Active SLAM using Deep Reinforcement Learning

Julius Arolovitch[1] and Micah Reich[2]

Abstract— We present a reinforcement learning framework for Active Simultaneous Localization and Mapping (Active SLAM) that learns high-level exploration policies to reduce pose uncertainty and maximize map coverage in unknown environments. Our method uses model-free deep reinforcement learning to learn an end-to-end policy that implicitly balances exploration and loop closure, without relying on a differentiable SLAM module. We show preliminary results demonstrating the feasibility of learning and executing exploration strategies in 2D LiDAR-based environments. Additionally, we present early work on learning actions that promote self-consistency in the constructed map, and explore the use of generative policies to capture the inherent multimodality of the action space.

I. INTRODUCTION

In traditional Simultaneous Localization and Mapping (SLAM), the objective is to incrementally build a consistent map of the environment while simultaneously estimating the robot's trajectory within it. Most SLAM systems assume that exploration is either manually guided or carried out via random or predefined motion, leaving the process of autonomous data collection largely unaddressed.

Active SLAM, by contrast, seeks to generate control actions that efficiently explore the environment while reducing map and pose uncertainty. Yet, defining a reward signal that meaningfully captures this dual objective is difficult. While metrics derived from occupancy maps or pose graph structure are theoretically grounded, they often result in sparse, noisy, or poorly shaped learning signals. This makes it challenging for a reinforcement learning agent to discover policies that balance exploration with the long-horizon goal of enforcing global map consistency through loop closures.

Classical Active SLAM methods address this by separating the process into generating candidate actions, computing their expected utility (e.g., expected information gain), and executing the highest scoring one [1]. More recent works attempt to integrate exploration and uncertainty reduction in a single policy. Frontier-based methods identify boundaries between known and unknown space and compute paths that maximize information gain, using utility functions defined over geometry and prior observations [2]. However, these methods rely on handcrafted utility functions and are limited in their ability to generalize to unseen environments or adapt to perceptual aliasing.

Deep RL-based approaches instead learn policies directly from data. Placed and Castellanos [3] proposed a Dueling Double DQN agent that uses a handcrafted reward signal incorporating D-optimality to guide exploration. Other work has focused on learning pointgoal-style exploration policies [4], though these often ignore SLAM uncertainty and operate in known maps or under oracle localization.

In this work, we propose a deep reinforcement learning framework for Active SLAM that learns to select high-level navigation goals to efficiently explore unknown environments while enforcing global map consistency through loop closures. Our framework uses a dual-objective reward function that promotes both map coverage and pose graph constrainedness. We decouple high-level planning from low-level control by training the agent to select desirable goal positions, which are then executed using classical motion planners. We evaluate both continuous and discrete action spaces using PPO and DQN, and validate our reward design through controlled experiments in 2D LiDAR environments.

Motivated by the observation that the optimal action distribution in exploration tasks is often multimodal, we explore the potential of generative policies—such as diffusion-based or energy-based models [5], [6]—to better capture the diversity of plausible exploration strategies, particularly in ambiguous or perceptually aliased settings. While our current work focuses on simpler policy architectures to validate the overall framework, we lay the groundwork for future integration of generative methods, which we believe offer a promising path toward robust and flexible Active SLAM.

II. METHODS

A. Graph-based SLAM

Graph-based SLAM formulates the problem of simultaneous localization and mapping as the optimization of a factor graph, where each node represents a robot pose and each factor encodes a probabilistic constraint between poses, with these constraints arising either from odometry or from loop closure detections. Mathematically, given a set of robot poses $\mathcal{X} = x_0, x_1, \ldots, x_T$ and measurements $\mathcal{Z} = z_{i,j}$ relating poses i and j, the Graph SLAM problem seeks the configuration of poses which maximizes the log-likelihood of the joint measurement density:

[1]Julius Arolovitch is with the Department of Electrical and Computer Engineering, Carnegie Mellon University, jarolovi@andrew.cmu.edu

[2]Micah Reich is with the Robotics Institute, Carnegie Mellon University, mreich@andrew.cmu.edu

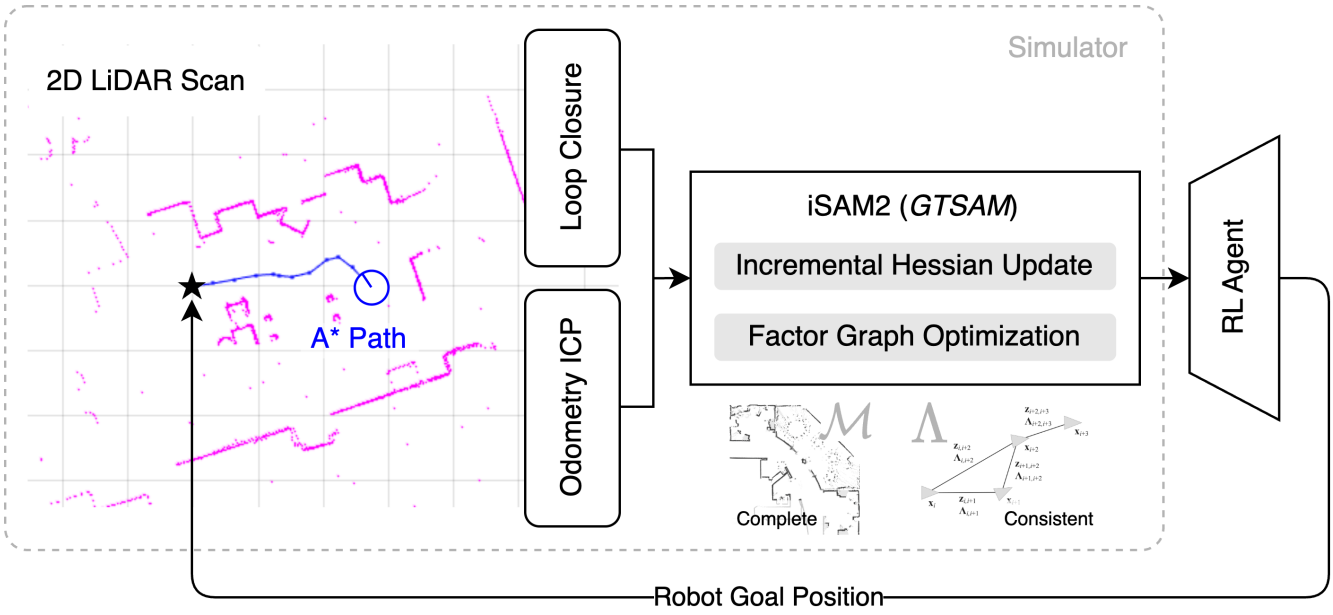


Fig. 1: Our proposed Active SLAM framework. A reinforcement learning agent learns to maximize map coverage as well as map consistency through a dual-objective reward incentivizing information gain. Nonlinear least squares residuals from off-the-shelf solvers like iSAM2 provide reward signals.

$$\mathcal{X} = \underset{\mathcal{X} \in SE(n)}{\arg\max}\, p(\mathcal{Z}|\mathcal{X}) = \underset{\mathcal{X} \in SE(n)}{\arg\min} \sum_{(i,j)} \|h(x_i, x_j) - z_{i,j}\|^2_\Sigma \tag{1}$$

where $\|\cdot\|^2_\Sigma$ represents the squared Mahalanobis distance, Σ is the covariance matrix corresponding to the measurement between poses i and j, and h is the measurement function. As noted earlier, measurements may come from odometry sources like wheel encoders or LiDAR-ICP, or they may be virtual measurements induced by loop closure detections.

In our work, we assume a planar robot operating in $SE(2)$ equipped with a 2D LiDAR which provides bearing and range measurements over a 360-degree field-of-view. Odometry factors come from the 2D Iterative Closest Point (ICP) algorithm, while loop closures candidates are periodically searched for by considering previous poses in the factor graph which are within a $4m$ radius of the latest pose and observe similar geometric features to those observed by the robot currently; once loop closure candidates have been determined, the 2D ICP algorithm is used once again to determine the approximate transformation between poses involved in a loop closure. The global poses of the robot drift over time due to error in odometry, while the addition of loop closure constraints seeks to rectify this drift by enforcing global consistency.

Modern SLAM systems solve this optimization problem over $\mathcal{X}$ in an online, incremental fashion using nonlinear least squares optimization techniques. One popular piece of open source software which implements various optimizers, including the incremental iSAM2 optimizer [7] used in this work, is GTSAM [8]. GTSAM is a performant C++ library with Python bindings that provides access to a variety of on-manifold nonlinear least squares solvers and is readily used throughout modern SLAM systems, hence it was chosen as the backend solver for our work.

For the optimization problem in equation (1), the associated Hessian or Information Matrix is defined as:

$$\Lambda = \sum_{(i,j)} J^T_{i,j} \Sigma^{-1} J_{i,j} \tag{2}$$

where $J_{i,j}$ is the Jacobian of the residual function $h(x_i, x_j) - z_{i,j}$ with respect to $\mathcal{X}$. The Hessian of the graph SLAM problem contains meaningful structure and sparsity, and the block patterns within are heavily related to the underlying graph connectivity as shown in [9]. Norms on the SLAM Hessian Λ serve as meaningful metrics on how well constrained the underlying pose graph is and will be leveraged in reward function design.

B. Formulating Active SLAM as an RL Problem

We cast Active SLAM as a reinforcement learning (RL) problem where an agent learns a policy that selects exploration targets to both map unknown space and reduce pose uncertainty through loop closures. In order to make the RL problem tractable and avoid reward

sparsity, we abstract the learning problem into a dual-level hierarchy of planning; at a higher level, we train an RL policy to output robot goal waypoints, while at a lower level, we make use of classical planners like A^* to route the robot to the desired goal point. We segment the policy in this way to allow the policy to more easily reason over long horizons, reducing the pressure to learn long-horizon reward structures which are often a core difficulty of episodic RL; in this way, our policy acts more like a contextual bandit, choosing either exploration or loop closure goals at each query step. We experiment with allowing the robot to choose a continuous (x, y) position goal versus a discrete choice of one of k randomly sampled position goals. The motivation of the latter choice is inspired by classical frontier-based exploration.

In our formulation, the agent receives observations s_t consisting of:

- The occupancy grid map induced from the scans collected at the pose graph nodes
- A one-hot encoded image with the agent's current position within the occupancy grid map
- The marginal information quantities associated with each pose in the pose graph projected on to the occupancy grid map

And, depending on whether the policy uses continuous waypoint actions or is choosing from one of k sampled goals, we include in the observation the vector of available waypoints as well. Upon selecting a frontier, the low-level planner routes the agent to the goal and integrates sensor information along the way to update the SLAM estimate and occupancy grid map.

The agent is rewarded based on a dual-objective reward function:

$$\mathcal{R} = \lambda_{\text{explore}} \left(\frac{1}{|\mathcal{M}|} \Delta H_t \right) + \lambda_{\text{closure}} \Delta \log(\det \Lambda_t) \quad (3)$$

The first term in the reward computes the change in entropy of the occupancy grid map $\mathcal{M}$ where $|\mathcal{M}|$ denotes the size of the map. The entropy of the occupancy grid map at time t is defined as:

$$H_t = -\sum_{\mathcal{M}} p_i \log(p_i) \quad p_i \sim \text{Bernoulli}(\mathcal{M}^i) \quad (4)$$

which is the sum of entropies of each occupancy grid map cell, where each cell $\mathcal{M}^i$ is distributed according to a Bernoulli random variable indicating occupancy probability.

The second term in the reward computes the change in the log-determinant of the Hessian matrix Λ between consecutive time steps. As informative loop closures are added to the pose graph, the hessian will accumulate more significant off-diagonal terms which heavily constrain the least squares solution and result in a consistent map.

Taken together, this dual-objective reward function can be viewed as incentivizing information gain in two modalities: information gain in the occupancy grid map via exploration and information gain in the pose graph via seeking loop closures. The relative weighting of these oftentimes contrasting objectives is controlled by $\lambda_{\text{explore}}, \lambda_{\text{closure}} \in \mathbb{R}$.

C. Learning an Active SLAM RL Agent

In our experiments, we employ standard RL algorithms including PPO [10] and DQN [11] to the Active SLAM problem as formulated earlier. The implementations for these algorithms are based on the popular Stable Baselines 3 references [12] to ensure consistency and correctness. The network architectures for actor and critic networks is based on a convolutional encoder network followed by an MLP.

The optimal Active SLAM action distribution is inherently multi-modal, as multiple decisions are often equally beneficial to the agent in a given environment state. Take, for example, a robot faced with the decision of going left or right when entering a hallway: both options may yield high exploration reward. This multi modality may be easily captured by the categorical probability distribution produced by discrete Q-learning frameworks, but multi-modal continuous action distributions are a challenging and active area of RL research. Although we were unable to explore the latter challenge in detail, we touch on potential solutions and related research in our discussions of future work.

III. Results

A. Preliminary Reward Function Verification

We first verify that the reward signal as formulated above indeed incentivizes informative loop closures and frontier exploration. To verify the former, we formulate a toy SLAM problem with nine poses in a figure-eight pattern where two locations are visited twice while varying the number of loop closures added to the optimization. We inspect the value of the log determinant of the Hessian and its block sparse structure in Figure 2 and clearly note the off-diagonal terms introduced by each additional loop closure as well as the increasing value of $\log(\det \Lambda)$. To verify the decreasing map entropy as the agent becomes more certain of a cells occupancy, we simply compute and plot the average occupancy grid cell entropy at various stages of map exploration.

B. Exploration-only Experiments

Using only the coverage term of the reward function combined with a time penalty term results in a learned policy which seeks to maximize map coverage as quickly as possible. We use this experiment setup to verify all aspects of our custom simulator, including the occupancy grid creation, updates, A^* planner, and

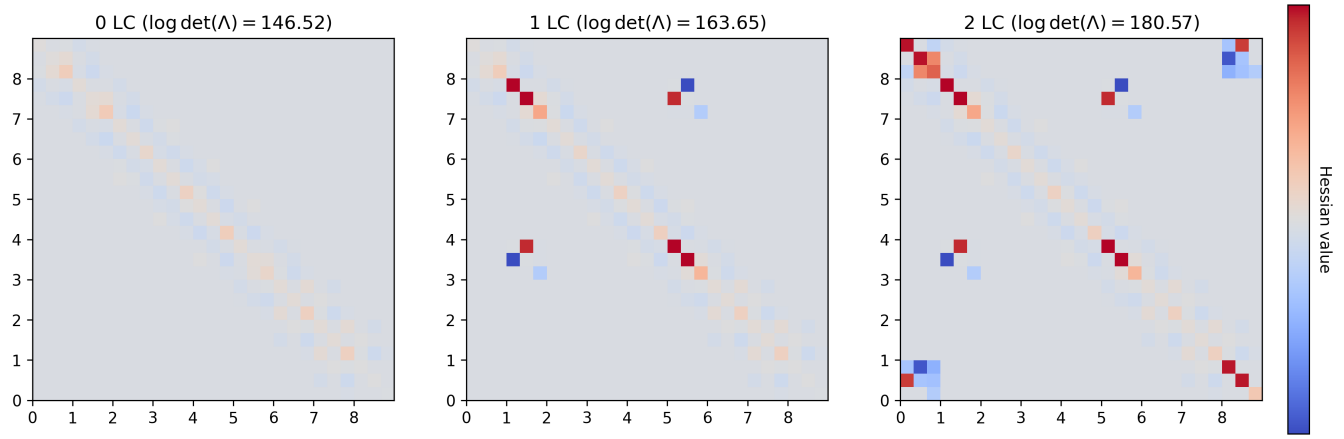


Fig. 2: Visualization of the SLAM Hessians Λ after graph optimization for a simple environment with nine poses in a figure-eight-like configuration with noisy odometry. We consider how 0, 1, and 2 loop closures included in the optimization problem affect the log determinant value and Hessian structure.

RL algorithm parameters. We first formulate the active exploration problem using a continuous action space using PPO, allowing the agent to choose the goal (x, y) coordinate freely; we only allow the agent to proceed if the chosen coordinate lies within the free space known to the agent at the current timestep to avoid planning in unknown space; otherwise, the agent does not move and incurs a negative reward. The agent is trained for $3 \cdot 10^6$ environment steps and achieves an average episodic reward of 13 and episode length of 7 by the end of training which is on-par with expert human performance [1]. An agent which chooses goal positions randomly achieves an average episodic reward of -12 and average episode length of 30, indicating our agent's performance is significantly better than random exploration. A sample query is shown in Figure 3.

While the continuous action formulation is simple and elegant, we seek to make the learning problem easier for the agent by using a discrete action space with DQN instead. This choice is motivated by the fact that DQN is often more sample efficient than PPO, easier to train, and can easily encode multi-modality in the action distribution when compared to the uni-modal Gaussian distributions often employed by PPO or SAC. To efficiently discretize the action space, we consider two methods:

1) Sample k frontiers between explored and unknown space, having the agent to choose one of these k as the goal position
2) Sample k cells in the free space, having the agent to choose one of these k as the goal position

The first is motivated by classical frontier-based exploration methods [2] while the latter is more computationally efficient, alleviating the need for frontier detection. Both of these methods, utilizing the same neural network backbone as the continuous action formulation, result in similar performance to PPO in only $3 \cdot 10^5$ environment steps.

[1]The human experts are the authors of this project who know the map geometry a-priori and can efficiently choose goal positions based on promising visual frontiers and features.

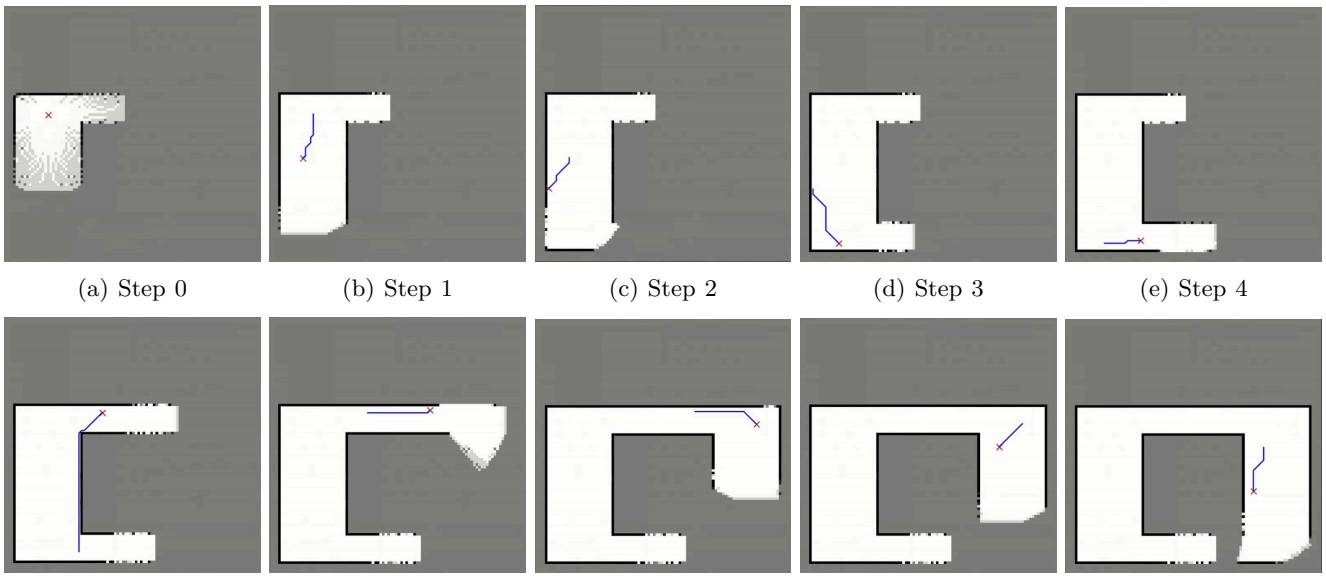
C. Exploration and Loop Closure Experiments

To assess the feasibility of learning policies that jointly promote exploration and loop closure, we trained agents using the full reward signal defined in Equation (3), which combines occupancy map entropy reduction with improvements in pose graph consistency via the SLAM Hessian log-determinant. In this setting, the agent must not only expand the known free space but also revisit previously observed regions to reduce global drift through loop closures.

We initially used Proximal Policy Optimization (PPO) with a convolutional encoder over the occupancy map, agent position, and marginal information channels. However, training with this setup yielded unstable results: the learned policies did not exhibit meaningful loop closure behavior. While PPO was effective in the pure exploration setting, our use of a unimodal Gaussian action distribution within PPO may have hindered learning in this more complex, multimodal setting with sparse loop closure rewards, as it limited the agent's ability to represent multiple equally viable actions.

These difficulties suggest two key limitations of the current approach. First, the marginal information map may be too coarse or indirect a signal to capture the full structure of pose graph constrainedness, particularly in ambiguous or symmetric environments. Second, unimodal policies may struggle to represent multiple equally promising behaviors, such as revisiting different loop closure candidates or continuing frontier exploration.

To address these issues, we propose two future directions: (1) incorporating generative policy architectures—such as energy-based normalizing flows or diffusion-based



(a) Step 0 (b) Step 1 (c) Step 2 (d) Step 3 (e) Step 4

(f) Step 5 (g) Step 6 (h) Step 7 (i) Step 8 (j) Step 9

Fig. 3: A ten-step query demonstrating agent exploration in a simulated 2D environment, whereby red crosses demonstrate the policy-dictated high level actions and low-level planning-based trajectories are shown in blue.

actors—which naturally model multimodal action distributions, and (2) designing more expressive encodings of pose graph structure, such as using graph neural networks or spectral features of the SLAM information matrix. We hypothesize that these changes will improve both training stability and the agent's ability to reason about long-horizon, uncertainty-reducing behaviors.

IV. Discussion and Future Work

Our results demonstrate the feasibility of learning high-level exploration strategies in Active SLAM via deep reinforcement learning. In the exploration-only setting, both continuous and discrete action formulations enabled agents to efficiently traverse and map simulated environments. However, when loop closure incentives were introduced, policies trained with PPO struggled to capture long-horizon uncertainty-reducing behaviors, underscoring several key challenges and future research directions.

One core limitation is the use of a unimodal Gaussian policy for continuous action selection. Active SLAM naturally induces multi-modal action distributions, especially in symmetric or ambiguous environments where multiple frontier or loop closure options exist. To this end, we believe that generative policy architectures—such as diffusion-based policies [5], some of which draw inspiration from entropy-regularized reinforcement learning frameworks like SAC—offer a promising solution. These approaches model expressive, multi-modal action distributions and could more effectively represent the diversity of valid strategies in exploration and relocalization. Future work will explore training such generative policies in an online RL setting tailored for Active SLAM.

Another significant challenge arises from simultaneously learning to explore and close loops. These objectives can conflict, and balancing them requires both a well-shaped reward function and an expressive policy. While our dual-objective reward captures exploration (via entropy reduction) and consistency (via Hessian log-determinant), the observation space—especially the marginal information heatmap—may lack sufficient fidelity to guide nuanced decisions. More structured representations of the SLAM graph, such as learned embeddings using graph neural networks or spectral descriptors of the information matrix, could provide richer signals for decision making. Another promising direction is to incorporate temporally extended behaviors via the Option-Critic architecture [13]. The task presents a fundamentally multi-objective decision space and these objectives often require distinct behavioral patterns over long time horizons. The Option-Critic framework provides a natural mechanism for learning such high-level behaviors—called options—end-to-end, along with their internal policies and termination conditions. In our context, this could result in the emergence of specialized options for frontier exploration versus loop closure, enabling the agent to dynamically switch between them based on environmental context and map structure. We believe this hierarchical control structure could significantly improve long-horizon reasoning in our setting.

Our simulation environment, while powerful, is computationally intensive. Each episode requires SLAM optimization, occupancy grid updates, and low-level planning. As a result, sample efficiency becomes critical. While PPO served as a strong baseline, off-policy

algorithms such as Soft Actor-Critic (SAC) [14] may be better suited due to their improved sample reuse and entropy-regularized stochastic policies that better encourage exploration.

Lastly, our current architecture assumes a fixed-size map and observation space, limiting its scalability to large or open-ended environments. One exciting direction is to adopt transformer-based policies with memory or autoregressive representations that can reason over dynamically sized graphs and observations. This would enable deployment in more realistic, unbounded domains while retaining long-horizon reasoning over the evolving map.

References

[1] C. Stachniss, G. Grisetti, and W. Burgard, "Information gain-based exploration using rao-blackwellized particle filters," in Robotics: Science and Systems (RSS), 2005.

[2] S. Baek, B. Moon, S. Kim, M. Cao, C. Ho, S. Scherer, and J. H. Jeon, "Pipe planner: Pathwise information gain with map predictions for indoor robot exploration," arXiv preprint arXiv:2503.07504, 2025. [Online]. Available: https://arxiv.org/abs/2503.07504

[3] J. A. Placed and J. A. Castellanos, "A deep reinforcement learning approach for active slam," Applied Sciences, vol. 10, no. 23, p. 8386, 2020.

[4] D. S. Chaplot, D. Gandhi, S. Gupta, A. Gupta, and R. Salakhutdinov, "Learning to explore using active neural slam," in International Conference on Learning Representations (ICLR), 2020.

[5] H. Ma, T. Chen, K. Wang, N. Li, and B. Dai, "Soft diffusion actor-critic: Efficient online reinforcement learning for diffusion policy," arXiv preprint arXiv:2502.00361, 2025.

[6] C.-H. Chao, C. Feng, W.-F. Sun, C.-K. Lee, S. See, and C.-Y. Lee, "Maximum entropy reinforcement learning via energy-based normalizing flow," arXiv preprint arXiv:2405.13629, 2024.

[7] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. Leonard, and F. Dellaert, "isam2: Incremental smoothing and mapping with fluid relinearization and incremental variable reordering," in 2011 IEEE International Conference on Robotics and Automation, 2011, pp. 3281–3288.

[8] F. Dellaert and G. Contributors, "borglab/gtsam," May 2022. [Online]. Available: https://github.com/borglab/gtsam

[9] J. A. Placed and J. A. Castellanos, "A general relationship between optimality criteria and connectivity indices for active graph-slam," IEEE Robotics and Automation Letters, vol. 8, no. 2, pp. 816–823, 2023.

[10] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," arXiv preprint arXiv:1707.06347, 2017. [Online]. Available: https://arxiv.org/abs/1707.06347

[11] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing atari with deep reinforcement learning," arXiv preprint arXiv:1312.5602, 2013. [Online]. Available: https://arxiv.org/abs/1312.5602

[12] A. Raffin, A. Hill, A. Gleave, A. Kanervisto, M. Ernestus, and N. Dormann, "Stable-baselines3: Reliable reinforcement learning implementations," Journal of Machine Learning Research, vol. 22, no. 268, pp. 1–8, 2021. [Online]. Available: http://jmlr.org/papers/v22/20-1364.html

[13] P.-L. Bacon, J. Harb, and D. Precup, "The option-critic architecture," in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 31, no. 1, 2017.

[14] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in International Conference on Machine Learning (ICML). PMLR, 2018, pp. 1861–1870.