38	7.28	12.55	2.23	4.92	5.82	5.84	5.54	9.06	9.60	9.06
	r_{rel}	2.44	1.75	2.26	10.62	2.02	2.34	2.06	3.56	2.39	3.60	3.41	3.31
	ATE	58.97	105.33	84.82	14.20	1.44	21.64	14.30	15.03	25.04	41.15	13.02	35.90
DF-VO	t_{rel}	2.25	66.98	3.60	2.67	1.44	**1.15**	**1.03**	**0.93**	**2.23**	2.46	**1.96**	7.88
	r_{rel}	0.58	17.04	0.52	0.50	0.29	0.30	**0.26**	**0.29**	0.30	0.30	**0.31**	1.88
	ATE	**12.64**	695.75	23.11	1.23	1.36	**3.75**	**2.63**	**1.74**	**7.87**	11.02	**3.37**	69.50
ORB-SfMLearner (concatenate, with SOA)	t_{rel}	5.43	5.01	4.71	5.26	2.35	3.27	2.91	3.83	5.73	4.11	5.74	4.40
	r_{rel}	1.70	0.66	1.76	2.54	1.34	1.79	1.24	2.45	2.05	1.69	2.50	1.79
	ATE	28.62	22.63	38.27	3.38	1.95	15.65	8.01	8.30	17.84	7.88	9.60	14.74
ORB-SfMLearner (attention, with SOA)	t_{rel}	5.23	**4.47**	4.66	5.76	3.29	4.05	2.97	2.39	4.88	3.85	5.98	4.32
	r_{rel}	1.90	0.71	1.80	2.85	1.95	2.02	1.39	2.24	1.85	1.67	2.44	1.89
	ATE	34.63	**17.09**	43.18	3.74	2.27	15.77	7.32	3.52	17.70	**7.15**	10.50	14.81
GeoStruct-VO	t_{rel}	**1.94**	8.68	**2.44**	2.06	1.35	1.90	1.39	1.58	2.56	**2.24**	3.53	**2.70**
	r_{rel}	**0.51**	**0.64**	0.46	0.41	0.68	0.58	0.30	1.18	0.52	0.39	0.45	0.56
	ATE	15.58	26.45	**17.90**	1.90	**1.10**	7.36	2.98	2.70	10.51	7.66	7.47	**9.24**

as a supplementary comparison rather than strictly held-out test performance. Under this protocol, GeoStruct-VO obtains the lowest average translational error and ATE among the compared methods, with $t_{\mathrm{rel}} = 2.70$ and $\mathrm{ATE} = 9.24$. It also obtains the lowest rotational error among the learning-based methods, reaching $r_{\mathrm{rel}} = 0.56$. Compared with the best ORB-SfMLearner configuration for each metric, GeoStruct-VO reduces the average t_{rel}, r_{rel}, and ATE by 37.5%, 68.7%, and 37.3%, respectively. These results indicate that combining learned motion estimation with selective geometric refinement provides a more favorable balance between translational and rotational accuracy.

Several representative comparisons further explain this behavior. ORB-SLAM2 achieves the lowest average rotational error, with $r_{\mathrm{rel}} = 0.36$, but produces substantially larger translational errors and ATE, reaching 17.12 and 69.73, respectively. Its explicit multi-view constraints and local pose optimization provide strong rotational estimation, whereas monocular scale ambiguity and accumulated scale drift limit its translational accuracy, even without loop closure. In contrast, GeoStruct-VO preserves the translation magnitude predicted by the pose network while replacing the rotation and translation direction with reliable geometric estimates. The SPGR module further evaluates geometric observability and rejects unreliable estimates, which prevents degenerate correspondences from directly propagating to the final pose. Consequently, GeoStruct-VO reduces the average t_{rel} and ATE by 84.2% and 86.7% relative to ORB-SLAM2, while maintaining competitive rotational accuracy.

DF-VO provides another informative comparison. By explicitly using optical flow and geometric constraints, DF-VO performs particularly well on Seq. 06 and Seq. 07, where it achieves t_{rel} values of 1.03% and 0.93%, respectively, and lower ATE than GeoStruct-VO. However, its performance degrades severely on Seq. 01, with $t_{\mathrm{rel}} = 66.98$, $r_{\mathrm{rel}} = 17.04$, and $\mathrm{ATE} = 695.75$. This behavior suggests that the method is sensitive to optical-flow quality and geometric initialization; erroneous correspondences can accumulate into large trajectory drift on challenging sequences. GeoStruct-VO also uses image correspondences, but employs forward–backward validation, spatially balanced selection, RANSAC filtering, positive-depth checks, and a fallback to the learned pose when geometry is unreliable. This reliability-aware design explains its much more stable performance on Seq. 01 and across the complete evaluation set.

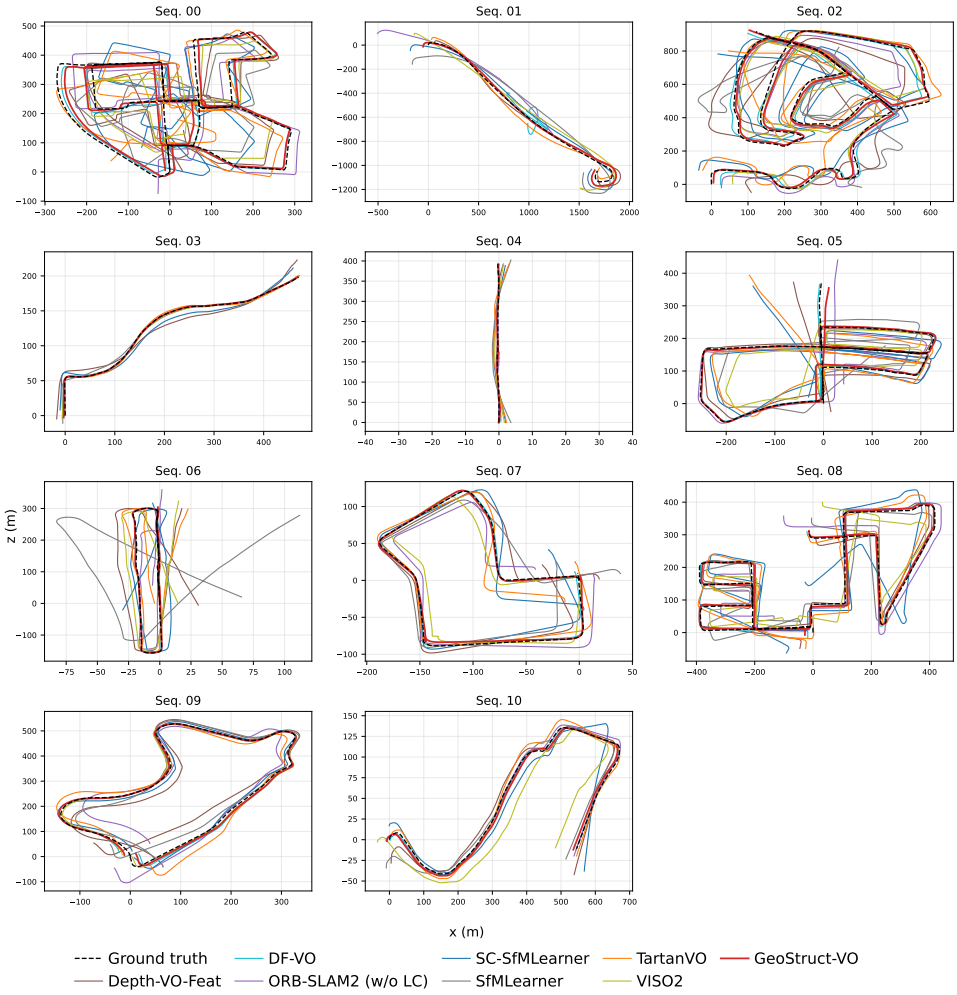


Figure 7: Projected x–z trajectories on KITTI odometry sequences 00–10 under the Eigen-split training protocol. Dashed black curves show ground truth; the remaining curves show GeoStruct-VO and the VO baselines identified in the legend.

The two ORB-SfMLearner variants with SOA achieve significantly better results than the conventional SfMLearner, demonstrating that online parameter adaptation improves cross-sequence generalization. However, SOA requires online model updates and introduces a considerable inference overhead, making it unsuitable for real-time deployment. GeoStruct-VO avoids online parameter updates and instead performs lightweight selective geometric refinement during inference, achieving both higher accuracy and real-time operation. Other monocular learning-based methods, including SfMLearner, SC-SFM, Monodepth2, TartanVO, and Depth-VO-Feat, generally suffer from scale ambiguity, photometric ambiguity, or sensitivity to appearance and motion changes, leading to larger average errors.

The trajectory plots in Fig. 7 provide qualitative support for these observations. The GeoStruct-VO trajectory generally remains close to the dashed ground-truth trajectory and preserves the correct global structure across the evaluated sequences. The agreement is especially clear on Seq. 03, Seq. 04, Seq. 07, Seq. 09, and Seq. 10. On more challenging sequences such as Seq. 00, Seq. 01, Seq. 02, Seq. 05, Seq. 06, and Seq. 08, several competing methods exhibit larger drift, distorted turns, or unstable trajectory shapes, whereas GeoStruct-VO maintains a more consistent alignment with the ground truth. In particular, the strong performance of DF-VO on Seq. 06 and Seq. 07 but its severe drift on Seq. 01 is also visible in the trajectory plots, further highlighting the robustness advantage of the proposed selective geometric correction.

A.8 GEOMETRIC RELIABILITY GATE ABLATION IN SPGR

Experimental setup. We evaluate the reliability gates in SPGR on KITTI 09–10 and vKITTI2 test sequences. The full configuration is compared with three ablations. Ungated uses the recovered geometric pose without reliability-based rejection. The remaining variants disable either γ_1, which checks geometric observability and support, rotation consistency, and epipolar residuals, or γ_2, which checks translation-direction agreement and temporal consistency.



(a) KITTI (b) vKITTI clone (c) vKITTI fog (d) vKITTI morning

Figure 8: Comparison of scenes. Fog reduces the visibility of distant vehicles and road structure, illustrating a challenging condition for correspondence-based geometric pose recovery.

Results and analysis. Table 6 compares the SPGR reliability-gate configurations. The KITTI example in Fig. 8 has clear visibility, providing favorable conditions for feature correspondence and geometric pose recovery. Under such conditions, the geometric estimate can produce accurate poses even without reliability gating: on KITTI 09–10, Ungated achieves a mean translational drift of 2.97% and rotational drift of 0.49°/100 m. Full gated reduces these errors only modestly, to 2.93% and 0.46°/100 m. Consistently, Table 7 shows that γ_1 rejects only 15 of 2,790 KITTI frames, while γ_2 rejects no additional frames. In contrast, the evaluated vKITTI2 sequences include more varied visual conditions. The fog example in Fig. 8 shows how distant scene structure can become obscured, making geometric recovery less reliable. On vKITTI2, Full gated reduces mean translational drift from 5.445% to 4.35% and rotational drift from 1.505 to 1.42°/100 m relative to Ungated, reductions of 20.1% and 5.6%, respectively. Both single-gate ablations have higher errors than the full configuration. Across vKITTI2, γ_1 rejects 269 of 3,846 frames, and γ_2 rejects a further 580. These results indicate that reliability gating is most useful when visual conditions make recovered geometry less dependable.

Table 6: Comparison of SPGR reliability-gate configurations on KITTI 09–10 and vKITTI2.

Dataset	Variant	t_{rel} (%)	r_{rel} (°/100 m)
KITTI 09–10	Ungated	2.97	0.49
	w/o γ_1	2.96	0.48
	w/o γ_2	2.95	0.46
	Full gated	2.93	0.46
vKITTI2	Ungated	5.45	1.51
	w/o γ_1	5.13	1.76
	w/o γ_2	4.88	1.47
	Full gated	4.35	1.42

Table 7: Frame-level activity of the full SPGR reliability gate. "Both pass" counts frames passing γ_1 and γ_2. Additional γ_2 rejects are counted only among frames that first passed γ_1.

Dataset	γ_1 pass	Both pass	γ_1 reject	Additional γ_2 reject
KITTI 09–10	2775/2790 (99.5%)	2775/2790 (99.5%)	15	0
vKITTI2	3577/3846 (93.0%)	2997/3846 (77.9%)	269	580

A.9 INFERENCE TIME AND TIME-COST CALCULATION

We evaluate the optimized inference pipeline on the first 200 frames of KITTI odometry Sequence 00. The first 20 frames are used for warm-up, and the remaining 180 consecutive frame pairs are included in the reported averages. Image loading and preprocessing are excluded, and GPU operations are synchronized before and after execution. The reported values therefore characterize steady-state inference after initialization.

Detailed Runtime Analysis. The timing measurements are divided into isolated diagnostics and online wall-clock measurements. For the isolated measurements, the geometry worker is disabled

and depth and pose inference are timed separately on the GPU; their sum characterizes the neural-network workload, but it is not the latency of an online frame update. During normal inference, the network branch and the geometry branch run concurrently: the former performs depth and pose prediction, while the latter performs feature extraction, adjacent-frame tracking, and essential-matrix recovery in a CPU process. The rows labeled "Network lane" and "Geometry lane" are therefore the quantities that determine the front-end latency; the CPU component rows only describe the internal cost of the geometry lane. Let τ_{net} and τ_{geo} denote the measured wall times of the two concurrent lanes, and let τ_{join} denote the time required after the critical lane finishes for inter-process communication, synchronization, branch joining, and SPGR fusion. Since the two lanes overlap, the front-end latency is modeled as

$$\tau_{\mathrm{front}} \approx \max\left(\tau_{\mathrm{net}}, \tau_{\mathrm{geo}}\right) + \tau_{\mathrm{join}}. \tag{43}$$

This expression makes clear that the lane times are not added serially. The corresponding measurements are reported in Table 8.

Table 8: Detailed runtime breakdown on KITTI Sequence 00. Latencies are averaged over 180 measured frame pairs. Isolated rows characterize the network workload, whereas concurrent wall-clock rows determine the online front-end latency. The asterisk indicates that the logged essential-matrix recovery time includes the separately reported SPGR-fusion measurement, so the diagnostic component rows are not an additive decomposition.

Component	Measurement	Latency (ms)
Depth inference	Isolated GPU	3.677
Pose inference	Isolated GPU	10.341
Network total	Isolated GPU	14.018
Feature extraction	CPU component	5.309
Adjacent optical flow	CPU component	8.402
Essential-matrix recovery*	CPU component	3.491
Network lane	Concurrent wall time	16.813
Geometry lane	Concurrent wall time	16.869
SPGR fusion	Join operation	0.336
GeoStruct-VO†	Front-end wall time	20.916
Pose graph back end	Amortized per-frame time	10.500
GeoStruct-VO	Full amortized time	31.416

The isolated network total is 14.018 ms, equal to the sum of the depth and pose measurements ($3.677 + 10.341$ ms), but the active geometry process changes the realized network-lane time to 16.813 ms. With both branches active, the geometry lane is the critical path at 16.869 ms, and the measured front-end latency is 20.916 ms. Equivalently, $\max(16.813, 16.869) + 4.047 = 20.916$ ms. Consequently, the post-lane join cost is $20.916 - 16.869 = 4.047$ ms, of which 0.336 ms is the logged SPGR-fusion operation and approximately 3.711 ms is communication and synchronization. This gives a front-end rate of about 47.8 Hz for GeoStruct-VO†, the real-time configuration without MRG. Optical-flow tracking is the largest listed CPU component (8.402 ms), followed by feature extraction (5.309 ms). The concurrent depth and pose timings (5.697 and 10.789 ms, respectively) are diagnostic overlap measurements; neither these values nor the isolated component rows should be added to the measured front-end wall time.

Back-end amortization. The full GeoStruct-VO configuration additionally runs the pose-graph back end, including MRG refinement. We report its joint amortized cost as $\tau_{\mathrm{pg}} = 10.500$ ms/frame rather than splitting the back end into internal stages. The total amortized cost is therefore

$$\tau_{\mathrm{full}} = \tau_{\mathrm{front}} + \tau_{\mathrm{pg}} = 20.916 + 10.500 = 31.416\ \mathrm{ms/frame}. \tag{44}$$

This corresponds to approximately 31.8 Hz when the back-end cost is amortized over frames, with the pose graph accounting for about 33.4% of the total budget. Because the pose graph runs asynchronously, it does not block the adjacent-frame estimate: GeoStruct-VO† is reported using the 20.916 ms front-end latency, whereas GeoStruct-VO additionally reports the 10.500 ms/frame back-end cost.