

MapExRL: Human-Inspired Indoor Exploration with Predicted Environment Context and Reinforcement Learning

Narek Harutyunyan^{*1}, Brady Moon^{*2}, Seungchan Kim², Cherie Ho², Adam Hung³, Sebastian Scherer²

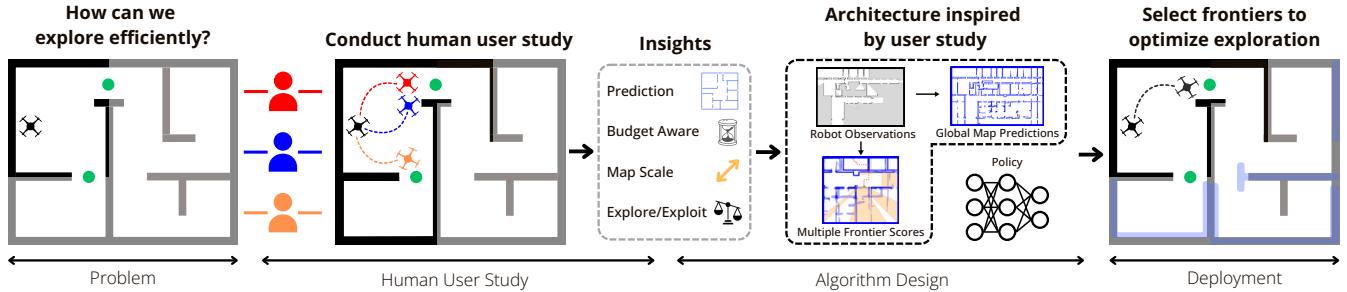


Fig. 1. An overview of the development and research process for MapExRL. This work examines the question: How can a robot explore efficiently? We conduct a human user study to gain insights into effective exploration strategies. These insights, in turn, inform the design of our learning-based exploration policy, leveraging global map predictions and other environmental contexts and enabling state-of-the-art performance.

Abstract—Path planning for robotic exploration is challenging, requiring reasoning over unknown spaces and anticipating future observations. Efficient exploration requires selecting budget-constrained paths that maximize information gain. Despite advances in autonomous exploration, existing algorithms still fall short of human performance, particularly in structured environments where predictive cues exist but are underutilized. Guided by insights from our user study, we introduce MapExRL, which improves robot exploration efficiency in structured indoor environments by enabling longer-horizon planning through a learned policy and global map predictions. Unlike many learning-based exploration methods that use motion primitives as the action space, our approach leverages frontiers for more efficient model learning and longer horizon reasoning. Our framework generates global map predictions from the observed map, which our policy utilizes, along with the prediction uncertainty, estimated sensor coverage, frontier distance, and remaining distance budget, to assess the strategic long-term value of frontiers. By leveraging multiple frontier scoring methods and additional context, our policy makes more informed decisions at each stage of the exploration. We evaluate our framework on a real-world indoor map dataset, achieving up to an 18.8% improvement over the strongest state-of-the-art baseline, with even greater gains compared to conventional frontier-based algorithms. Website: mapexrl.github.io

I. INTRODUCTION

Robot exploration is a long-studied problem in robotics due to its wide applications, including search and rescue, reconnaissance, and information gathering. In exploration, a robot navigates in an unknown space to build map representations and understand the environment. At its core, it is the problem of deciding where to go to gain the most information [1]–[3]. A conventional paradigm of deciding where to go is frontier-based exploration [1], in which a robot incrementally updates the boundary between known and unknown spaces, or *frontier*, and systematically visits the frontiers. While this approach has proven to be robust in large-scale environments [3], it heavily relies on simple heuristics, such as visiting the nearest frontiers first [1], or hand-crafted heuristics [4]. However, such algorithms are often short-sighted, reasoning only on the next step. In contrast, humans excel at crafting long-term strategies even with limited observations, that balance efficiency, uncertainty, and resource constraints. *How can we achieve human-level long-horizon exploration?*

To extend the perception horizon in robot exploration, some studies have proposed methods that go beyond reasoning about only the observed areas. These approaches use observed data to predict regions beyond what has been seen [5]–[7]. By leveraging deep learning models, these works generate broader map predictions from partial observations of the environment, improving exploration strategies. However, while these works augment perception by employing predictions, their decision-making remains focused on greedy and myopic action selection.

Reinforcement learning (RL) has been used to train robots to learn navigation policies through trial and error [8], [9]. Previous works have shown that data-driven RL can outperform classical planning approaches [10], [11]. However, these RL-based methods often rely on motion-primitives

^{*} Equal Contributions

This work was supported by the National Science Foundation Graduate Research Fellowship under Grant No. DGE1745016, a hardware grant from Nvidia, and the Office of Naval Research under Grant No. N00014-21-1-2110 and Contract No. N6833522C0179.

This work involved human subjects or animals in its research. The authors and CMU's IRB under Application No. STUDY2024_00000372 confirm that all human/animal subject research procedures and protocols are exempt from review board approval under the 2018 Common Rule 45 CFR 46.104.d.

¹Author is with the Brown University, Providence, RI, USA
narek.harutyunyan@brown.edu

²Authors are with the Robotics Institute, School of Computer Science at Carnegie Mellon University, Pittsburgh, PA, USA {bradm, seungch2, cherieh, basti}@andrew.cmu.edu

³Author is with the University of Michigan, Ann Arbor, MI, USA adamhung@umich.edu

for action spaces, which could present challenges for long-horizon planning in complex, large-scale environments.

Existing approaches remain limited in achieving long-term human-level decision making for exploration. Toward human-level exploration, we identify two key gaps: (1) an insufficient understanding of *what* signals humans use to make long-horizon decisions and *how* they leverage them, and (2) the lack of a planning framework that integrates these human-inspired insights into robotic exploration.

To bridge these gaps, we conduct a user study comparing human performance to a state-of-the-art (SOTA) exploration algorithm [7] and derive insights to guide our approach. Based on these insights, we introduce MapExRL, a *budget-aware* RL-based exploration framework that, like humans, leverages predictive context about the environment. We validate MapExRL on real-world floorplan data, evaluated against baseline methods and human users. An overview of this work is shown in Fig. 1.

Our contributions are as follows:

- We present insights on human performance for indoor exploration through a user study that illustrates the gap in performance to existing methods and extract actionable insights on how humans explore.
- We develop a new RL-based exploration planner, MapExRL, that leverages the insights from our user study on observation space features and action spaces.
- Finally, we conduct thorough comparisons of SOTA frontier-based and learning-based methods, as well as human performance, to show the benefits of MapExRL.

II. RELATED WORK

A. Robot Exploration

Autonomous robot exploration has been studied from multiple perspectives. Frontier-based exploration [1], [3] guides robots toward the boundary between known and unknown space but often suffers from slow progress and a lack of global planning. Information-theoretic approaches seek to maximize the information gain of consecutive actions [12], [13], yet they are often constrained by their greedy and myopic nature. Sampling-based methods [14] construct a randomly generated tree or graph structure, such as [15], to expand into a traversable space, typically selecting the most informative nodes or viewpoints for exploration.

B. Map Prediction for Exploration

A common limitation of the previously mentioned approaches is their reliance on sensor data with a limited perceptual range, often failing to reason beyond the observed areas. Recently, a growing body of work has explored leveraging deep learning models to predict maps and use these predictions to enhance exploration [5]–[7], [16]–[21]. These approaches utilize accumulated observations in a 2D top-down view as input and infer unexplored areas using predictive models trained on offline datasets. Using these predictions, robots can prioritize areas with high uncertainty [5], areas with high expected coverage gain [6], [16], [17], [21], or areas with both [7].

These approaches allow robots to reason beyond observed areas and focus on exploring areas guided by predictions; however, they still struggle due to their action selection strategy, particularly when handling environments of varying sizes and topological complexities or operating under budget constraints. In this work, we aim to address these limitations.

C. Learning-based Exploration

Studies have addressed exploration problems using deep reinforcement learning [8]–[11], [22]–[24]. Some have focused on training policies that generate simple, primitive actions, such as moving to adjacent cells or taking a few steps in a diagonally connected grid [9], [23]. However, this approach often results in myopic decision-making, prioritizing immediate actions over long-term strategy.

The work in [8] used an RL method that outputs a single weight to balance frontier scoring based on distance and estimated information gain, selecting the highest-scoring frontier. While this approach enables exploration of large, complex topologies, it does not directly score frontiers or exploit the predictability of indoor structures. Some studies have employed graph structures to encode the states, defining actions as movements toward nearby nodes within the graphs [10], [11], [22]. In our work, we use frontier points as the action space and score them directly, allowing us to consider the full frontier context and avoid myopic decision-making.

Lastly, some studies have integrated the map prediction paradigm into RL-based exploration to expand the observation space [20], [25]. However, these studies commonly have limitations in that they set fixed primitive motions as the action space or validate their methods only on small, simplified maps. In our work, we expand the observation space using map prediction while simultaneously training a policy that outputs longer-horizon decision-making.

Our method, MapExRL, differs from previous approaches in three key ways: (1) it expands the observation space beyond direct observations using map predictions, (2) it leverages RL-based methods to enhance adaptability to various human-inspired temporal and spatial contexts, and (3) it sets frontiers as the action space, enabling longer-horizon action selection compared to primitive motions.

III. PROBLEM STATEMENT

We address the problem of efficient indoor mapping and exploration. Our simulation environment is a 2D space $\mathcal{E} \subset \mathbb{R}^2$ where the robot's state is represented as a discrete 2D position $\mathbf{x} = [x, y]$. The robot is equipped with a 2D LiDAR that provides 360° coverage to construct an occupancy grid map O_t (observed map), categorizing the space as free, unknown, or occupied. At each timestep t , the robot moves one step Δx . We assume that the range measurements are noise-free and the robot pose is known. We aim to plan paths that maximize environment understanding as quickly as possible within a distance budget of B .

We define the predicted map's Intersection over Union (IoU) as a measure of exploration quality. The exploration

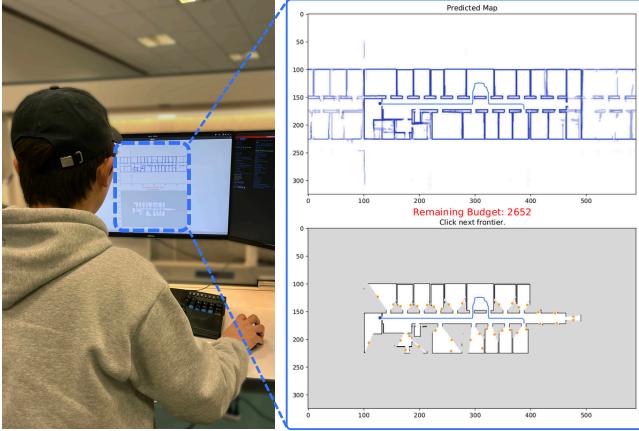


Fig. 2. User interface for the study: Participants are shown the observed occupancy grid (bottom) and predicted maps (top), and must select the next frontier to explore among the frontiers marked as orange dots. The goal is to reach 95% IoU as efficiently as possible within the distance budget B .

continues until the distance budget B is reached or a sufficient understanding of the environment is achieved, which we define as 95% IoU for the predicted map.

IV. USER STUDY ON INDOOR EXPLORATION

We conducted a user study to uncover key human exploration strategies and compare the performance of a state-of-the-art (SOTA) algorithm MapEx [7] with human decision-making in exploring large indoor buildings. This section covers the design and results of the user study and how we leveraged its insights to inform the design of our approach. The study aims to answer the following questions:

- Q1:** What is the performance gap between human decision-making and the SOTA exploration method, MapEx [7]?
- Q2:** How much does prior exposure to the map improve human exploration efficiency?
- Q3:** What key context do humans use during exploration?
- Q4:** What long-term strategies do humans use to explore?

A. User Study Design

To answer the above questions, we conducted a user study with 13 participants of varying experience levels in robotic exploration. To ensure a fair evaluation, participants were not allowed to see the maps beforehand, preventing any advantage from prior knowledge or mental priors. This ensured that their performance was based solely on their ability to explore the environment without prior familiarity. Users are given the same problem formulation as defined in Section III—either achieve 95% IoU on the predicted map as quickly as possible or maximize IoU within the distance budget. The interface (Fig. 2) displays the average output of the ensemble global map prediction network (same as [7]) and the accumulated observed occupancy grid map O_t . Frontiers are displayed on the observed map, and the user chooses which frontier to explore next. The robot then moves to the selected frontier, updating the maps after each robot step. This process repeats until the episode ends.

Each participant first completes a training round to familiarize themselves with the interface and task. They then



Fig. 3. The set of test maps used in this work. These maps were selected to ensure a diverse range of common floorplan layouts and topologies. Note that the maps are not shown to scale relative to one another, and Map 5 is the largest in actual size, followed by Map 4.

complete nine test episodes across three different maps, completing three episodes per map before moving to the next. We selected maps 1, 4, and 5 (Fig. 3) to ensure a range of map sizes and shapes. Successive episodes on the same map are referred to as rounds 1, 2, and 3. In rounds 1 and 2, participants start from the same position, while round 3 uses a different starting location. Round 1 assesses performance without prior knowledge of the map scale or features. Round 2 evaluates improvements based on prior knowledge and the opportunity to adapt their previous strategy. Round 3 examines how strategies adapt to new starting conditions while retaining prior map knowledge. Multiple runs per map also help capture participants' best performances. To introduce variety while maintaining comparability, we changed the set of starting positions for every three participants. Testing took approximately 30 minutes, and participants could withdraw at any point to prevent fatigue-related bias.

B. Quantitative Results

The user study results are summarized in Table I, showing the average reward and 95% confidence interval. The reward was calculated as $R = \text{IoU} \times 1000 + B_r$, where R is the reward and B_r is the remaining budget at the end of the episode. MapEx results are divided based on the changes in starting positions between rounds 2 and 3. Notably, MapEx has large confidence intervals due to its deterministic nature; unlike human participants, it could not repeat runs from the same starting location.

The combined scores in the rightmost column of Table I show that in round 1, even without prior knowledge of the map, participants already outperform the SOTA baseline. Examining the scores more closely, participants surpassed the baseline across all maps and rounds, except for Map 2 in round 2 and Map 4 in round 3. Notably, user performance on Map 2 did not improve between rounds 1 and 2. Additionally, the confidence interval for this map was significantly larger than for the others. This discrepancy may stem from the difficulty of discovering the bottom left region, which was easily overlooked. Many participants missed these rooms and

	Map 1	Map 2	Map 4	Combined
Human Round 1	1156 ± 32	902 ± 78	768 ± 37	911 ± 60
Human Round 2	1160 ± 39	885 ± 67	795 ± 32	915 ± 56
MapEx Round 1&2	1071 ± 770	888 ± 446	731 ± 103	875 ± 146
Human Round 3	1189 ± 35	941 ± 61	814 ± 50	951 ± 57
MapEx Round 3	1037 ± 1036	899 ± 467	875 ± 215	924 ± 116

TABLE I. Results of the user study compared to MapEx [7]. Values reflect the average final reward (higher is better) and the 95% confidence interval.

continued searching elsewhere, while those who found the region achieved much higher rewards. This led to greater variance in the results, highlighting the need for reasoning beyond directly observed areas.

Generally, we observe a performance gap between humans and the SOTA exploration method. While this gap is small in round 1, it generally increases across rounds, indicating that exposure to the maps aids in exploration. Notably, performance varied among participants. Most were graduate students in robotics or related fields, with some conducting planning research and field tests using frontier-based planning methods. These participants consistently outperformed others across all maps and rounds. One participant with extensive hands-on systems experience with robot exploration scored 907, 1220, and 1089 in round 1, and 873, 1260, and 1056 in round 3—significantly higher than peers or the baseline. This suggests that certain strategies or heuristics are more effective and that general planning experience may be beneficial. These higher scores further highlight the disparity between SOTA performance and expert human behavior.

C. Qualitative Results and Insights

Observing the participants’ strategies, those who performed best focused on maximizing prediction IoU rather than defaulting to maximizing map coverage. Some took unnecessary actions, such as searching rooms that already had clear predictions. Users often adjusted their strategies based on the map’s scale—exploring rooms more in smaller maps and focusing on main hallways in larger ones. Participants also aimed to minimize backtracking, describing different approaches of thinking a few steps ahead or considering map topology for efficient traversal. The benefit of an action was often estimated by imagining the observations at the frontier. Most ignored the remaining budget until later in the episodes, when they began estimating the cost of reaching frontiers with limited steps left. Many were surprised at the task’s difficulty and reported feeling fatigued by the end.

Fig. 4 shows the paths chosen by three users in round 1 compared to MapEx. MapEx concentrated its exploration budget in a limited region, resulting in insufficient exploration of the entire space. In contrast, the human users distributed their exploration more evenly, despite having no prior knowledge of the environment. We observed that between rounds 1 and 2, where the start position remained the same, users realized the overall size and shape of the space and adjusted their strategies to explore more evenly.

From the quantitative and qualitative results from the user experiments, we found the following actionable insights:



Fig. 4. Three user runs and MapEx on Map 4, with colors differentiating the separate users and the red dot marking the starting position. Users explored more evenly across the space, leading to higher final reward.

- 1) Leveraging the global map predictions can likely benefit exploration as it allows for maximization of prediction accuracy through prioritizing observing uncertain areas in the predictions instead of exploring exhaustively.
- 2) The value of an action relates to the balance of exploitation and exploration while also considering both area coverage and prediction uncertainty.
- 3) Strategies must adapt to the map scale relative to the budget, as a single heuristic cannot perform effectively across all map sizes.
- 4) An ideal policy has budget awareness throughout the entire planning process to minimize backtracking. Thinking more than one step ahead helps avoid backtracking and creates more efficient paths.

V. APPROACH

Based on our user study, we propose a Reinforcement Learning (RL) pipeline using the Soft Actor-Critic (SAC) algorithm [26], leveraging frontiers as an efficient action space. Our policy input incorporates global map predictions for global context and calculates predicted frontier information gain. This approach enables the model to integrate multiple scoring methods (user insights 1 and 2), global context (user insight 3), and remaining budget (user insight 4) to select actions more effectively than heuristic-based methods. Fig. 5 provides an overview. For implementation, we used Stable Baselines 3 [27], with SAC consisting of two fully connected layers for both actor and critic.

A. Map Encoder

Building on user insight 1, we use the map prediction model [28], which predicts the global map based on the accumulated occupancy map from LiDAR observations. Prior work [7] has shown its effectiveness in estimating global topology beyond the robot’s current observations. Following [5], [7], we model the prediction uncertainty by generating multiple independent predictions from an ensemble of networks and calculating the variance of their outputs. This encourages a robot to prioritize uncertain areas.

Using the average of the ensemble of predicted maps, we generate a top-down map centered on the robot’s position, applying padding and cropping to produce a 1600x1600

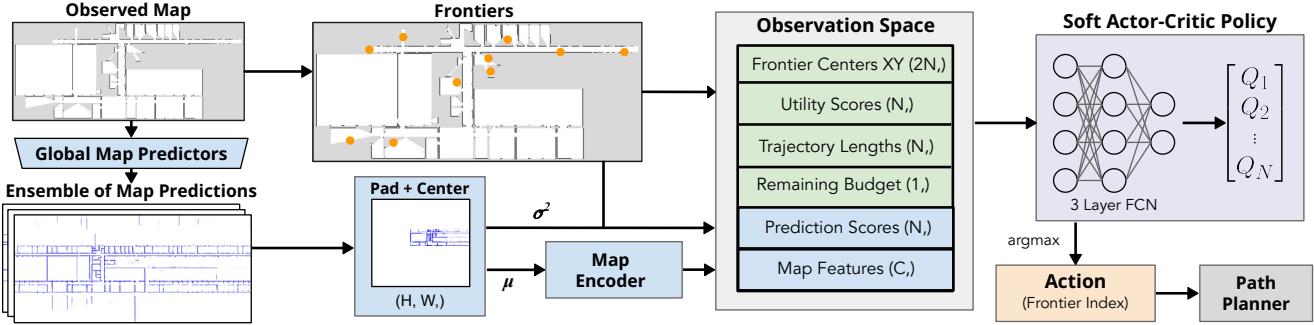


Fig. 5. The MapExRL pipeline, where observed maps are processed through three independent global map prediction models, generating prediction maps. These maps are averaged and passed through a convolutional encoder to extract a 256-dimensional feature vector. This vector is concatenated with frontier centers, prediction and utility scores, distances from the agent, and remaining budget. The resulting vector is fed into a fully connected network that outputs N values, and the argmax is selected as the index of the frontier action.

pixel map. This map is then resized to 256x256 pixels using interpolation, followed by max pooling to obtain a 128x128 pixel input for the encoder. These operations ensure consistency across maps of varying shapes and sizes, following user insight 3—no single heuristic works optimally in all scenarios, and strategies must adjust to different map scales.

The 128x128 pixel map is processed by an encoder with three convolutional layers and a fully connected layer, producing a $C=256$ dimensional latent vector. Separate map encoders are used for the actor and critic.

B. Observation and Action Spaces

When a new action needs to be chosen, frontiers are calculated from the current occupancy map. In order to have a fixed action space size, this set of frontiers must be reduced to a fixed number N . We choose the top N frontiers based on their prediction scores, excluding those outside of the robot-centered map. In cases when there are fewer than N frontier centers, we add zero padding for the remaining. In our implementation, we select $N = 10$ frontiers based on experimental results, as they provide sufficient exploration options in all directions across training and testing maps.

To ensure the selected frontiers are unique and well-distributed across the map, we compare the prediction and utility scores of frontier points within a 5-meter distance threshold. If the score differences are below an experimentally determined threshold (0.01) and the points are spatially close, we remove duplicates to maintain diversity in the observation space.

The observation space includes the 256 encoded map features, $[x, y]$ coordinates of the N frontiers, utility and prediction scores for these frontiers, trajectory length from the robot's current position to each frontier, and the remaining distance budget. Each frontier center is defined relative to the robot's position to align with the map transform.

The frontier features are calculated as follows:

- Frontier Centers: convolve an edge detection kernel across the observed map, group and filter to be above a minimum frontier edge size (following [1]), find the center, translate into a robot-centered convention, and normalize by half the robot-centered map width.

- Utility Scores: raycast and create visibility mask in the occupancy map, sum the number of unknown cells in the visibility mask, normalize by the A* distance to the frontier, and apply min-max normalization.
- Prediction Scores: probabilistic raycast in the mean predicted map, sum the variance of the cells in the visibility mask, normalize by the A* distance (rather than Euclidean as in [7]), and apply min-max normalization.
- Trajectory Lengths: A* distances to the frontiers from the robot's current position, normalized by B
- Remaining Budget B_r : the remaining mission budget, normalized by the total budget B .

The policy network outputs a N -dimensional vector (action space), where each value represents one of the top N frontier options. The frontier corresponding to the highest value in this vector is selected as the next action.

C. Reward Function

To achieve 95% IoU as efficiently as possible, our reward function considers both the final episode IoU and the remaining budget. We designed the reward function to provide feedback only at the end of the episode, based on the accuracy of the predicted map compared to the ground truth. This fully sparse reward is feasible given our efficient action space, which inherently limits the number of steps per episode. The reward R is defined as

$$R = \begin{cases} \max(0, \text{IoU} - \delta_{\text{IoU}}) \times 1000 + B_r, & B_r = 0 \text{ or } \text{IoU} \geq 0.95 \\ 0, & \text{otherwise} \end{cases}$$

where B_r represents the remaining mission time or budget, while $\delta_{\text{IoU}} = 0.4$ clips the IoU to create a more evenly distributed reward signal and strengthen the feedback to the network. With nearly all training episodes achieving an IoU above 0.4, this clipping significantly improved training performance. This reward function encourages efficient exploration by rewarding the model for reaching 95% IoU while minimizing total distance usage.

We use the Intersection over Union (IoU) of the occupied cells for the global map prediction as our metric of performance. IoU is computed as $\text{IoU} = \frac{\text{TP}}{\text{FP} + \text{TN} + \text{TP}}$, where TP, FP, TN denote true positives, false positives, and

Method	Map 1		Map 2		Map 3		Map 4		Map 5		Combined	
	Reward	IoU										
MapExRL (ours)	726 ± 50	0.96 ± 0.01	465 ± 91	0.84 ± 0.07	722 ± 41	0.96 ± 0.00	370 ± 40	0.77 ± 0.04	329 ± 27	0.73 ± 0.03	522 ± 46	0.85 ± 0.03
MapEx [7]	715 ± 40	0.96 ± 0.00	404 ± 90	0.79 ± 0.07	695 ± 52	0.95 ± 0.00	397 ± 36	0.80 ± 0.04	277 ± 40	0.68 ± 0.04	498 ± 48	0.83 ± 0.03
MapExRL No Pred	723 ± 69	0.95 ± 0.01	386 ± 94	0.77 ± 0.07	677 ± 49	0.95 ± 0.00	368 ± 31	0.76 ± 0.03	314 ± 38	0.71 ± 0.03	494 ± 47	0.83 ± 0.02
Random Policy	562 ± 88	0.90 ± 0.04	384 ± 57	0.78 ± 0.06	577 ± 52	0.94 ± 0.01	266 ± 23	0.67 ± 0.02	232 ± 32	0.63 ± 0.03	401 ± 41	0.78 ± 0.03
Nearest Frontier	482 ± 15	0.88 ± 0.02	310 ± 33	0.65 ± 0.08	561 ± 67	0.90 ± 0.04	232 ± 27	0.63 ± 0.03	94 ± 47	0.49 ± 0.05	336 ± 44	0.71 ± 0.04
RL Motion Primitive	57 ± 36	0.45 ± 0.04	16 ± 23	0.33 ± 0.06	77 ± 40	0.45 ± 0.06	0 ± 0	0.29 ± 0.05	0 ± 0	0.20 ± 0.07	30 ± 13	0.34 ± 0.03

TABLE II. Average and 95% confidence interval for reward and IoU on the test maps. Each map includes 15 experiments from different starting positions, totaling 75 experiments across 5 maps.

true negatives, respectively. To avoid penalizing minor map discrepancies—like wall thickness variations, gaps, dataset artifacts, or small unobservable spaces—we use binary dilations as a buffer in IoU calculation.

Let P represent the predicted binary map of occupied cells, G represent the ground truth occupied binary map, and O represent the observed map. We define the masked prediction P_m such that $P_m = P$ where $O = 0.5$ (unknown space) and $P_m = O$ otherwise, where only unknown space in the observed map is replaced by the global map prediction. We then compute a corrected prediction P_c from P_m by excluding all predictions that fall outside the building. We dilate P_m and G by a fixed kernel size of 5 pixels (50 cm) to create P_d and G_d , respectively. By evaluating all pixels i , we can then calculate $\text{TP} = \sum_i (P_d(i) \wedge G(i))$, $\text{FP} = \sum_i (P_m(i) \wedge P_c(i) \wedge \neg G_d(i))$, and $\text{FN} = \sum_i (\neg P_d(i) \wedge G(i))$, where \wedge and \neg denote the logical AND and NOT operators, respectively. This process allows the IoU to focus on overall map topology and wall placement. Without this correction, IoU plateaus at 75-85%, even with full area coverage.

D. Overall Exploration Cycle

At the start of each episode, the robot continuously builds a 2D occupancy map with the incoming LiDAR observations. The observed map is then fed into 3 independent global map prediction models, each trained on different subsets of the training maps, to produce an average map and variance map for frontier’s prediction score calculations. After selecting the top N frontiers, the frontier features and the mean map are computed and subsequently provided to the policy model, which determines the frontier for the robot to explore. An A* path from the current location to the frontier center is found, and the robot advances until the goal is reached or is deemed unreachable. This iterative process persists until the mission budget B or the robot successfully predicts the global occupancy map with 95% IoU.

E. Simulation Details

We evaluate MapExRL on the real-world KTH floorplan dataset [29]. The dataset consists of 149 campus floorplan blueprints from the KTH Royal Institute of Technology, which detail wall and door positions. We use 7 maps for training and 5 maps (shown in Fig. 3) for testing MapExRL, selecting maps that are medium to large in size, have complex topologies, and were not in the training set for the global map predictor model. We simulate a robot operating in the building with a LiDAR sensor that has 2500 scanning beams at a 20-meter range. Our maps are scaled such that one pixel is equal to 10 cm.

F. Baselines

Our first baseline is the conventional frontier-based exploration method [1], where the agent chooses the nearest frontier based on Euclidean distance, without utilizing any global map predictions. Our second baseline, MapEx [7], selects the frontier with the highest information gain, calculated by jointly considering sensor coverage and variance from predicted maps, normalized by the Euclidean distance. We also include a random exploration baseline, where the agent randomly selects a frontier from the top N frontiers in the observation space. This baseline validates that our model learns to navigate efficiently using the top N frontiers rather than randomly picking from the list.

Additionally, we compare against a baseline inspired by [20], which uses RL with motion primitives as actions. We adopt their action spaces and reward functions to train the policy while employing our global map prediction module. The action space consists of 8 primitive motions ($\mathcal{A} = \nwarrow, \uparrow, \nearrow, \leftarrow, \rightarrow, \swarrow, \downarrow, \searrow$), each with a constant step size Δx , uniform across all methods to ensure consistency with our approach. The reward function heavily penalizes actions that result in collisions with occupied cells and rewards actions that increase the number of newly observed cells. For the details of the reward function, refer to Equation 4 of [20]. This baseline helps validate the impact of using frontier points as action spaces instead of primitive motions. We used the PPO algorithm to train the policy, following the approach in [20], and trained the model with the same number of steps to ensure a fair comparison with our method.

Lastly, we compared our model with an ablation model, referred to as MapExRL No Pred, which excludes global map predictions from its observation space. If our model outperforms this ablation, it validates user insight 1, confirming the importance of incorporating global map predictions.

G. Training Details

During training, we set a maximum mission budget of $B = 500$, which is equivalent to 300 meters. The episode ends when the predicted map reaches 95% IoU or the robot exhausts the mission time. For comparison across all methods, we calculate IoU for each step using the predicted map, including for baselines that do not incorporate global predictions for frontier scoring.

The discount factor for SAC is $\gamma = 0.99$. The batch size is set to 256, the replay buffer capacity is 10,000, and training begins once the replay buffer exceeds 1,000 data points. Training steps is set to 4, τ is set to 0.02, and training frequency is set to 1. The Adam optimizer is used

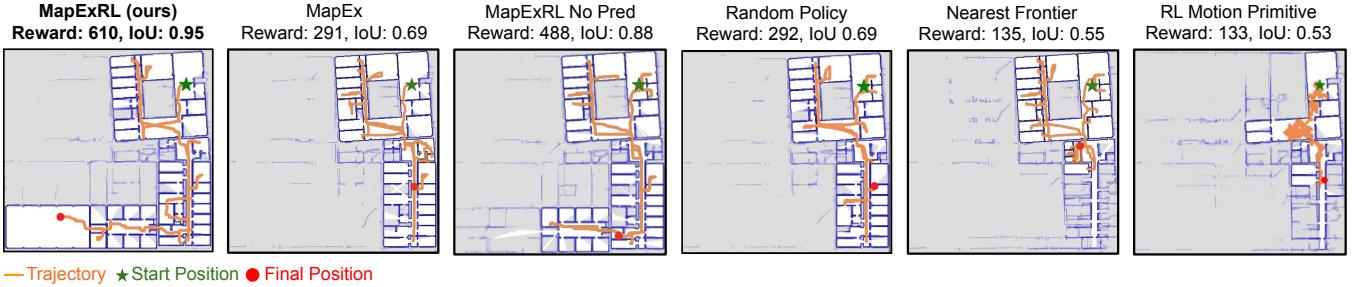


Fig. 6. Comparison of the six evaluated methods on Map 2 from the same start position. Notably, MapExRL is the only one that successfully explored the entire building and navigated the narrow passage leading to the bottom-left section that was not reached by the baseline methods.

with a learning rate of 0.00073 for both policy and critic networks. Our model is trained on an NVIDIA A100 GPU, which typically takes less than a day without parallelization to achieve 10,000 steps and train the optimal model.

VI. RESULTS

The performance of our model was evaluated on the 5 test maps with the same 15 randomly assigned start positions per map, resulting in a total of 75 experiments per method. The detailed results showing the average and 95% confidence interval for reward and IoU are presented in Table II.

Our policy outperforms all baseline methods in four of five test maps, ranking second only on Map 4. It also surpasses the ablation model across all maps, validating user insight 1, which motivated this ablation. These results demonstrate our model’s ability to reason about and effectively utilize predicted maps, exploration budget, and robot observations, reinforcing the correctness of the users’ insights. Notably, our policy performed exceptionally on Map 2 and Map 5, with Map 5 being the largest and most complex in our dataset. For comparison, the nearest frontier-based exploration method achieves an average IoU of only 49% on Map 5, whereas our method reaches 73%. This map also underscores the inherent challenges associated with large maps and emphasizes the advantages of predictive global maps. The nearest-frontier method exhausts its budget, exploring only a small portion of the map, due to many close-proximity frontiers.

Map 2 is an irregular L-shaped building that is challenging to explore, creating numerous opportunities for backtracking, which is a common issue with the baseline methods. In contrast, our policy often effectively selected paths that minimized backtracking and efficiently explored hard-to-predict areas using the global map prediction models. A sample trajectory across all methods on Map 2 is illustrated in Fig. 6, demonstrating the efficiency of our policy in topologically challenging environments. As observed, the global map prediction models struggled to predict the bottom-left part of the building accurately, and our approach was the only method to successfully explore that area.

Overall, we observe a 4.8% improvement in exploration efficiency in all 75 combined evaluations compared to MapEx, the strongest baseline, with improvements reaching up to 18.8% on the largest and most topologically challenging Map 5. Additionally, we achieved a 30% improvement in performance compared to the random policy and a 55%

improvement over the traditional nearest-frontier exploration method. Notably, our model outperformed all baselines on the largest and most topologically complex map (Map 5) and demonstrated strong performance on the irregular L-shaped map (Map 2), highlighting its advantages in challenging exploration tasks and its efficient use of Global Map Predictions and budget awareness. While performance gains are less pronounced on smaller maps due to their simplicity and limited time available to differentiate in performance, our model still maintains a consistent advantage.

Despite our careful implementation of the RL motion primitive method and earnest efforts to optimize the various parameters, the planner showed limited effectiveness. As shown in Fig. 6, this approach tended to resemble a random walk, with prolonged training primarily leading to a reduction in wall collisions. We verified our motion-primitive RL baseline on a simpler map, where it converged and explored well. However, when trained and tested on larger, complex maps, performance degraded significantly, likely due to many factors, including narrow doorways that are difficult to navigate and a weak training signal, as most actions yield no reward given the maps’ scale. While the model learned to avoid collisions, efficient exploration remained a challenge. This outcome highlights one of the advantages of our action space, which employs classical methods to effectively extract a set of actions from the map that will facilitate new observations.

To evaluate our method against the baseline established by our user study, we tested the MapExRL planner using the two sets of starting positions utilized in the study. Using the reward function from the user study, the combined reward for the start positions for rounds 1 and 2 was 860 ± 176 , while for round 3 it was 956 ± 164 . Notably, our model outperformed the human baseline in round 3. However, it did not surpass the users’ performance in rounds 1 and 2, primarily due to a few low-reward runs that significantly impacted the average score. While a larger sample size would allow a more definitive comparison with human operators, these results underscore the continued high performance of skilled human operators and highlight the necessity for robustness and consistency in an exploration planner. Furthermore, the findings indicate that there remains room for improvement in our approach to further bridge the performance gap between robot and human exploration strategies.

VII. CONCLUSION

In this work, we present MapExRL, a novel human-inspired method combining Reinforcement Learning with frontier-based exploration and global map predictions to improve indoor mapping efficiency. Our user study informs key design choices and provides a benchmark for evaluating exploration methods. By optimizing frontier selection through a learned policy, MapExRL improves exploration strategies, achieving higher predicted map IoU with fewer resources. Experimental results on a real-world floorplan dataset shows its advantage over traditional SOTA methods and ablation studies, particularly in large, complex environments.

In future work, we plan to extend our approach to cluttered environments and real-world robotic deployment. We also aim to refine the problem formulation by initiating exploration with a map of the building’s exterior walls, a practical assumption for many applications. This provides spatial scale awareness, enabling the model to optimize policy selection within budget constraints without requiring prior map scale discovery.

ACKNOWLEDGMENT

The authors want to thank all the people behind the CMU RISS program, especially Rachel Burcin and Dr. John M. Dolan. Special thanks to Kasper Grøntved, Jeric Lew, and Graeme Best for helpful discussions.

REFERENCES

- [1] B. Yamauchi, “A frontier-based approach for autonomous exploration,” in *Proc. of IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA)*, pp. 146–151, 1997.
- [2] G. Oriolo, G. Ulivi, and M. Vendittelli, “Real-time map building and navigation for autonomous robots in unknown environments,” *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 28, no. 3, pp. 316–333, 1998.
- [3] G. Best, R. Garg, J. Keller, G. A. Hollinger, and S. Scherer, “Multi-robot, multi-sensor exploration of multifarious environments with full mission aerial autonomy,” *The International Journal of Robotics Research*, vol. 43, no. 4, pp. 485–512, 2024.
- [4] S. Kim, M. Corah, J. Keller, G. Best, and S. Scherer, “Multi-robot multi-room exploration with geometric cue extraction and circular decomposition,” *IEEE Robotics and Automation Letters*, 2023.
- [5] G. Georgakis, B. Bucher, A. Arapin, K. Schmeckpeper, N. Matni, and K. Daniilidis, “Uncertainty-driven planner for exploration and navigation,” in *IEEE Conference on Robotics and Automation (ICRA)*, 2022.
- [6] R. Shrestha, F.-P. Tian, W. Feng, P. Tan, and R. Vaughan, “Learned map prediction for enhanced mobile robot exploration,” in *Proc. of International Conference on Robotics and Automation (ICRA)*, pp. 1197–1204, 2019.
- [7] C. Ho, S. Kim, B. Moon, A. Parandekar, N. Harutyunyan, C. Wang, K. Sycara, G. Best, and S. Scherer, “Mapex: Indoor structure exploration with probabilistic information gain from global map predictions,” in *2025 IEEE International Conference on Robotics and Automation (ICRA)*, 2025.
- [8] F. Niroui, K. Zhang, Z. Kashino, and G. Nejat, “Deep reinforcement learning robot for search and rescue applications: Exploration in unknown cluttered environments,” *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 610–617, 2019.
- [9] H. Li, Q. Zhang, and D. Zhao, “Deep reinforcement learning-based automatic exploration for navigation in unknown environment,” *IEEE transactions on neural networks and learning systems*, vol. 31, no. 6, pp. 2064–2076, 2019.
- [10] W.-C. Lee, M. C. Lim, and H.-L. Choi, “Extendable navigation network based reinforcement learning for indoor robot exploration,” in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 11508–11514, IEEE, 2021.
- [11] Y. Cao, T. Hou, Y. Wang, X. Yi, and G. Sartoretti, “Ariadne: A reinforcement learning approach using attention-based deep networks for exploration,” in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 10219–10225, IEEE, 2023.
- [12] F. Bourgault, A. A. Makarenko, S. B. Williams, B. Grocholsky, and H. F. Durrant-Whyte, “Information based adaptive robotic exploration,” in *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, vol. 1, pp. 540–545, 2002.
- [13] S. Bai, J. Wang, F. Chen, and B. Englot, “Information-theoretic exploration with Bayesian optimization,” in *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1816–1822, 2016.
- [14] A. Bircher, M. Kamel, K. Alexis, H. Oleynikova, and R. Siegwart, “Receding horizon” next-best-view” planner for 3d exploration,” in *2016 IEEE international conference on robotics and automation (ICRA)*, pp. 1462–1468, IEEE, 2016.
- [15] S. LaValle, “Rapidly-exploring random trees: A new tool for path planning,” *Research Report 9811*, 1998.
- [16] M. Luperto, L. Fochetta, and F. Amigoni, “Exploration of indoor environments through predicting the layout of partially observed rooms,” in *Proceedings of the 20th International Conference on Autonomous Agents and MultiAgent Systems*, AAMAS ’21, (Richland, SC), p. 836–843, International Foundation for Autonomous Agents and Multiagent Systems, 2021.
- [17] M. Luperto and F. Amigoni, “Reconstruction and prediction of the layout of indoor environments from two-dimensional metric maps,” *Engineering Applications of Artificial Intelligence*, vol. 113, p. 104910, 2022.
- [18] M. Saroya, G. Best, and G. A. Hollinger, “Online exploration of tunnel networks leveraging topological CNN-based world predictions,” in *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 6038–6045, 2020.
- [19] S. K. Ramakrishnan, Z. Al-Halah, and K. Grauman, “Occupancy anticipation for efficient exploration and navigation,” in *Proc. of European Conference on Computer Vision (ECCV)*, pp. 400–418, 2020.
- [20] E. Zwecher, E. Iceland, S. R. Levy, S. Y. Hayoun, O. Gal, and A. Barel, “Integrating deep reinforcement and supervised learning to expedite indoor mapping,” in *2022 International Conference on Robotics and Automation (ICRA)*, pp. 10542–10548, IEEE, 2022.
- [21] L. Ericson and P. Jensfelt, “Beyond the frontier: Predicting unseen walls from occupancy grids by learning from floor plans,” *IEEE Robotics and Automation Letters*, vol. 9, no. 8, pp. 6832–6839, 2024.
- [22] D. Zhu, T. Li, D. Ho, C. Wang, and M. Q.-H. Meng, “Deep reinforcement learning supervised autonomous exploration in office environments,” in *2018 IEEE international conference on robotics and automation (ICRA)*, pp. 7548–7555, IEEE, 2018.
- [23] F. Chen, S. Bai, T. Shan, and B. Englot, “Self-learning exploration and mapping for mobile robots via deep reinforcement learning,” in *Aiaa scitech 2019 forum*, p. 0396, 2019.
- [24] F. Chen, J. D. Martin, Y. Huang, J. Wang, and B. Englot, “Autonomous exploration under uncertainty via deep reinforcement learning on graphs,” in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 6140–6147, IEEE, 2020.
- [25] Y. Tao, E. Iceland, B. Li, E. Zwecher, U. Heinemann, A. Cohen, A. Avni, O. Gal, A. Barel, and V. Kumar, “Learning to explore indoor environments using autonomous micro aerial vehicles,” in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 15758–15764, 2024.
- [26] T. Haarnoja, A. Zhou, K. Hartikainen, G. Tucker, S. Ha, J. Tan, V. Kumar, H. Zhu, A. Gupta, P. Abbeel, et al., “Soft actor-critic algorithms and applications,” *arXiv preprint arXiv:1812.05905*, 2018.
- [27] 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.
- [28] R. Suvorov, E. Logacheva, A. Mashikhin, A. Remizova, A. Ashukha, A. Silvestrov, N. Kong, H. Goka, K. Park, and V. Lempitsky, “Resolution-robust large mask inpainting with Fourier convolutions,” in *Proc. of IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 2149–2159, 2022.
- [29] A. Aydemir, P. Jensfelt, and J. Folkesson, “What can we learn from 38,000 rooms? Reasoning about unexplored space in indoor environments,” in *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4675–4682, 2012.