# Dual-Agent Reinforcement Learning for Adaptive and Cost-Aware Visual–Inertial Odometry

Supplementary Material

## A. Training Details

### A.1. Bias Encoder Training

Our training process is staged, following the kinematic dependencies.

**Stage 1: Gyro Bias Training.** We first train $f^g_{bias}$ to correct the raw angular velocity: $\hat{\omega}(k) = \tilde{\omega}(k) - f^g_{bias}(\tilde{\Omega}, \mathbf{n}_g)$. The corrected $\hat{\omega}(\tau)$ is integrated to predict orientation $\hat{\mathbf{q}}_t$, starting from $\hat{\mathbf{q}}_0$ ($\otimes$ is quaternion multiplication):

$$\hat{\mathbf{q}}_t = \hat{\mathbf{q}}_0 + \int_0^t \frac{1}{2}\hat{\mathbf{q}}(\tau) \otimes \begin{bmatrix} 0 \\ \hat{\omega}(\tau) \end{bmatrix} d\tau \tag{8}$$

Supervision is applied via the orientation error: $\mathcal{L}_{gyro} = \|\hat{\mathbf{q}}_t \ominus \mathbf{q}^{GT}_t\|^2$ ($\ominus$ denotes quaternion error)

**Stage 2: Accel Bias Training.** After $\mathcal{L}_{gyro}$ converges, we freeze $f^g_{bias}$ and train $f^a_{bias}$ to correct acceleration: $\hat{a}(k) = \tilde{a}(k) - f^a_{bias}(\tilde{\mathbf{A}}, \mathbf{n}_a)$. Using the accurate rotation $\hat{\mathbf{R}}(\tau)$ from Stage 1, we integrate $\hat{a}(\tau)$ in the world frame to predict velocity $\hat{\mathbf{v}}_t$:

$$\hat{\mathbf{v}}_t = \hat{\mathbf{v}}_0 + \int_0^t \left(\hat{\mathbf{R}}(\tau)\hat{a}(\tau) - \mathbf{g}\right) d\tau \tag{9}$$

Supervision is applied via the velocity L2 loss: $\mathcal{L}_{accel} = \|\hat{\mathbf{v}}_t - \mathbf{v}^{GT}_t\|^2$. All reinforcement learning experiments were implemented using a custom Gym-style environment that simulates both the Select and Fusion Agents' decision processes within a differentiable VIO pipeline. The PPO algorithm from the stable-baselines3 library was used for both agents, with separate replay buffers and reward normalization. Each agent interacts with the environment in an episodic fashion until convergence.

### A.2. PPO Hyperparameter Summary

The main hyperparameters used for PPO training are summarized in Table 7. Values were selected empirically for stable convergence across both agents.

The training stability of our dual-agent system is ensured through specific strategies tailored to each agent's reward structure:

- **Select Agent:** Due to sparse rewards and high penalties for incorrect skipping, the training process is naturally sensitive. We stabilized convergence by employing a higher entropy coefficient to encourage exploration and a conservative learning rate, preventing premature convergence to sub-optimal policies.
- **Fusion Agent:** We utilize a Curriculum Learning approach. Before RL training, MLP1 undergoes supervised pre-training. This provides robust velocity priors, avoiding the instability often associated with end-to-end reinforcement learning from scratch.

Table 7. PPO hyperparameters used for Select and Fusion Agents.

| Parameter | Select Agent | Fusion Agent |
|---|---|---|
| Environment steps | 1M | 1M |
| Learning rate | $3 \times 10^{-4}$ | $5 \times 10^{-4}$ |
| Discount factor $\gamma$ | 0.99 | 0.99 |
| GAE $\lambda$ | 0.95 | 0.95 |
| Clip ratio | 0.2 | 0.2 |
| Batch size | 64 | 64 |
| Epochs per update | 10 | 10 |
| Entropy coefficient | 0.05 | 0.02 |
| Value loss coefficient | 0.5 | 0.5 |

### A.3 Reward Design

**Select reward** Recall that the Select Agent receives a terminal, episode-level reward Eq 1. The scalars $A$, $B$, and $\epsilon$ control the trade-off between accuracy and computation.

**Choice of $\epsilon$, $A$, and $B$.** On EuRoC, running DPVO with either no skipping or fixed-ratio skipping (50%, 75%, 87.5%) yields ATE values typically in the range $[0.05, 0.2]$ m. We set $\epsilon = 0.05$, so that for "reasonable" trajectories the accuracy term

$$R_{\text{acc}} = \frac{A}{\text{ATE} + \epsilon}$$

falls roughly between $4A$ and $10A$. For the same sequences, a full run contains about $4{,}000$ frames, while 50%, 75%, and 87.5% skipping lead to approximately $N_f \approx 2{,}000$, $1{,}000$, and $500$ VO calls, respectively, so the cost term $R_{\text{cost}} = -BN_f$ lies in $[-4000B, -2000B]$ for typical policies.

Under these simplified assumptions, the reward difference between each skipping policy and the full-frame baseline reduces to a linear function of $(A, B)$. Figure 7 visualizes these boundaries and the corresponding heatmap of $\Delta R$ over $(A, B)$, showing which regions of the parameter space favor aggressive skipping versus dense VO updates. We deliberately set $(A, B)$ slightly in favour of computation, since the dense per-step shaping term is already biased toward accuracy.We also clip the episode reward to a

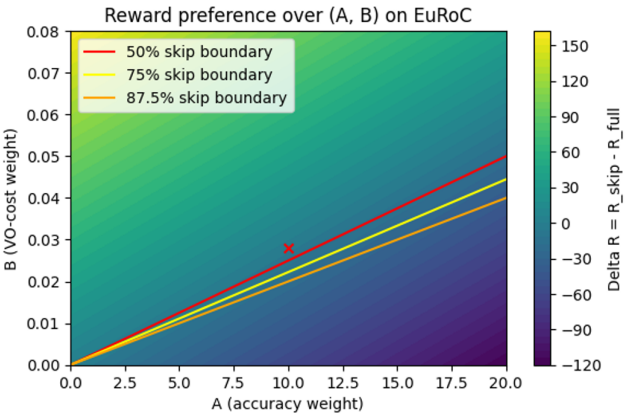


Figure 7. Fusion Agent training curve.

bounded range to avoid extreme outliers caused by occasional tracking failures.

**Dense per-step shaping.** In addition to the terminal reward, we provide a dense shaping signal at each time step $t$ based on the instantaneous pose error:

$$r_t^{\text{shape}} = -\mathrm{s} \cdot \mathrm{RMSE}_t,$$

where $\mathrm{RMSE}_t$ denotes the per-step position RMSE with respect to ground truth. This current-step RMSE acts as a low-variance shaping signal correlated with the final ATE, stabilizing PPO training, while the terminal reward enforces the desired long-horizon trade-off between trajectory accuracy and VO call frequency.

**Fusion reward and uncertainty regularization.** For the Fusion Agent, the reward has the same two-part structure: the first term directly rewards accuracy (low pose error) and is defined analogously to the Select Agent, so we omit details here. The second term penalizes the predicted uncertainty of the fused state. Concretely, we construct a diagonal covariance proxy $\Sigma_k$ from the IMU noise statistics and the VO confidence weights (as defined in the main text), and add a penalty proportional to $\mathrm{tr}(\Sigma_k)$ to the reward. This uncertainty term can be interpreted as a regularizer: it discourages the policy from relying too heavily on sensor modalities in regions where their confidence is low. In particular, when training on sequences without ground-truth trajectories, the uncertainty penalty becomes the main shaping signal, pushing the Fusion Agent toward conservative, low-uncertainty fusion patterns rather than overfitting to noisy measurements.

### A.4 Visualization of the Fusion Agent

Figure 8 plots the training loss of the Fusion Agent over environment steps. After an initial warm-up phase with high

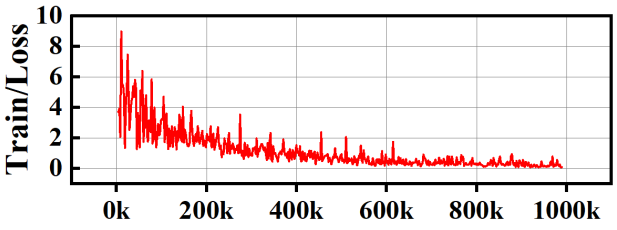


Figure 8. Reward preference over the accuracy weight A and VO-cost weight B on a EuRoC example.

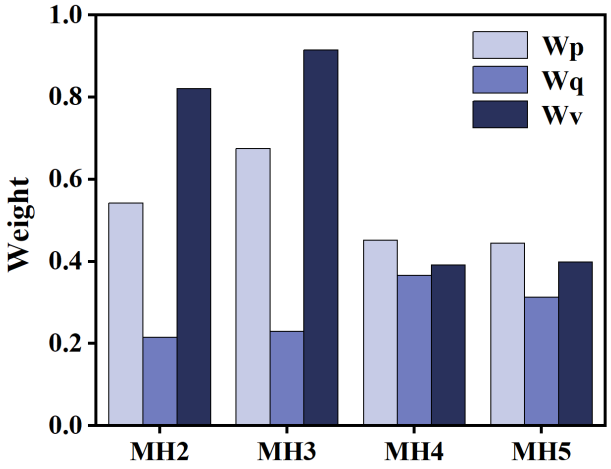


Figure 9. Visualization of learned fusion weights on EuRoC (MH2–MH5).

variance, the loss steadily decreases and stabilizes as training progresses (up to $\sim$ 1M steps), indicating that PPO converges without noticeable divergence or oscillation.

Figure 9 visualizes the learned fusion weights on several EuRoC sequences (MH2–MH5). For each sequence, we show the average weights for position, orientation, and velocity $(w_p, w_q, w_v)$, aggregated over all time steps. The agent assigns different weight patterns across sequences, reflecting how it adapts the relative contribution of each state component to the fused estimate under varying motion and visual conditions. This behaviour illustrates that the Fusion Agent learns a context-dependent policy that adapts the IMU–visual weighting to the underlying motion and visual conditions, rather than using a fixed, hand-tuned fusion scheme.

## B. IMU Pre-integration Details

Between two consecutive camera frames $(t_k, t_{k+1})$, we receive a sequence of raw IMU readings

$$\{(\omega_i^m, a_i^m, t_i)\}_{i=k}^{k+1}, \quad \Delta t_i = t_{i+1} - t_i,$$

where $\omega_i^m$ and $a_i^m$ denote the measured angular velocity and linear acceleration, respectively. At inference time, the

pre-trained bias networks $f^g_{\text{bias}}$ and $f^a_{\text{bias}}$ output gyro and accelerometer bias estimates

$$\hat{\mathbf{b}}^g_i = f^g_{\text{bias}}(\cdot), \qquad \hat{\mathbf{b}}^a_i = f^a_{\text{bias}}(\cdot),$$

which are used to correct the raw measurements:

$$\hat{\omega}_i = \omega^m_i - \hat{\mathbf{b}}^g_i, \qquad \hat{a}_i = a^m_i - \hat{\mathbf{b}}^a_i. \tag{10}$$

We follow a standard discrete-time IMU pre-integration scheme to obtain the relative motion $(\Delta\mathbf{p}_k, \Delta\mathbf{q}_k, \Delta\mathbf{v}_k)$ over the interval $[t_k, t_{k+1}]$. Let $\Delta\mathbf{R}_{k,i} \in \mathrm{SO}(3)$, $\Delta\mathbf{v}_{k,i}$, and $\Delta\mathbf{p}_{k,i}$ denote the pre-integrated rotation, velocity, and position from $t_k$ to $t_i$ in the IMU frame. We initialize

$$\Delta\mathbf{R}_{k,k} = \mathbf{I}, \quad \Delta\mathbf{v}_{k,k} = \mathbf{0}, \quad \Delta\mathbf{p}_{k,k} = \mathbf{0},$$

and iteratively propagate for $i = k, \ldots, k+1$:

$$\begin{aligned}
\Delta\mathbf{R}_{k,i+1} &= \Delta\mathbf{R}_{k,i}\,\mathrm{Exp}\big(\hat{\omega}_i\,\Delta t_i\big),\\
\Delta\mathbf{v}_{k,i+1} &= \Delta\mathbf{v}_{k,i} + \Delta\mathbf{R}_{k,i}\,\hat{a}_i\,\Delta t_i,\\
\Delta\mathbf{p}_{k,i+1} &= \Delta\mathbf{p}_{k,i} + \Delta\mathbf{v}_{k,i}\,\Delta t_i + \frac{1}{2}\,\Delta\mathbf{R}_{k,i}\,\hat{a}_i\,\Delta t_i^2,
\end{aligned} \tag{11}$$

where $\mathrm{Exp}(\cdot)$ is the exponential map from $\mathfrak{so}(3)$ to $\mathrm{SO}(3)$. At the end of the interval, we obtain

$$\Delta\mathbf{R}_k \triangleq \Delta\mathbf{R}_{k,k+1}, \quad \Delta\mathbf{v}_k \triangleq \Delta\mathbf{v}_{k,k+1}, \quad \Delta\mathbf{p}_k \triangleq \Delta\mathbf{p}_{k,k+1}.$$

The rotation $\Delta\mathbf{R}_k$ is converted to a unit quaternion $\Delta\mathbf{q}_k$ (and to axis–angle form when constructing the RL observations).

In summary, for each interval $(t_k, t_{k+1})$ the module outputs $(\Delta\mathbf{p}_k, \Delta\mathbf{q}_k, \Delta\mathbf{v}_k)$ and bias estimates $\hat{\mathbf{b}}_k = (\hat{\mathbf{b}}^g_k, \hat{\mathbf{b}}^a_k)$, which are fed to the Select and Fusion Agents as part of their state.

## C. Analysis of Deployment Feasibility

We analyze our architecture's feasibility based on two key factors: First, we analyze the model size, the total combined model size for our entire VIO framework is only 32.1 MB, making them trivial to store and load on resource-constrained edge devices. Second, these new modules introduce negligible computational overhead. The inference cost of these small networks is minimal (typically $<$ 1ms), our framework does not add any new, computationally heavy layers to the main loop, and our architecture's primary efficiency gain comes from intelligent computational gating to reduces the total average FLOPs and inference time required per frame.

## D. Additional Experimental Results

**Comparison with classical CPU-based VIO on TUM-VI.** Table 8 reports SE(3)-aligned RMSE ATE on TUM-VI for classical CPU-based monocular VIO systems. Our method achieves accuracy comparable to DM-VIO on average, while clearly outperforming VINS and OKVIS on most sequences.

Table 8. Comparison with classical CPU-based monocular visual-inertial odometry systems on the TUM-VI dataset. We report SE(3)-aligned RMSE ATE (m).

| Seq | VINS | OKVIS | DM-VIO | Ours | Length |
|---|---|---|---|---|---|
| Corr1 | 0.63 | 0.33 | 0.19 | 0.23 | 305 |
| Corr2 | 0.95 | 0.47 | 0.47 | 0.39 | 322 |
| Corr3 | 1.56 | 0.57 | 0.24 | 0.52 | 300 |
| Mag1 | 2.19 | 3.49 | 2.35 | 3.12 | 918 |
| Mag2 | 3.11 | 2.73 | 2.24 | 2.21 | 561 |
| Mag3 | 0.40 | 1.22 | 1.69 | 0.56 | 566 |
| Room1 | 0.07 | 0.06 | 0.03 | 0.11 | 146 |
| Room2 | 0.07 | 0.11 | 0.13 | 0.10 | 142 |
| Room3 | 0.11 | 0.07 | 0.09 | 0.09 | 135 |
| Slide1 | 0.68 | 0.86 | 0.31 | 0.55 | 289 |
| Slide2 | 0.84 | 2.15 | 0.87 | 0.91 | 299 |
| Slide3 | 0.69 | 2.58 | 0.60 | 0.76 | 383 |
| Avg | 0.94 | 1.22 | 0.77 | 0.80 | 363.8 |

**Cumulative Component Analysis.** Finally, to provide a holistic view of our contributions, Table 9 reports the cumulative impact of the three main components (bias encoder, Fusion Agent, Select Agent) on EuRoC and TUM-VI.

Table 9. Cumulative contribution of each component. Removing a component from the full system (ALL) degrades accuracy or efficiency.

| Component | ATE (EuRoC) | ATE (TUM-VI) | FPS |
|---|---|---|---|
| ALL | 0.092 | 0.80 | 39 |
| - Bias Encoder | 0.279 | 1.13 | 40 |
| - Fusion Agent | 0.133 | 0.94 | 39 |
| - Select Agent | 0.087 | 0.76 | 21 |

Removing the bias encoder leads to a large increase in ATE on both datasets, confirming its importance for stable pre-integration. Disabling the Fusion Agent also degrades accuracy, whereas removing the Select Agent mainly hurts efficiency (FPS drops from 39 to 21), illustrating the complementary roles of the three components.

Table 10. Cross-dataset transfer performance of Visual-Selective-VIO (VS-VIO) on EuRoC MAV. We report SE(3)-aligned RMSE ATE (m) on four sequences (MH_02–MH_05) using the official KITTI-pretrained model without retraining.

| Seq | MH2 | MH3 | MH4 | MH5 |
|---|---|---|---|---|
| VS-VIO | 1.134 | 1.032 | 1.029 | 1.355 |

**Additional baseline: Visual-Selective-VIO on EuRoC.**

To further assess adaptive VIO methods trained on driving data, we evaluate Visual-Selective-VIO (VS-VIO) on EuRoC without any retraining. We reuse the official KITTI-pretrained model and convert EuRoC sequences MH_02–MH_05 into the KITTI-style format used by the original code, including nearest-neighbor timestamp alignment between images and ground-truth and IMU interpolation as described in Sec. 4.1. Table 10 reports the SE(3)-aligned RMSE ATE (m). Compared to both classical VIO and our method (see Tables 1 and 8), VS-VIO exhibits substantially larger trajectory errors on these aggressive 6-DoF MAV sequences, confirming a strong domain gap between ground-vehicle training data and EuRoC. We therefore treat VS-VIO as a conceptually related method, but do not include it as a main numerical baseline in the core results tables.

**Evaluation on failure modes.** We conduct stress tests on the EuRoC dataset (MH_04) under three extreme conditions: (1) VO Outage, (2) Severe Blur, and (3) IMU Noise. The results in Table 11 demonstrate that while accuracy degrades under extreme stress, our method significantly outperforms the DPVO baseline. This resilience is primarily attributed to the **Fusion Agent**, which dynamically detects low visual reliability and suppresses visual weights, effectively leveraging the IMU Bias Estimator as a safety net.

Despite the improved robustness, residual drift in VO-denied areas suggests two directions for future work: (1) incorporating long-range temporal context into confidence estimation, and (2) establishing a feedback mechanism to re-initialize the VO backend using IMU states during prolonged outages.

Table 11. Robustness Stress Test under Severe Sensor Degradation.

| **Method** | **Nominal** (Clean Data) | **VO Outage** (Blackout 2s) | **Severe Blur** (30% Frames) | **IMU Noise** (2× Noise) |
|---|---|---|---|---|
| DPVO | 0.106 | 1.928 | 1.122 | - |
| **Ours** | **0.092** | **1.114** | **0.472** | **0.128** |

**Scale Robustness and Monitoring.** Regarding scale estimation, we acknowledge that our base MLP1 primarily performs velocity smoothing and has limited authority to correct large initial scale errors. To address this, we integrated a lightweight **Online Scale Monitor**. This module calculates the motion magnitude ratio between IMU pre-integration and VO over a sliding window (15 frames). If a persistent deviation is detected, a correction factor realigns the VO scale with the IMU's metric reference. As shown in Table 12, under $\pm 20\%$ initialization error, our system with the Monitor effectively mitigates the scale drift.

**Generalization:** To assess performance with minimal domain adaptation, we evaluated a model fine-tuned only on TUM-VI (Corridor_4) applied to EuRoC (MH_04). ATE increases (**0.092m** $\rightarrow$ **0.173m**) due to the domain gap in IMU noise profiles. However, the system successfully completes the trajectory, validating robustness to unseen sensors.

Table 12. Scale Robustness Test (ATE [m]).

| **Method** | **Nominal** | **Small Init** | **Large Init** |
|---|---|---|---|
| Ours (Base) | **0.092** | 0.661 | 0.682 |
| **Ours (+Monitor)** | 0.096 | **0.522** | **0.508** |

## E. Discussion on Algorithmic Novelty

To further clarify our contributions relative to existing RL-based Visual Odometry (VO) works, we emphasize two fundamental distinctions in our dual-agent architecture:

**1. Paradigm Shift: "Shift Left" Pre-emptive Gating.** Traditional RL-based VO methods typically focus on post-processing, such as filtering keyframes or weighting features after they have been extracted. This represents a sunk cost where significant computation has already been spent. In contrast, our **Select Agent** proposes a "Shift Left" strategy: it performs pre-emptive gating on raw sensor data streams. By making skipping decisions before the visual frontend is invoked, we bypass the entire tracking and bundle adjustment pipeline for redundant frames.

**2. First RL-based VIO Fusion.** While prior works often use RL for visual parameter tuning, our framework is, to the best of our knowledge, the first to employ RL for dynamic IMU-Visual fusion. Our **Fusion Agent** goes beyond simple weighting; it learns a context-dependent policy that adapts the IMU-visual contribution based on motion dynamics and visual reliability.

## F. Limitations and Future Work

Although our dual-agent framework achieves a favourable accuracy–efficiency trade-off across EuRoC and TUM-VI, several aspects of the current study remain limited in scope and open up promising directions for future work.

**Dataset and platform coverage.** Our RL policies are trained and evaluated on EuRoC and TUM-VI using offline, log-driven environments, and thus implicitly adapt to the motion patterns, sensor characteristics, and visual statistics of these datasets. We do not yet explore cross-dataset transfer or deployment on substantially different platforms (e.g., wheeled robots, handheld AR devices, or cameras/IMUs with different noise profiles), which would be important to fully assess generalization.

**Dependence on the VO backend and initialization.** The framework assumes a reasonably calibrated VO backend and a successful initial scale estimate. In challenging conditions with persistent VO degradation or severe miscalibra-

tion, the current policies may still fail to recover, as they are not explicitly trained to handle hard re-initialization events. Designing agents that can detect such failure modes and proactively trigger re-initialization or fallback behaviours is an interesting extension.

**Hardware evaluation.** All experiments are conducted on a desktop-class CPU/GPU platform. While we show that the proposed scheduling reduces the load on the bundle adjustment module and improves throughput, we have not yet performed a systematic study on embedded or low-power hardware, nor an explicit analysis of latency, energy, and memory footprints. A more comprehensive hardware evaluation would be needed to fully validate the benefits for real-world onboard systems.

**Future work.** A natural next step is to push the framework towards real edge deployment. On the algorithmic side, we aim to incorporate hardware constraints such as latency, power, and memory directly into the reward design, and to train and evaluate the policies on embedded platforms (e.g., Jetson-class devices) under realistic compute budgets. On the system side, extending the scheduling mechanism to coordinate multiple perception modules (VO, mapping, semantics) sharing the same edge device, and developing even lighter-weight agent variants that can run at high frequency on low-power hardware, are promising directions. Ultimately, our goal is to turn the proposed dual-agent controller into a drop-in, resource-aware VIO front-end for mobile robots, drones, and AR devices operating under tight real-time and power constraints.