PRoMiSe: Persistent Multi-Robot Monitoring Sim-to-Real-to-Field

Alessio Salvatore[1,2] **Juan de los Rios**[1] **David Rytz**[1] **Maria Stamatopoulou**[1]
alessio@laelaps.ai juan@laelaps.ai david@laelaps.ai maria@laelaps.ai
[1]Laelaps AI, Zürich, Switzerland [2]ETH Zürich, Switzerland

Abstract: Persistent multi-robot patrolling is a complex real-world problem, requiring repeated observation of an environment while adapting to changes in layout and team composition. We present PRoMiSe, a multi-agent reinforcement learning policy for visibility-based persistent monitoring that commands body-frame velocities, with no patrol graph, task allocation or planner in the control loop. It combines a compact egocentric idleness summary and lidar ranges with attention over a variable number of teammates and recurrent temporal reasoning. A single policy transfers without fine-tuning to unseen simulated maps and teams of up to sixteen robots, and runs unmodified on Unitree Go2 quadrupeds at three physical sites. Matched indoor experiments with one to four robots show cumulative coverage within 0.6 percentage points of simulation. Outdoor deployments with two robots demonstrate coordinated patrolling in previously unseen environments, including an industrial site eight times the average training-map area. These results provide evidence of sim-to-real-to-field transfer for persistent multi-robot monitoring across layouts, scales, and team sizes. For visual results: https://paperanonyms.github.io/promise/.

1 Introduction

Persistent monitoring requires robots to repeatedly observe an environment so that no region remains unobserved for long. Unlike one-time coverage, patrolling is a continuing task, commonly evaluated through idleness: the time elapsed since a location was last observed [1, 2, 3].

Classical multi-robot patrolling commonly represents an environment as a graph or target set, while persistent-monitoring approaches also optimize motion along prescribed paths [4, 2, 3]. Learned visibility-based methods instead credit observation of free space within a sensor footprint. MA-G-PPO applies multi-agent reinforcement learning to this formulation [5], and GALOPP evaluates learned monitoring on physical aerial robots under sensing, communication, and localization constraints [6]. These advances motivate a further question: can a shared policy trained in compact simulated environments preserve monitoring and coordination when deployed on physical robots at previously unseen indoor and outdoor sites?

We address this question with PRoMiSe, a policy for persistent monitoring with variable team sizes, trained using Multi-Agent Proximal Policy Optimization [7]. We train PRoMiSe in a lightweight 2D simulator that captures planar motion and obstacle geometry, while omitting detailed legged dynamics and photorealistic rendering. Abstracting away these details makes training faster while retaining the information the deployed policy depends on. To improve robustness, training combines procedurally generated layouts and varying team sizes with perturbations to velocity tracking and lidar measurements. We evaluate transfer in three stages. First, in simulation, we test whether the policy generalizes to unseen maps and unseen teams of up to four times the size of the largest training team. Second, in indoor experiments with one to four Unitree Go2 quadrupeds, we evaluate the policy's ability to transfer to real-world deployment. Third, we deploy two robots at an outdoor parking lot and an industrial site to evaluate coordinated patrolling using the same policy without fine-tuning. Our contributions are a policy formulation for transferable multi-robot monitoring and an evaluation connecting simulated generalization, matched hardware performance, and deployment at unseen outdoor sites.

10th Conference on Robot Learning (CoRL 2026), Austin TX, USA.

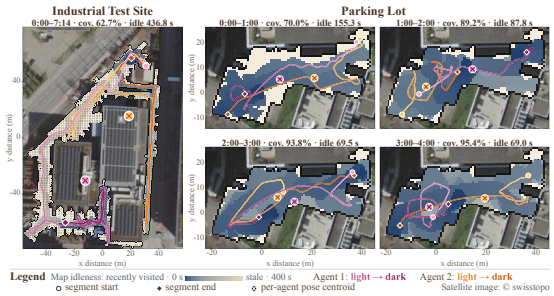


Figure 1: **PRoMiSe deployed in two real sites.** Left: a seven-minute patrol of a large industrial test site. Right: four consecutive one-minute segments of a parking lot patrol.

2 Method

Simulation and robot interface PRoMiSe issues body-frame velocity commands $u_t = [v_t^x, v_t^y, \omega_t]$ to the robot's low-level motion controller. Observation is credited to free cells within a finite-range, heading-dependent sensor field of view. Cell age resets when observed and otherwise increases; effective sensing range decreases with yaw rate to approximate reduced observation quality during rapid turns. To avoid assuming instantaneous command execution, commands are clipped to actuator limits and passed through a first-order low-pass filter:

$$\hat{u}_t = \mathrm{clip}(u_t), \qquad \bar{v}_{t+1} = \bar{v}_t + \alpha(\hat{u}_t - \bar{v}_t), \qquad \alpha = 1 - e^{-\Delta t/\tau_v}. \tag{1}$$

Multiplicative actuation noise is applied after filtering, $\tilde{v}_t = \bar{v}_{t+1} \odot (1 + \sigma_v \epsilon_t)$, where $\epsilon_t \sim \mathcal{N}(0, I_3)$ is sampled independently at each step and is not fed back into the filter state. Pose evolves as

$$p_{t+1} = p_t + \Delta t\, R(\theta_t)[\tilde{v}_t^x, \tilde{v}_t^y]^\top, \qquad \theta_{t+1} = \mathrm{wrap}_{[-\pi,\pi)}(\theta_t + \Delta t\, \tilde{\omega}_t), \tag{2}$$

with wall collisions resolved by projection.

Observations available on the robots The actor receives position normalized by the map half-extents, sine–cosine heading, 16 body-frame lidar ranges normalized by the 8 m sensing range, an eight-sector egocentric idleness summary, and identifiers encoding the robot and active team size. Hardware position and heading come from localization, while range observations are derived from onboard lidar measurements. Each idleness sector summarizes the mean staleness of free cells within a 45° angular region around the robot.

None of these representations fixes the deployment map to a particular physical size or grid dimension. Normalized position accommodates different extents, lidar describes local geometry, and the idleness summary retains eight sectors regardless of map area. The same actor can therefore process larger maps without resizing its inputs.

Teammate tokens contain normalized obstacle-aware distance, relative bearing and heading, and a slot identifier, constructed from shared poses and the map. On hardware, one robot maintains team observation timestamps and broadcasts them, allowing each robot to construct its idleness summary locally.

Reward for monitoring and executable motion The reward combines team monitoring performance with individual motion penalties:

$$r_t^{(i)} = -w_I \bar{I}_t - w_c c_t^{(i)} - w_h h_t^{(i)} - w_a q_t^{(i)} - w_s \left\| u_t^{(i)} - u_{t-1}^{(i)} \right\|_2. \tag{3}$$

Here, $\bar{I}_t$ averages bounded cell idleness, $I_k = 1 - e^{-g_k/\tau_I}$, with unseen cells assigned $I_k = 1$. Wall-contact cost $c_t^{(i)}$ measures the fraction of commanded motion blocked by obstacles. Clearance cost $h_t^{(i)} = \max(0, 1 - d_t^{(i)}/d_m)^2$ discourages approaching obstacles within margin d_m, while $q_t^{(i)}$ indicates teammate contact. The final term discourages abrupt command changes.

Shared recurrent policy We train using Multi-Agent Proximal Policy Optimization [7], with a shared actor and a centralized critic. As shown in Fig. 2, cross-attention pools the variable-size teammate set into a fixed-width embedding [8], masking absent teammates. The embedding is fused with ego observations and processed by two GRU layers, with each robot maintaining its own

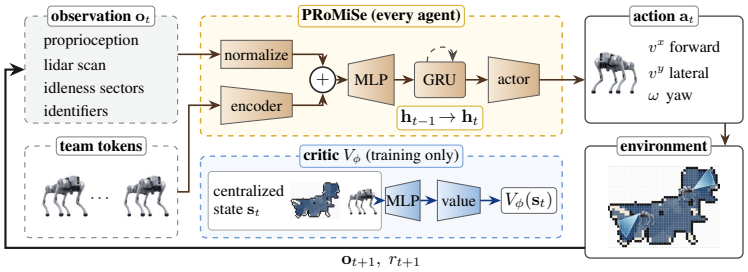


Figure 2: **PRoMiSe architecture.** Each robot executes a shared actor using ego observations and a variable-size set of teammate tokens.

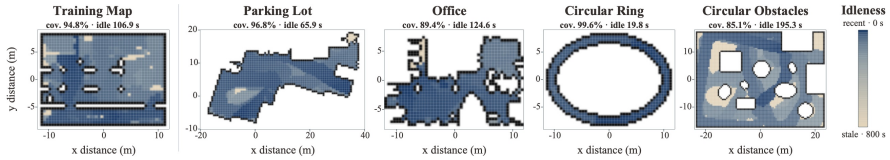


Figure 3: **Training and evaluation layouts.** Left: a procedural training environment. Right: two real floor plans and two synthetic evaluation layouts.

recurrent state. The action head selects among five velocity values per axis. Privileged team and map information is restricted to the training critic, which is discarded at deployment.

Training diversity and domain randomization Training uses 1,000 procedurally generated 18×24 m layouts with varying corridors, rooms, doors, and free-standing obstacles. Layout diversity exposes the policy to different connectivity patterns and obstacle geometries, encouraging monitoring strategies that generalize beyond a particular floor plan (Fig. 3).

Motion and sensing perturbations target discrepancies between the simplified simulator and physical deployment [9, 10]. Multiplicative Gaussian lidar noise with standard deviation 0.15 exposes the policy to uncertain ranges, approximating discrepancies in simulated sensing and real lidar measurements. Independently, replacing each ray by maximum range with probability 0.05 approximates missing returns, discouraging reliance on consistently complete scans. Training runs for 25 million agent-steps with 400 s episodes.

3 Experimental evaluation

We evaluate whether our observation and action spaces, together with the simulation model and domain randomization, capture the task-relevant information and variability needed for zero-shot transfer to physical robots. We examine three stages of transfer. **E1 (sim → sim):** Does the policy generalize to unseen maps and team sizes beyond those used in training? **E2 (sim → real):** Is simulated patrol performance preserved on physical robots as team size increases? **E3 (sim → field):** Does this transfer extend to previously unseen outdoor sites?

Setup. For E1, we evaluate on 14 unseen simulated maps with teams of 1, 2, 4, 8, and 16 robots over 800 s episodes. For E2 and E3, the simulation-trained policy is deployed without fine-tuning on Unitree Go2 robots: in a laboratory office with 1–4 robots and three runs per team size, and at an outdoor parking lot and an industrial site with two robots. Hardware runs last 300 s, and no deployment site is used during training. Each hardware run is compared with a simulation replica using the same checkpoint, site map, and initial robot poses.

Metrics. We measure cumulative coverage, the percentage of free cells observed at least once, and average idleness, the time since each free cell was last observed, averaged over cells and episode duration. For hardware comparisons, we report final coverage and episode-average idleness.

E1 On unseen maps and over 800 s episodes, twice the training episode length, increasing team size from 1 to 16 robots reduces average idleness from 304.3 to 25.8 s and raises final coverage from 91.6% to 99.7%. Performance continues to improve beyond the training range of 1–4 robots, suggesting that coordination remains effective with larger teams. MA-G-PPO [5] is in wall contact in 17.9%

site	agents	runs	avg idleness (s) ↓ sim	real	Δ (%)	coverage (%) sim	real	Δ (%)
Office	1	3	84	85	+1.2	97	97	0.0
Office	2	3	54	58	+7.4	99	98	-1.0
Office	3	3	47	49	+4.3	99	99	0.0
Office	4	3	36	43	+19.4	100	99	-1.0
Parking lot	2	5	118	124	+5.1	98	97	-1.0
Industrial	2	4	317	318	+0.3	48	50	+4.2

Table 1: Sim-to-real-to-field: Mean simulated and real patrol metrics over runs for each deployment. Relative differences are computed as $\Delta = 100(\text{real} - \text{sim})/\text{sim}$.

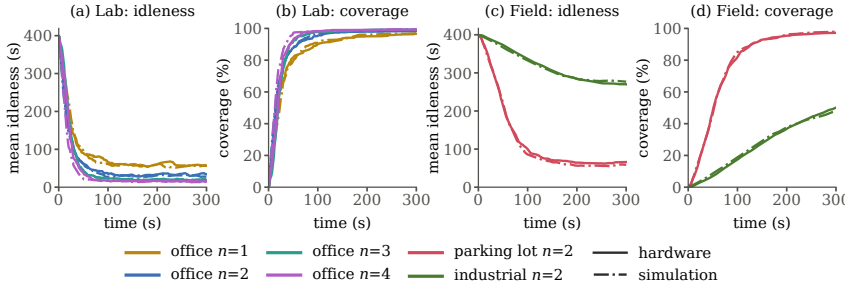


Figure 4: Mean idleness and cumulative coverage over the 300 s run in the lab (a, b) and at the field sites (c, d), hardware (solid) vs. matched simulation (dash-dot).

of agent-steps, so we did not deploy any MA-G-PPO policy on the real hardware. **E2** The policy deploys without fine-tuning across laboratory teams of 1–4 robots. Final coverage remains within one percentage point of simulation (Table 1). Average idleness is higher on hardware by 1.2%, 7.4%, 4.3%, and 19.4%, respectively. The coverage and idleness curves in Fig. 4(a, b) illustrate performance throughout the patrol. We attribute the larger gap with four robots to lower simulation fidelity at high team density: close-range interactions in the confined office are where legged dynamics deviate most from the planar motion model. **E3** The same policy operates without fine-tuning at two outdoor sites with different spatial scales: the parking lot is comparable in area to the training maps, while the industrial site is eight times the average training-map area. At the parking lot, average idleness increases by 5.1% relative to simulation, while final coverage decreases by one percentage point. At the industrial site, average idleness differs by 0.3% and coverage increases by two percentage points (Table 1; Fig. 4(c, d)). These results support zero-shot transfer across both unseen layouts and substantially larger environments. The approximately 50% coverage reached at the industrial site within 300 s also highlights the greater monitoring demands at this scale.

4 Conclusion

We presented PRoMiSe, a multi-robot patrolling policy that combines compact observations, attention over a variable number of teammates, and recurrent reasoning to issue body-frame velocity commands. Its observation and action spaces, lightweight simulation model, and domain randomization support deployment without a patrol graph or site-specific fine-tuning.

A policy trained entirely in simulation generalizes to unseen maps and teams of up to sixteen robots, with improving patrol performance beyond the training range of one to four robots. Zero-shot deployment on Unitree Go2 quadrupeds preserves high coverage across laboratory team sizes and achieves patrol performance close to matched simulation at two unseen outdoor sites. These sites span different scales, from a parking lot comparable in area to the training maps to an industrial site eight times larger.

Together, these results show that, for persistent monitoring, a lightweight simulation suffices when it captures the task-relevant quantities (obstacle geometry, sensor observations, and area idleness): detailed physical dynamics and photorealistic rendering are not always necessary to learn effective monitoring behaviors. Focusing simulation fidelity on the information the policy uses enables zero-shot transfer across unseen environments and team sizes, without training in realistic simulators or real-world fine-tuning. Future work will examine longer deployments, broader sensing and communication conditions, and explicit safety mechanisms. The partial coverage of the industrial site also motivates adapting team size and patrol priorities to larger environments.

References

[1] A. Machado et al. Multi-agent patrolling: An empirical analysis of alternative architectures. In *Multi-Agent-Based Simulation II*, pages 155–170. Springer, 2002.

[2] D. Portugal and R. Rocha. A survey on multi-robot patrolling algorithms. In *Technological Innovation for Sustainability*, pages 139–146, 2011.

[3] S. L. Smith et al. Persistent robotic tasks: Monitoring and sweeping in changing environments. *IEEE Transactions on Robotics*, 28(2):410–426, 2012.

[4] Y. Chevaleyre. Theoretical analysis of the multi-agent patrolling problem. In *Proceedings of the IEEE/WIC/ACM International Conference on Intelligent Agent Technology*, pages 302–308. IEEE Computer Society, 2004.

[5] J. Chen et al. Multi-agent reinforcement learning for visibility-based persistent monitoring. In *Proceedings IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2563–2570, 2021.

[6] M. Mishra et al. Multi-agent deep reinforcement learning for persistent monitoring with sensing, communication, and localization constraints. *IEEE Transactions on Automation Science and Engineering*, 22:2831–2843, 2025. doi:10.1109/TASE.2024.3385412.

[7] C. Yu et al. The surprising effectiveness of PPO in cooperative multi-agent games. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

[8] A. Jaegle et al. Perceiver: General perception with iterative attention. In *Proceedings International Conference on Machine Learning*, volume 139, pages 4651–4664. PMLR, 18–24 Jul 2021.

[9] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017. doi:10.1109/IROS.2017.8202133.

[10] X. B. Peng, M. Andrychowicz, W. Zaremba, and P. Abbeel. Sim-to-real transfer of robotic control with dynamics randomization. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1–8. IEEE, 2018. doi:10.1109/ICRA.2018.8460528.