IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

Robust Monocular Visual Odometry Via Dual-Paradigm Curriculum Learning

Assaf Lahiany[1] and Oren Gal[1]

***Abstract*—Monocular visual odometry (VO) is accurate in controlled settings yet drifts sharply under aggressive motion and sensor noise. We offer a fundamental rethinking of VO robustness as a training-schedule problem rather than an architectural challenge, introducing a novel dual-paradigm curriculum learning framework that operates at both trajectory and loss-component levels. (i) A motion-based curriculum orders trajectories by measured motion complexity. (ii) A hierarchical component curriculum adaptively re-weights optical-flow, pose, and rotation losses via Self-Paced and in-training Reinforcement Learning (RL) schedulers. Integrated into an unmodified DPVO baseline, these strategies cut TartanAir ATE by 33% with only 31% extra training wall-time, and reach baseline accuracy 47% faster (Self-Paced). Without fine-tuning, the same models improve zero-shot performance on EuRoC (13% ATE reduction), TUM-RGBD (9%; 46% on dynamic scenes), KITTI (21%), and ICL-NUIM (32%). We show that explicit difficulty progression or adaptive loss weighting provides a practical, zero-inference-overhead path to robust monocular VO and could extend to other geometric vision tasks.**

***Index Terms*—Localization, Mapping, SLAM, Autonomous Vehicle Navigation**

I. Introduction

LEARNING-based monocular visual odometry (VO) promises flexible, low-cost egomotion estimation but still fails under fast 6-DoF motion, where large inter-frame displacements overwhelm feature tracking and trigger drift. Prior work tackles the symptom by adding sensors [1] or global optimisation [2], increasing complexity and inference cost.

We contend that *how* VO models are trained leaves substantial room for improvement: random sampling over-represents easy, low-motion frames and leaves networks unprepared for aggressive manoeuvres.

Our results show that *training schedule alone* can significantly reduce the robustness gap, without modifying the baseline network architecture. We introduce two curricula that wrap an unmodified VO backbone:

1) **Motion-based ordering:** A baseline curriculum that validates motion complexity as a difficulty metric and establishes that even simple trajectory ordering improves VO robustness.
2) **Hierarchical component weighting:** Enhanced hierarchical scheduling strategies that adapt optical-flow, translation, and rotation losses online via self-paced progression or reactive Reinforcement Learning (RL) agents.

Across TartanAir [3] established monocular test sequences and four diverse benchmarks (EuRoC [4], TUM-RGBD [5], KITTI [6], and ICL-NUIM [7]), our curricula cut average trajectory error by up to $33\,\%$, reach baseline accuracy $47\,\%$ faster, generalize zero-shot to unseen environments, and incur *no* inference overhead.

We detail our curriculum methodology (Section III), evaluated on established monocular VO datasets (Section IV).

II. Background and Related Work

Learning-based monocular VO inherits two long-studied challenges: (i) aggressive 6-DoF motion that violates small-baseline assumptions behind feature tracking, and (ii) training regimes that over-fit to easy, low-motion data. Classical geometric pipelines such as ORB-SLAM3 [8] and DSO [9] break under fast motion; hybrid systems (e.g., DROID-SLAM [2]) add global optimisation at considerable computational cost. Deep VO methods evolved from supervised approaches (DeepVO [10]) to self-supervised frameworks (UnDeepVO [11], DPVO [12]) that learn robust features end-to-end, yet still drift on high-velocity trajectories and struggle with domain transfer.

Curriculum learning (CL) [13] structures data exposure to improve robustness, evolving from manual design to self-paced methods [14] and automatic discovery [15]. Applied to vision, CL has improved depth estimation [16], NeRF [17], and stereo [18].

To our knowledge, the primary CL application to VO is GA-CL-VO [19], which applies synthetic geometric augmentations with increasing transformation complexity. Our approach fundamentally differs in two ways: (i) we rank real trajectories by actual 6-DoF motion dynamics rather than artificial perturbations, enabling realistic difficulty progression, and (ii) we introduce adaptive loss weighting via a self-paced scheduler and reinforcement-learning agents that respond to training dynamics rather than following fixed curricula. While GA-CL-VO requires custom CNN architectures, our framework operates as an architecture-agnostic wrapper that generalizes across diverse environments—from synthetic to real-world, aerial to handheld—without dataset-specific tuning.

III. Methodology

We propose two complementary curricula paradigms that operate as light wrappers around a state-of-the-art monoc-

Manuscript received: August, 17, 2025; Revised October, 26, 2025; Accepted November, 10, 2025.

This paper was recommended for publication by Editor Sven Behnke upon evaluation of the Associate Editor and Reviewers' comments. This work was supported by the Maurice Hatter Foundation.

[1]Assaf Lahiany and Oren Gal are with the Swarm & AI Lab (SAIL), University of Haifa, Hatter Department of Marine Technologies, Leon H. Charney School of Marine Sciences. alahiany@campus.haifa.ac.il, orengal@univ.haifa.ac.il

Digital Object Identifier (DOI): see top of this page.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

ular VO architecture, DPVO [12], no architecture changes required—thus could generalize well to other learning-based VO methods.

A. Paradigm 1: Motion-Based Curriculum

We first establish a baseline curriculum to validate two hypotheses: (i) motion complexity correlates with VO difficulty, and (ii) simple trajectory ordering alone improves robustness. This baseline ranks trajectories according to their motion complexity and progressively exposes the model to more challenging sequences during training. Motion complexity scores are pre-computed in a single preprocessing pass, creating a deterministic training schedule that systematically introduces increasingly difficult motion patterns.

1) Motion-Complexity Metric and Validation: We quantify motion complexity for each sequence by analyzing frame-to-frame pose differences. For each consecutive pose pair (T_i, T_{i+1}) where $T_i = [\mathbf{R}_i|\mathbf{t}_i] \in SE(3)$, we compute:

$$d_{trans}(i) = \|\mathbf{t}_{i+1} - \mathbf{t}_i\|_2 \tag{1}$$

$$d_{rot}(i) = \left\|\log_{SO(3)}\left(\mathbf{R}_i^{-1}\mathbf{R}_{i+1}\right)\right\|_2 \tag{2}$$

where $\mathbf{t}_i$ and $\mathbf{R}_i$ represent the translation vector and rotation matrix at frame i, and $\log_{SO(3)}$ is the logarithmic map that converts relative rotation to axis-angle representation for magnitude computation. The sequence difficulty score is defined as a convex combination of the mean translational and rotational motion magnitudes:

$$\text{Difficulty}_{seq} = \alpha \cdot \text{mean}(d_{trans}) + (1-\alpha) \cdot \text{mean}(d_{rot}) \tag{3}$$

We evaluated motion complexity as a difficulty indicator by examining its correlation with Average Trajectory Error (ATE) across the complete TartanAir training set, comparing it against optical-flow magnitude. With normalized weighting ($\alpha = 0.5$), motion difficulty demonstrates a strong statistical relationship with ATE ($r = 0.459$, $p < 0.001$), whereas optical-flow magnitude shows negligible correlation ($r = 0.064$, $p = 0.729$), as illustrated in Fig. 1 (left). This analysis establishes motion complexity as our primary curriculum difficulty measure.

Using equal weighting ($\alpha = 0.5$) between translation and rotation in metric (3) produces a right-skewed distribution of motion complexity scores (Fig. 1, right). We partition this distribution at 0.20 and 0.33 to create three tiers with equal training frames each, enabling smooth progression from gentle to aggressive motion dynamics (Table I).

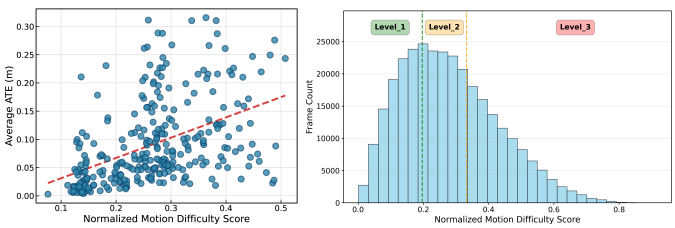

Fig. 1. Motion complexity metric validation. Left: Correlation with ATE ($r = 0.459$, $p < 0.001$). Right: Distribution across 369 TartanAir sequences; dashed lines show tier boundaries (0.20, 0.33).

TABLE I
DIFFICULTY LEVELS AND CORRESPONDING MOTION CHARACTERISTICS FOR TRAJECTORY-BASED CURRICULUM.

Level	Difficulty	Range	Included Motion Characteristics
1	Easy	< 0.20	Gentle translation, min. rotation
2	Medium	0.20–0.33	Moderate velocities, turns
3	Hard	> 0.33	Rapid motion, aggressive turns

2) Trajectory-Based Curriculum Schedule: We implement a tiered progression [13] through three phases: foundation (Level 1), integration (Levels 1–2), and mastery (all levels), advancing once validation ATE stops improving so the model consolidates learning before seeing harder trajectories. Difficulty scores are computed offline, so inference remains as fast as baseline DPVO.

This schedule relies on ground-truth motion scores; when unavailable, lightweight proxies (teacher VO confidence, IMU magnitude/variance, flow variance) can approximate ordering. If proxies are unreliable, we default to the hierarchical loss-based curricula (Paradigm 2), which need no trajectory labels and consistently deliver strong performance (Self-Paced or RL schedulers), making Trajectory-Based an optional add-on when GT is available.

B. Paradigm 2: Hierarchical Component Curriculum

Our second paradigm addresses a key limitation: fixed loss weights cannot adapt to varying motion difficulties. During easy motion, optical flow provides reliable gradients; under aggressive motion, pose losses become more informative. We decompose learning into hierarchical components for multi-granular curriculum control:

$$\mathcal{L}_{pose} = \mathcal{L}_{translation} + w_r \mathcal{L}_{rotation} \tag{4}$$

$$\mathcal{L}_{total} = w_p s_p \mathcal{L}_{pose} + w_f s_f \mathcal{L}_{flow} \tag{5}$$

This hierarchical structure incorporates two weight types: *Base weights* (s_f, s_p) maintain fundamental scaling between flow and pose components, preserving DPVO's empirically optimized balance with $s_f = 0.1$ and $s_p = 10$.
Curriculum weights (w_f, w_p, w_r) provide dynamic learning control, allowing the network to adaptively emphasize different loss components as training progresses.

Figure 2 illustrates our hierarchical component curriculum architecture. The baseline DPVO model generates three loss components—optical flow, translation, and rotation—each scaled by fixed factors (s_f, s_p) and modulated by adaptive curriculum weights (w_f, w_p, w_r). A dynamic weight scheduler adjusts these curriculum weights based on individual loss feedback, while the scale factors preserve DPVO's baseline balance. The hierarchical structure maintains natural dependencies between translation and rotation (both controlled by w_p) while allowing independent rotation weighting via w_r, enabling granular control over each component's contribution to learning.

We instantiate this framework with two dynamic weight schedulers:

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

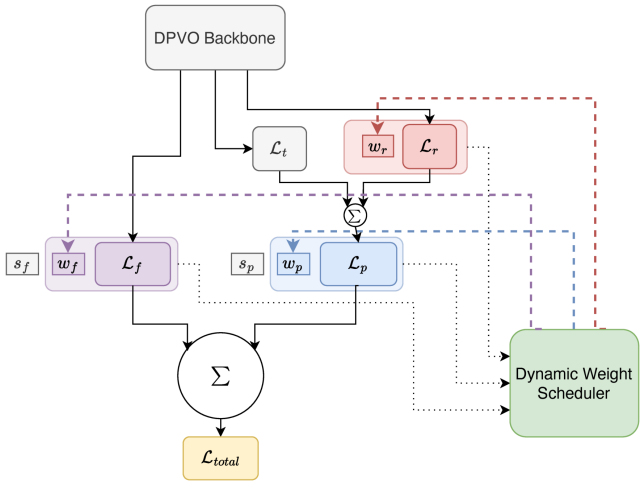

Fig. 2. Architecture of the hierarchical component curriculum (Paradigm 2). The Dynamic Weight Scheduler (Self-Paced or RL-DDPG) adjusts dynamic curriculum weights (w_f, w_p, w_r) based on individual loss feedback.

1) Self-Paced Scheduler: The Self-Paced scheduler maps component losses to curriculum weights through a progression factor ϕ that must be designed for the task. We chose inverse exponential progression to match training dynamics: early training naturally exhibits high losses, where $\phi(\mathcal{L}_i) \approx 0$ keeps weights near $w_0 = 0.1$ for stable initial learning. As training progresses and losses decrease, $\phi(\mathcal{L}_i) \to 1$ gradually increases weights toward $w_F = 1$, allowing full curriculum engagement once the network has stabilized. This temporal alignment between loss evolution and curriculum progression is formulated as:

$$w_i^{f,p,r} = w_0 + (w_F - w_0)\phi(\mathcal{L}_i) \tag{6}$$

$$\phi(\mathcal{L}_i) = e^{-\lambda \mathcal{L}_i} \tag{7}$$

where $w_i^{f,p,r}$ are the curriculum weights for flow, pose, and rotation components at training step i, and $\mathcal{L}_i$ is the corresponding loss. λ controls the sensitivity of the progression factor to loss magnitude changes. The curriculum weights are each naturally bounded (6) by the progression factor ϕ between initial value of $w_0 = 0.1$ and final value of $w_F = 1$.

2) Adaptive Learning Scheduler (RL-DDPG): The Adaptive Learning scheduler eliminates the need for manual progression design by using Reinforcement Learning to discover curriculum policies automatically. Three independent agents learn optimal weight schedules directly from training dynamics, potentially uncovering non-intuitive strategies that handcrafted functions might miss. We employ Deep Deterministic Policy Gradient (DDPG) [20] for its continuous action space and stable actor-critic learning, formulating each agent as:

$$w_i^{f,p,r} = w_0 + (w_F - w_0)a_i \tag{8}$$

$$s_i = [p_i,\ \mathcal{L}_i^{f,p,r}] \tag{9}$$

$$a_i = \mu_\theta(s_i) + \mathcal{N}_i \tag{10}$$

$$r_i = -|\mathcal{L}_i^{f,p,r}| \tag{11}$$

Each agent's actor network μ_θ (10) outputs action a_i based on state s_i, combining normalized training progress $p_i = i/N$ (where N is the estimated total training steps) with component-specific loss $\mathcal{L}_i^{f,p,r}$ (9). In practice, this estimate proves robust as RL agents adapt to minor deviations through their continuous learning. Exploration noise $\mathcal{N}_i$ and negative loss rewards (11) drive the agents to minimize their respective components while learning from replay buffers. The agents train alongside the VO model, dynamically adjusting curriculum weights to balance component-specific and overall optimization.

We instantiate actor and critic networks with two hidden layers (64, 32 units, ReLU activation; sigmoid output for actor). Default weight bounds are $w_0 = 0.1$ and $w_F = 1$. To stabilize learning, we normalize losses and rewards online (RewardScaler), apply adaptive Gaussian exploration noise scaled by $0.1 \min(a_i, 1 - a_i)$, use Adam optimizers (learning rates 10^{-4} actor, 10^{-3} critic), discount $\gamma = 0.99$, Polyak averaging $\tau = 0.001$, and maintain a 10 k-transition replay buffer. Agents update every 50 training steps with minibatches of 64 for 10 gradient iterations, softly updating target networks at each step.

Both adaptive schedulers (Self-Paced and RL-DDPG) operate online without preprocessing, dynamically responding to training dynamics—unlike the Trajectory-Based approach which requires pre-computed difficulty scores.

IV. Experiments

We evaluate with ATE (7-DoF alignment [21]) and AUC ($[0, 1]$m window [12]). Training on the full TartanAir set follows DPVO's protocol using the 32-sequence validation split; at each checkpoint we compute AUC over 30 runs (vs 5 in DPVO) with distinct random seeds. Unless noted, all reported ATE/AUC are medians over 30 independent runs. Final benchmarks report scale-aligned ATE on TartanAir test (synthetic), EuRoC (aerial), TUM-RGBD (indoor handheld; includes dynamic-object sequences), ICL-NUIM (synthetic with noise), and KITTI odometry (outdoor driving). All models are trained on an NVIDIA DGX-1 system with 8× Tesla V100 GPUs (32GB memory each) using distributed training.

A. Training Analysis

Fig. 3 tracks validation AUC during training:

Trajectory-Based (Green): This baseline curriculum follows a classical three-phase progression through easy, medium, and hard motion complexity tiers. Training reaches baseline performance at 22 K steps (31 % time reduction) and achieves modest improvement with final AUC of 0.83 (3.75 % over baseline), validating structured difficulty progression effectiveness.

Self-Paced (Blue): Using exponential progression with $\lambda = 0.1$, this approach delivers the fastest convergence to baseline (18 K steps, 47 % reduction) and highest final performance (AUC = 0.87, 8.75 % improvement). The smooth adaptive weighting acts as implicit regularization, enabling stable training beyond baseline limitations.

RL-DDPG (Red): The RL-DDPG scheduler (Section III-B2) demonstrates autonomous curriculum discovery, achieving AUC = 0.85 (6.25 % over baseline) despite slower initial

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

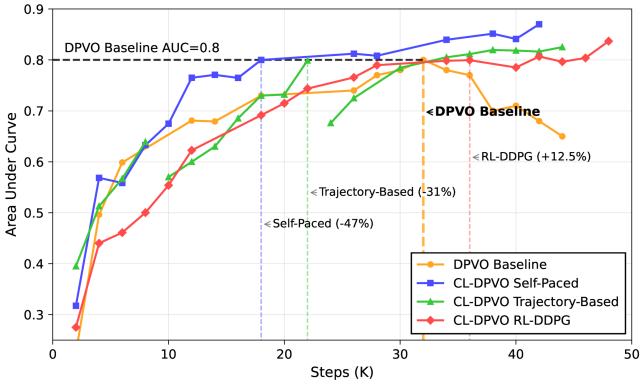

Fig. 3. Validation set progression during training. Early stopping applied when ATE plateaus. Vertical lines: steps to match baseline.

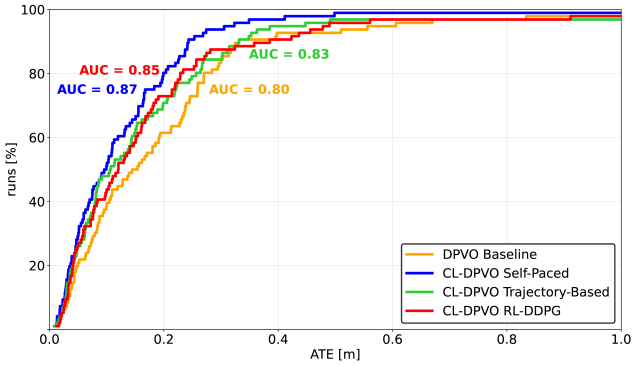

Fig. 5. AUC ($[0, 1]$m window) on the 32-sequence TartanAir validation set. All curricula improve on baseline DPVO; Self-Paced attains the highest score (AUC = 0.87).

convergence due to exploration. Importantly, while baseline DPVO degrades after 32 K steps, all curriculum methods maintain stable convergence through structured learning progressions.

Fig. 4 reveals the DDPG agents' evolving strategy through exploration to convergence. Initial exploration (0–10 K steps) shows dramatic volatility—rotation (green) plunges to 0.25, flow (blue) peaks at 1.0—before a stable hierarchy emerges at 20–22 K: flow 0.9, rotation 0.74, pose (orange) 0.43. This convergence to stable policy coincides with validation performance gains.

The discovered hierarchy addresses a fundamental training challenge: despite scale factors (10 for pose, 0.1 for flow) balancing loss magnitudes, standard training underemphasizes optical flow due to its weak correlation with trajectory error. Yet flow provides critical geometric correspondences for generalization. Through exploration, the DDPG agent autonomously establishes flow dominance with moderate rotation attention while reducing translation weight.

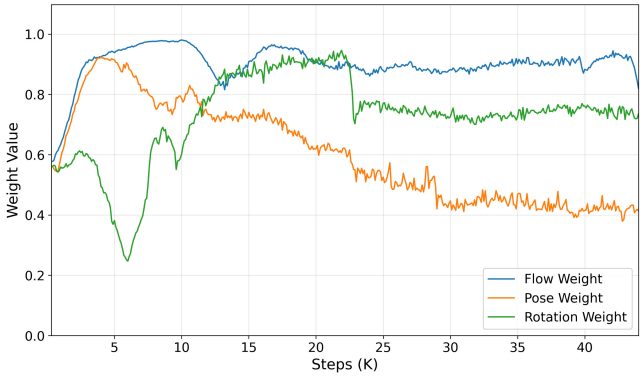

Fig. 4. RL-DDPG curriculum weights during training. After volatile exploration (rotation dips to 0.25 at 7 K; flow peaks at 1.0 then dips), stable hierarchy emerges at 20–22 K: flow (blue) 0.9, rotation (green) 0.74, pose (orange) 0.43.

B. Validation Set Comparison

Fig. 5 presents cumulative AUC performance within the $[0, 1]$m error range for the TartanAir validation set, following DPVO's evaluation protocol. All curricula outperform baseline: Self-Paced (0.87; $+8.75\,\%$), RL-DDPG (0.85; $+6.25\,\%$), and Trajectory-Based (0.83; $+3.75\,\%$). The pronounced knee at 0.25 m marks the transition between typical and challenging cases. Methods achieving steeper curves at this point demonstrate superior mid-error regime robustness, with adaptive component weighting (Self-Paced, RL-DDPG) outperforming motion-based ordering alone.

C. Motion Complexity Robustness Analysis

We evaluate robustness across motion complexity by binning validation trajectories into a 5×5 grid based on translation and rotation magnitudes (Section III-A1), computing ATE per bin. This analysis reveals whether methods maintain consistent performance across the full difficulty spectrum.

Fig. 6 reveals distinct error patterns. Baseline DPVO (d) exhibits chaotic error distribution with severe degradation at high rotation/low translation (bin 0,4: yellow hotspot) and generally poor performance in high-motion regions.

In contrast, curriculum methods (a-c) display smooth error gradients from low (dark purple) to high (light blue/green) motion complexity. Self-Paced (a) achieves the most uniform

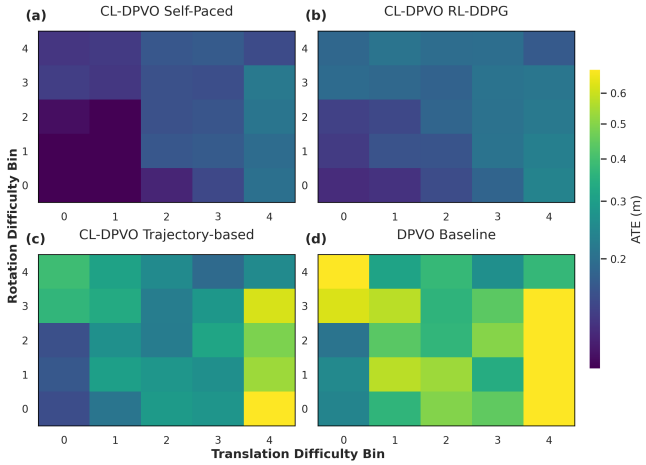

Fig. 6. ATE across motion-difficulty bins on the TartanAir validation set (darker = lower error). Curriculum variants (a-c) keep errors low across bins, unlike baseline DPVO (d).

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

TABLE II

TARTANAIR MONOCULAR TEST SPLIT ATE (SCALE-ALIGNED). BEST METHOD WITHOUT GLOBAL OPTIMIZATION IN **BOLD**; SECOND BEST UNDERLINED; * USES GLOBAL OPTIMIZATION; [2] IMAGE-ONLY; [3] IMAGE-EVENT. LOWER IS BETTER. ABBREV.: TRAJ.-BASED = TRAJECTORY-BASED.

	ME 000	ME 001	ME 002	ME 003	ME 004	ME 005	ME 006	ME 007	MH 000	MH 001	MH 002	MH 003	MH 004	MH 005	MH 006	MH 007	Avg
ORB-SLAM3* [8]	13.61	16.86	20.57	16.00	22.27	9.28	21.61	7.74	14.44	2.92	13.51	8.18	2.59	21.91	11.70	25.88	14.38
COLMAP* [22]	15.20	5.58	10.86	3.93	2.62	14.78	7.00	18.47	12.26	13.45	13.45	20.95	24.97	16.79	7.01	7.97	12.50
DROID-SLAM* [2]	0.17	0.06	0.36	0.87	1.14	0.13	1.13	**0.06**	**0.08**	0.05	0.04	**0.02**	**0.01**	0.68	0.30	0.07	0.33
DROID-VO[2] [2]	0.22	0.15	0.24	1.27	1.04	0.14	1.32	0.77	0.32	0.13	0.08	0.09	1.52	0.69	0.39	0.97	0.58
DPVO[2] [12]	0.16	0.11	0.11	0.66	0.31	0.14	0.30	0.13	0.21	0.04	0.04	0.08	0.58	**0.17**	0.11	0.15	0.21
RAMP-VO[3] [23]	0.20	**0.04**	**0.10**	0.46	**0.16**	0.13	**0.12**	0.12	0.36	0.06	0.04	0.04	0.41	0.25	0.11	**0.07**	0.17
CL-DPVO (Traj.-Based)	0.13	0.05	**0.10**	0.40	0.24	**0.06**	0.21	0.10	0.40	**0.02**	**0.03**	0.03	0.54	0.42	0.17	0.09	0.19
CL-DPVO (RL-DDPG)	0.12	0.05	0.11	**0.35**	0.45	0.09	0.16	0.11	0.45	0.03	0.04	0.03	0.35	0.32	**0.09**	0.08	0.18
CL-DPVO (Self-Paced)	**0.10**	0.05	0.14	0.38	0.19	**0.06**	0.34	0.11	0.26	0.03	0.05	**0.02**	0.18	0.21	0.10	0.10	**0.14**

TABLE III

ATE STANDARD DEVIATION ON TARTANAIR TEST: MOTION EASY (ME), MOTION HARD (MH), AND OVERALL.

Model	ME Std[m]	MH Std[m]	Total Std[m]
DROID-VO [2]	0.483	0.477	0.483
RAMP-VO [23]	0.119	0.141	0.131
DPVO [12]	0.176	0.164	0.173
CL-DPVO (Self-Paced)	**0.117**	**0.083**	**0.105**

low-error coverage, while RL-DDPG (b) and Trajectory-Based (c) show progressive degradation primarily in extreme motion bins. This uniformity demonstrates that curriculum learning develops robust representations across the entire motion spectrum rather than overfitting to specific difficulty regions. The consistent performance across bins validates our core hypothesis: robustness emerges from structured training exposure rather than architectural modifications.

D. Benchmark Evaluation

TartanAir Test Results: Table II presents results on the TartanAir test split (ECCV 2020 SLAM competition). CL-DPVO (Self-Paced) achieves the lowest average ATE (0.14 m) across the 16-sequence benchmark—a 33 % reduction from baseline DPVO and 18 % improvement over RAMP-VO. The RL-DDPG and Trajectory-Based variants deliver 14 % and 9 % improvements respectively. Self-Paced maintains sub-0.20 m accuracy on 13 of 16 sequences and reduces worst-case error by over 50 % on challenging tracks (ME003, MH004). Fig. 7 demonstrates this on MH004, where CL-DPVO maintains sub-0.4 m errors while DPVO exceeds 1.0 m.

Fig. 8 visualizes performance consistency across runs. On the challenging MH004 track, DPVO exhibits high variability (0.3 m–1.4 m range) while CL-DPVO maintains stable ∼0.2 m accuracy. This robustness extends across all motion-hard sequences with tighter error distributions.

Table III quantifies between-sequence consistency: Self-Paced reduces cross-sequence standard deviation to 0.117 m (ME) and 0.083 m (MH)—a 49 % improvement on hard sequences and 39 % overall compared to DPVO, significantly outperforming DROID-VO's 0.483 m and RAMP-VO's 0.131 m. Because these statistics aggregate per-sequence medians across 30 stochastic runs, the tighter spread reflects genuine robustness to scene content rather than seed noise.

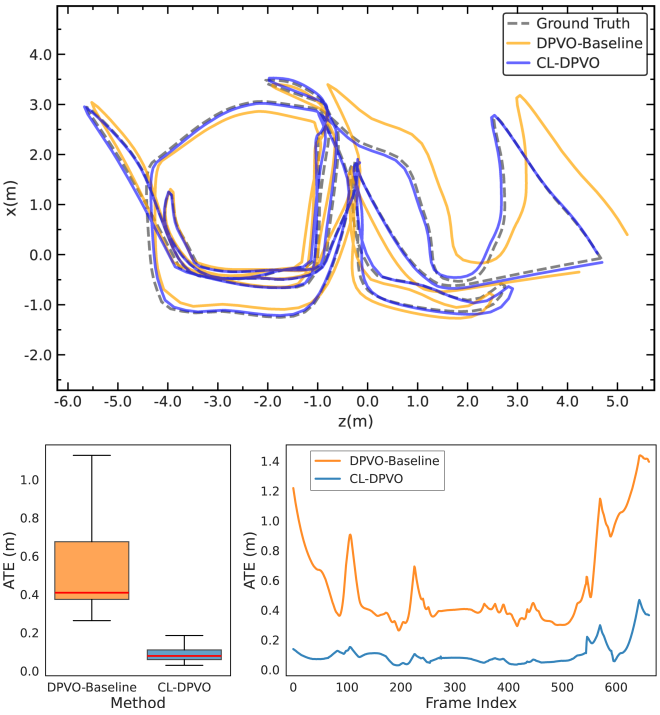

Fig. 7. TartanAir MH004: x-z trajectory (top) and frame-wise ATE (bottom; left strip shows ATE distribution: median, quartiles). CL-DPVO Self-Paced (blue) follows ground truth more closely, reducing total error (79 m vs 150 m).

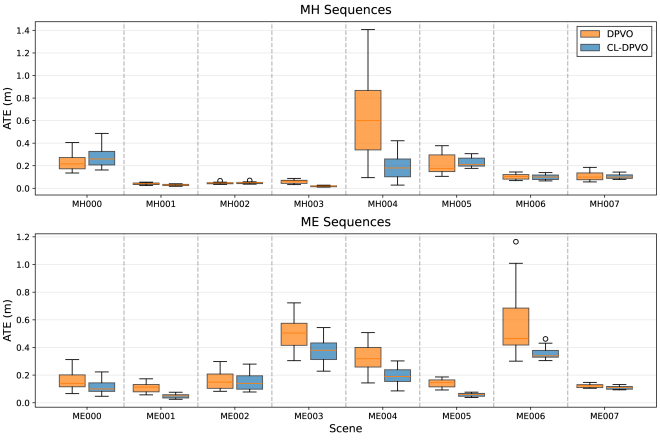

Fig. 8. TartanAir test split robustness. ATE box plots for DPVO (orange) and CL-DPVO (blue) on ME/MH. CL-DPVO yields lower median error and variance.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

TABLE IV
ABLATION STUDY ISOLATING THE EFFECT OF ADAPTIVE FLOW WEIGHTING (w_f) IN THE SELF-PACED CURRICULUM.

Method	ATE ↓ [m]	Steps to baseline [K]
DPVO (baseline)	0.21	32
CL-DPVO (Self-Paced)	**0.14**	**18**
CL-DPVO ($w_f = 1$ fixed)	0.17	27

E. Ablation Study

We investigate the contribution of key curriculum components through controlled ablations.

Impact of Adaptive Flow Weighting: To isolate the effect of curriculum-based flow adaptation, we compare against a variant with fixed flow weight ($w_f = 1$) while maintaining the Self-Paced schedule for pose and rotation weights (w_p, w_r). As shown in Table IV, fixing w_f degrades performance by $21\,\%$ (ATE: $0.14\,\mathrm{m} \rightarrow 0.17\,\mathrm{m}$) and increases convergence time by $50\,\%$ ($18\,\mathrm{K} \rightarrow 27\,\mathrm{K}$ iterations). This confirms that *adaptive* flow weighting—not merely its inclusion—drives the robustness gains.

Curriculum Schedule Analysis: The Self-Paced variant outperforms the baseline by $33\,\%$ in accuracy and $44\,\%$ in convergence speed, validating our progressive difficulty approach. The fixed-weight variant's intermediate performance (between baseline and full curriculum) demonstrates that partial curriculum application yields partial benefits, emphasizing the importance of coordinated weight scheduling across all loss components.

F. Zero-Shot Transfer to Real-World Benchmarks

A critical test of VO robustness is generalization beyond training conditions. We evaluate zero-shot transfer by testing TartanAir-trained models on three real-world datasets without fine-tuning.

EuRoC MAV: CL-DPVO (Self-Paced) achieves an average ATE of $0.091\,\mathrm{m}$ on the 11 EuRoC sequences (indoor MAV flights, Table V)—$13\,\%$ lower than DPVO ($0.105\,\mathrm{m}$) and $51\,\%$ below DROID-VO ($0.186\,\mathrm{m}$) despite seeing no real-world data. RL-DDPG attains $0.094\,\mathrm{m}$ ($10\,\%$ improvement). These zero-shot gains indicate that curriculum learning boosts fundamental robustness beyond dataset-specific tuning. Fig. 9 demonstrates this on the challenging V2_03 sequence, where CL-DPVO maintains tighter path following with fewer error spikes during aggressive maneuvers.

TUM-RGBD: On the indoor freiburg1 set (Table VI), Self-Paced averages $0.079\,\mathrm{m}$ ATE—$9\,\%$ below DPVO ($0.089\,\mathrm{m}$) and $20\,\%$ under DROID-VO ($0.098\,\mathrm{m}$). RL-DDPG also surpasses all baselines, with largest gains on the challenging *360* and *desk* sequences, confirming robustness to blur and rapid motion.

On the dynamic freiburg3 walking_xyz sequence (Table VII), designed to test robustness to pervasive moving objects [5], CL-DPVO (Self-Paced) attains $0.025\,\mathrm{m}$ ATE—a $46\,\%$ reduction from DPVO's $0.041\,\mathrm{m}$. This performance is competitive with DROID-VO ($0.019\,\mathrm{m}$) while outperforming ORB-SLAM3 ($0.035\,\mathrm{m}$) and DVO-SLAM ($0.100\,\mathrm{m}$), confirm-

TABLE V
ATE ON EUROC MAV TEST SPLIT FOR ZERO-SHOT TRANSFER EVALUATION.

	TartanVO [24]	SVO [25]	DSO [9]	DROID-VO [2]	DPVO [12]	CL-DPVO (Traj.-Based)	CL-DPVO (RL-DDPG)	CL-DPVO (Self-Paced)
MH01	0.639	0.100	**0.046**	0.163	0.087	0.083	0.069	0.081
MH02	0.325	0.120	0.046	0.121	0.055	0.060	0.044	**0.030**
MH03	0.550	0.410	0.172	0.242	0.158	0.148	**0.120**	0.122
MH04	1.153	0.430	3.810	0.399	0.137	0.151	0.144	**0.133**
MH05	1.021	0.300	**0.110**	0.270	0.114	0.123	0.115	0.114
V101	0.447	0.070	0.089	0.103	**0.050**	0.051	0.053	0.051
V102	0.389	0.210	**0.107**	0.165	0.140	0.146	0.145	0.118
V103	0.622	-	0.903	0.158	0.086	**0.055**	0.061	0.063
V201	0.433	0.110	**0.044**	0.102	0.057	0.060	0.078	0.065
V202	0.749	0.110	0.132	0.115	0.049	0.056	0.052	**0.045**
V203	1.152	1.080	1.152	0.204	0.211	0.219	**0.149**	0.178
Avg	0.680	0.294	0.601	0.186	0.105	0.105	0.094	**0.091**

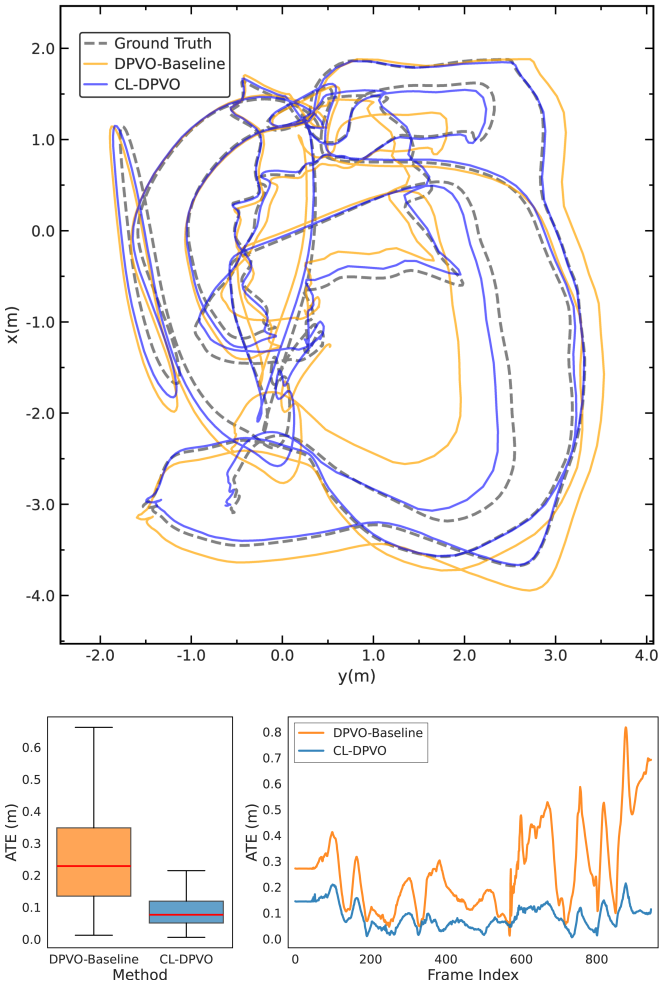

Fig. 9. EuRoC V2_03: x-y trajectory (top) and frame-wise ATE (bottom; left strip shows ATE distribution: median, quartiles). Self-Paced (blue) reduces final error ($0.13\,\mathrm{m}$ vs $0.21\,\mathrm{m}$).

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

TABLE VI
ATE ON TUM-RGBD FREIBURG1 SEQUENCES FOR ZERO-SHOT TRANSFER EVALUATION USING MONOCULAR VO ONLY.

	DROID-VO [2]	DPVO [12]	CL-DPVO (Traj.-Based)	CL-DPVO (RL-DDPG)	CL-DPVO (Self-Paced)
360	0.161	0.135	0.130	0.127	**0.122**
desk	0.028	0.038	0.050	0.061	**0.025**
desk2	0.099	0.048	**0.041**	0.049	0.048
floor	**0.033**	0.040	0.049	0.046	0.036
plant	0.028	0.036	0.026	**0.023**	0.027
room	**0.327**	0.394	0.393	0.329	0.351
rpy	**0.028**	0.034	0.045	0.031	0.031
teddy	0.169	0.064	0.074	**0.048**	0.056
xyz	0.013	0.012	**0.011**	0.012	0.013
Avg	0.098	0.089	0.091	0.081	**0.079**

TABLE VII
ZERO-SHOT ATE ON TUM-RGBD FREIBURG3 WALKING_XYZ DYNAMIC SEQUENCE, WHICH FEATURES PERVASIVE MOVING OBJECTS TO TEST ROBUSTNESS UNDER SCENE DYNAMICS.

	ORB-SLAM3 [8]	DVO-SLAM [26]	DROID-VO [2]	DPVO [12]	CL-DPVO (Traj.-Based)	CL-DPVO (Self-Paced)
fr3_walking_xyz	0.035	0.100	**0.019**	0.041	0.040	0.025

ing that adaptive loss weighting maintains robustness under pervasive scene dynamics.

ICL-NUIM: The ICL-NUIM dataset presents unique challenges with repetitive textures and enhanced noise models in indoor settings. Table VIII shows results against leading VO and SLAM methods, focusing on approaches that succeed across all sequences.

CL-DPVO (Trajectory-Based) achieves the best ATE on this indoor synthetic benchmark with $0.063\,\mathrm{m}$—$32\,\%$ lower than DPVO-Fast ($0.093\,\mathrm{m}$) and superior to all other methods. Unlike prior datasets where Self-Paced dominates, the explicit easy-to-hard schedule of the Trajectory-Based curriculum excels under ICL-NUIM's noise-heavy, precision-oriented scenes, underscoring the adaptability of our CL framework.

KITTI Odometry: To assess transfer to driving scenarios, Table IX reports zero-shot ATE (7-DoF) on KITTI sequences 00–10 (no fine-tuning). CL-DPVO attains the best average ATE on KITTI 00–10 ($42.09\,\mathrm{m}$), improving over DPVO ($53.61\,\mathrm{m}$), DROID-VO ($54.19\,\mathrm{m}$), and DPV-SLAM ($53.03\,\mathrm{m}$). Gains are strongest on longer, loop-heavy sequences (00, 05–07, 09), while DROID-VO leads on 08 and DPVO on short tracks (03, 10).

TABLE VIII
ATE ON ICL-NUIM SYNTHETIC BENCHMARK. METHODS MARKED WITH (*) USE GLOBAL OPTIMIZATION / LOOP CLOSURE.

	DROID-SLAM* [2]	DROID-VO [2]	SVO [25]	DSO [9]	DSO-Realtime [9]	DPVO [12]	DPVO-Fast [12]	CL-DPVO (Traj.-Based)	CL-DPVO (RL-DDPG)	CL-DPVO (Self-Paced)
lr-kt0	0.008	0.010	0.02	0.01	0.02	0.006	0.008	0.006	**0.005**	0.007
lr-kt1	0.027	0.123	0.07	0.02	0.03	0.006	0.007	**0.004**	0.008	0.005
lr-kt2	0.039	0.072	0.09	0.06	0.33	0.023	0.021	**0.018**	0.020	0.022
lr-kt3	0.012	0.032	0.07	0.03	0.06	0.010	0.010	**0.005**	0.006	0.006
of-kt0	0.065	0.095	0.34	0.21	0.29	0.067	0.071	**0.007**	**0.007**	**0.007**
of-kt1	0.025	0.041	0.28	0.83	0.64	0.012	0.015	**0.008**	**0.008**	0.009
of-kt2	0.858	0.842	0.14	0.36	0.23	0.017	0.018	**0.015**	0.026	0.024
of-kt3	0.481	0.504	**0.08**	0.64	0.46	0.635	0.593	0.442	0.459	0.466
Avg	0.189	0.215	0.136	0.270	0.258	0.097	0.093	**0.063**	0.067	0.068

TABLE IX
ATE ON KITTI ODOMETRY (00–10). METHODS MARKED WITH (*) USE GLOBAL OPTIMIZATION / LOOP CLOSURE.

	DPVO [12]	DROID-VO [2]	DPV-SLAM* [27]	DROID-SLAM* [2]	CL-DPVO (Self-Paced)
00	113.21	98.43	112.80	92.10	**69.37**
01	12.69	84.20	**11.50**	344.60	16.73
02	123.40	108.80	123.53	–	**101.38**
03	**2.09**	2.58	2.50	2.38	4.38
04	0.68	0.93	0.81	1.00	**0.52**
05	58.96	59.27	57.80	118.50	**48.29**
06	54.78	64.40	54.86	62.47	**38.87**
07	19.26	24.20	18.77	21.78	**15.41**
08	115.90	**64.55**	110.49	161.60	112.15
09	75.10	71.80	76.66	–	**41.10**
10	**13.63**	16.91	13.65	118.70	14.83
Avg	53.61	54.19	53.03	–	**42.09**

TABLE X
TRAINING COST COMPARISON. †EXCLUDES OFFLINE PREPROCESSING TIME.

Model	Steps [K]	FLOPs [P]	Time [h]
DPVO [12]	32	399	96
CL-DPVO Traj.-Based†	38	473	114
CL-DPVO Self-Paced	42	523	126
CL-DPVO RL-DDPG	48	598	144

G. Computational Analysis

Table X quantifies the training-cost trade-off: Self-Paced spends $31\,\%$ more wall-time to cut error by $33\,\%$, delivering a net efficiency gain. RL-DDPG yields $14\,\%$ improvement despite exploration overhead, and Trajectory-Based validates the motion metric with $9\,\%$ gains for only $19\,\%$ additional cost. Crucially, inference latency remains identical to DPVO, so the one-time training cost delivers permanent robustness at zero deployment overhead.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

V. Conclusions

Our findings demonstrate that curriculum learning substantially enhances monocular VO robustness via training methodology refinement. Our dual-paradigm framework achieves up to $33\,\%$ error reduction on TartanAir with only $31\,\%$ additional training time, while the Self-Paced variant converges $47\,\%$ faster.

These gains generalize without fine-tuning across outdoor driving (KITTI, $21\,\%$), aerial navigation (EuRoC, $13\,\%$), indoor handheld (TUM-RGBD, $9\,\%$), noise-corrupted scenes (ICL-NUIM, $32\,\%$), and dynamic environments ($46\,\%$ improvement on freiburg3 walking_xyz). Motion-difficulty analysis reveals the mechanism: curriculum training produces uniform robustness across the complexity spectrum, eliminating performance degradation on aggressive sequences. Notably, RL-DDPG agents autonomously discover flow-centric strategies that hand-crafted schedules might miss.

Both variants integrate as drop-in replacements preserving real-time inference. When ground-truth motion is unavailable, lightweight proxies (teacher VO, IMU statistics, flow variance) could enable trajectory-based curricula. The one-time training investment delivers permanent robustness at zero deployment overhead. While corner cases like extreme illumination or rare weather events remain challenging, our work establishes curriculum learning as a fundamental tool for perception robustness by demonstrating that reliability emerges from *how* we train rather than *what* we train.

References

[1] C. J. Schuler, M. Hirsch, S. Harmeling, and B. Schölkopf, "Learning to deblur," *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 7, pp. 1439–1451, 2015.

[2] Z. Teed and J. Deng, "Droid-slam: Deep visual slam for monocular, stereo, and rgb-d cameras," *Advances in neural information processing systems*, vol. 34, pp. 16 558–16 569, 2021.

[3] 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)*. IEEE, 2020, pp. 4909–4916.

[4] 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, 2016.

[5] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of rgb-d slam systems," in *2012 IEEE/RSJ international conference on intelligent robots and systems*. IEEE, 2012, pp. 573–580.

[6] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *The international journal of robotics research*, vol. 32, no. 11, pp. 1231–1237, 2013.

[7] A. Handa, T. Whelan, J. McDonald, and A. J. Davison, "A benchmark for rgb-d visual odometry, 3d reconstruction and slam," in *2014 IEEE international conference on Robotics and automation (ICRA)*. IEEE, 2014, pp. 1524–1531.

[8] C. Campos, R. Elvira, J. J. G. Rodríguez, J. 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, 2021.

[9] 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, 2017.

[10] 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)*. IEEE, 2017, pp. 2043–2050.

[11] R. Li, S. Wang, Z. Long, and D. Gu, "Undeepvo: Monocular visual odometry through unsupervised deep learning," in *2018 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2018, pp. 7286–7291.

[12] 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.

[13] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, "Curriculum learning," in *Proceedings of the 26th Annual International Conference on Machine Learning*, ser. ICML '09. New York, NY, USA: Association for Computing Machinery, 2009, p. 41–48.

[14] M. Kumar, B. Packer, and D. Koller, "Self-paced learning for latent variable models," *Advances in neural information processing systems*, vol. 23, 2010.

[15] A. Graves, M. G. Bellemare, J. Menick, R. Munos, Kavukcuoglu, and Koray, "Automated curriculum learning for neural networks," in *international conference on machine learning*. Pmlr, 2017, pp. 1311–1320.

[16] M. Wang, Y. Qin, R. Li, Z. Liu, Z. Tang, and K. Li, "Awdepth: Monocular depth estimation for adverse weather via masked encoding," *IEEE Transactions on Industrial Informatics*, 2024.

[17] Y. Xiangli, L. Xu, X. Pan, N. Zhao, A. Rao, C. Theobalt, B. Dai, and D. Lin, "Bungeenerf: Progressive neural radiance field for extreme multi-scale scene rendering," in *European conference on computer vision*. Springer, 2022, pp. 106–122.

[18] Z. Zhang, R. Peng, Y. Hu, and R. Wang, "Geomvsnet: Learning multi-view stereo with geometry perception," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2023, pp. 21 508–21 518.

[19] M. R. U. Saputra, P. P. De Gusmao, S. Wang, A. Markham, and N. Trigoni, "Learning monocular visual odometry through geometry-aware curriculum learning," in *2019 international conference on robotics and automation (ICRA)*. IEEE, 2019, pp. 3549–3555.

[20] H. Tan, "Reinforcement learning with deep deterministic policy gradient," in *2021 International Conference on Artificial Intelligence, Big Data and Algorithms (CAIBDA)*, 2021, pp. 82–85.

[21] S. Umeyama, "Least-squares estimation of transformation parameters between two point patterns," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 13, no. 04, pp. 376–380, 1991.

[22] J. L. Schönberger and J.-M. Frahm, "Structure-from-motion revisited," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 4104–4113.

[23] R. Pellerito, M. Cannici, D. Gehrig, J. Belhadj, O. Dubois-Matra, M. Casasco, and D. Scaramuzza, "Deep visual odometry with events and frames," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS. IEEE)*, 2024.

[24] W. Wang, Y. Hu, and S. Scherer, "Tartanvo: A generalizable learning-based vo," in *Conference on Robot Learning*. PMLR, 2021, pp. 1761–1772.

[25] C. Forster, M. Pizzoli, and D. Scaramuzza, "Svo: Fast semi-direct monocular visual odometry," in *2014 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2014, pp. 15–22.

[26] C. Kerl, J. Sturm, and D. Cremers, "Dense visual slam for rgb-d cameras," in *2013 IEEE/RSJ international conference on intelligent robots and systems*. IEEE, 2013, pp. 2100–2106.

[27] L. Lipson, Z. Teed, and J. Deng, "Deep patch visual slam," *arXiv preprint arXiv:2408.01654*, 2024.