Learning Goal-Directed Object Pushing in Cluttered Scenes Using Location-Based Attention

Nils Dengler[1,4,5*] Juan Del Aguila Ferrandis[2*] João Moura[2,3] Sethu Vijayakumar[2,3] Maren Bennewitz[1,4,5]

***Abstract*— In complex scenarios where typical pick-and-place techniques are insufficient, often non-prehensile manipulation can ensure that a robot is able to fulfill its task. We build on prior reinforcement learning methods for planar pushing by introducing location-based attention, enabling robust, collision-free manipulation in cluttered, dynamic scenes. Unlike previous approaches, our method needs no global path planning and considers target orientation. Simulated and real-world experiments with a KUKA iiwa arm validate the effectiveness of our policy.**

I. Introduction

Incorporating non-prehensile manipulation into a robot's skill set enhances its versatility beyond pick-and-place techniques [1], [2]. This capability allows robots to manipulate a wider range of ungraspable objects and access to otherwise unreachable grasping configurations through their repositioning and reorientation [3].

In cluttered environments, avoiding obstacles introduces a new dimension of complexity to non-prehensile manipulation, requiring advanced long-horizon spatial reasoning that integrates collision constraints while maintaining responsiveness to dynamic and unpredictable elements [4]. Therefore, a real-time scene understanding is essential to predict interactions, generate feasible trajectories, and adapt to both static and dynamic components in the scene. For example, Fig. 1 shows a scenario in which the robot pushes a cake to a person in order for them to reach it, while avoiding the other items on the table.

Current research predominantly focuses on precise object pushing in free space [5], [6] or on cluttered surfaces without restricting interactions between the objects [7], [8]. Only few studies consider pushing in cluttered environments while incorporating collision constraints [9], [10]. However, they rely on pre-computed path guidance and scale poorly to more complex scenarios [11]. Recently, Del Aguila Ferrandis *et al.* [12] demonstrated significant performance improvements in free-space pushing tasks by leveraging model-free reinforcement learning (RL) with categorical exploration to capture the multimodal behavior arising from the different possible contact interaction modes between the robot and the manipulated object.

∗ These authors contributed equally to this work.
[1]: Humanoid Robots Lab, University of Bonn, Germany
[2]: School of Informatics, The University of Edinburgh, Edinburgh, UK
[3]: The Alan Turing Institute, London, UK
[4]: The Lamarr Institute, Bonn, Germany
[5]: The Center for Robotics, University of Bonn, Germany

This work has partly been supported by the European Commission under grant agreement numbers 964854 (RePAIR) and by the BMBF within the Robotics Institute Germany, grant No. 16ME0999.

Fig. 1: Example scenario for pushing in a cluttered workspace. The robot moves a cake to a specified target pose while avoiding collisions with other objects on the table.

In this work, we extend [12] and [9] by proposing a system for cluttered pushing that replaces path priors with an occupancy grid map for flexible, generalizable RL. Unlike fixed object representations [9], our approach adapts to unseen layouts and dynamic obstacles. To mitigate the complexity of high-dimensional inputs, we incorporate a lightweight location-based attention mechanism [13] to focus on task-relevant spatial features. Our experiments show that this combination enables effective goal-directed pushing in complex, cluttered scenes.

In summary, our main contributions are as follows: (i) A reinforcement learning framework for non-prehensile object pushing in cluttered environments that operates without predefined guidance and incorporates location-based attention for improved spatial reasoning. (ii) A thorough quantitative evaluation in simulation, analyzing performance across different obstacle configurations, testing generalization through fine-tuning, and comparing the effectiveness of location-based attention against standard feature extractors in terms of success and collision rates. (iii) Qualitative and quantitative experiments on a KUKA iiwa robot, demonstrating reliable, smooth, and precise manipulation even in dynamic and realistically cluttered scenes.

II. Method

In this work, we consider the following problem. A robotic arm aims to push an object from its current pose to a target pose (x, y, θ) within a bounded planar workspace with its end effector, i.e., the pusher. In addition to the pushed object, there are other objects in the workspace which are obstacles the pushed object needs to avoid.

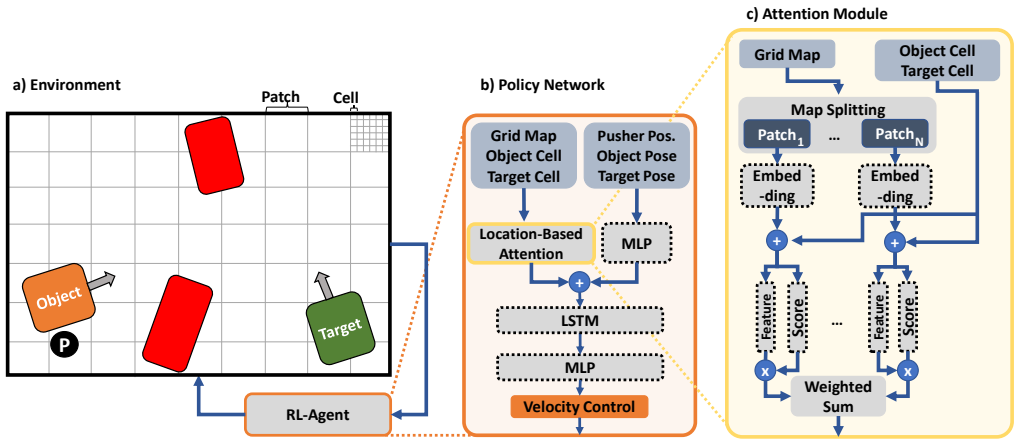


Fig. 2: Overview of our framework for learning goal-directed pushing using location-based attention. (a) The grid map of the environment together with the object and target pose, as well as the position of the pusher is fed to the RL-agent (b). In comparison to previous work [9], we use a location-based attention module (c) for feature extraction of the cluttered scene.

To address this problem, we propose an RL framework that leverages categorical exploration [12] to capture the multimodal nature of planar pushing, as well as location-based attention to extract and selectively focus on relevant spatial features from the workspace occupancy grid, achieving obstacle avoidance while manipulating the object towards the target pose. In the following, we describe the design of our RL framework, summarized in Fig. 2.

A. Feature Extraction

1) ***Preprocessing****:* At the beginning of each episode, we generate a binary occupancy grid of the workspace, where 1 represents obstacle and 0 free space. We use a resolution of $0.005\,\mathrm{m} \times 0.005\,\mathrm{m}$ per grid cell.

2) ***Location-Based Attention****:* Drawing inspiration from Visual Transformers [14], we decompose the occupancy map into n patches, each of size $P_s = 16 \times 16$, where $n \cdot P_s$ matches the size of the original map. We use a multilayer perceptron (MLP) of size $(192, 128)$ to embed each patch, as depicted in 2.a, encoding its features. This encoding process allows us to capture the essential characteristics of each patch, including obstacles and potential paths.

To provide positional context for each patch in the current task configuration, we concatenate them with the object and target positions, relative to the upper-left corner of each patch. From the patch embeddings and the positional context, we obtain the attention features and scores using separate MLPs of size $(128, 100, 64)$. Finally, we compute the weighted attention features as depicted in Fig. 2.c and feed the output of the location-based attention module to the RL agent.

B. Reinforcement Learning

The hybrid dynamics inherent in non-prehensile planar manipulation, characterized by varying contact modes such as sticking, sliding, and separation [15], make traditional unimodal exploration strategies, generally parametrized through multivariate Gaussian distributions, suboptimal. These strategies struggle to model the multimodal nature of interactions that arise from discrete contact transitions. Building on recent work in RL for accurate planar pushing [12], we adopt the on-policy RL algorithm Proximal Policy Optimization (PPO) [16], using a discretized action space to enable multimodal categorical exploration.

Below, we detail the main components of the RL pipeline.

1) ***Observation****:* The policy observation of the environment consists of the object and target poses (x, y, θ), the pusher position (x, y), and the binary occupancy grid that encodes the clutter layout. To reduce the computational cost during training, we keep the grid layout fixed throughout each episode. Nevertheless, we show in our hardware experiments that the grid representation can be updated in real time using, e.g., point cloud data or motion capture, and that the learned policies are robust to dynamic changes in the obstacle layout.

2) ***Action****:* We define the policy action as (v_x, v_y), the x and y velocity of the pusher. Furthermore, we limit the velocity on each axis to the range $[-0.1, 0.1]\ \mathrm{m\,s^{-1}}$ and use $0.02\ \mathrm{m\,s^{-1}}$ velocity steps for each categorical bin.

3) ***Reward****:* We define our reward function r_{total} as

$$r_{total} = r_{term} + k_1(1 - r_{dist}) + k_2(1 - r_{ang}) + r_{coll}, \quad (1)$$

with k_1, k_2 being scaling factors. r_{term} is a large sparse termination reward, which is positive when the object reaches the desired target pose and otherwise negative. r_{dist} is the Euclidean distance of the manipulated object to the target position, normalized to the range $[0, 1]$, and r_{ang} the angular distance of the object to the target orientation, also normalized to $[0, 1]$. In addition, we use r_{coll} as a binary negative reward to penalize at every step any kind of contact with an obstacle by the pusher or the object. If there is no collision during one time step then $r_{coll} = 0$.

4) ***Policy and Value Networks****:* We use the same architecture for the policy and value networks (seeFig. 2.b). In particular, the attention module extracts weighted attention features (size 64) from the occupancy grid. We also use an MLP (size 64) to extract features from the remaining observation, which consists of the object and target pose, as well as the pusher position. We concatenate these two feature vectors and feed them through a Long Short-Term Memory

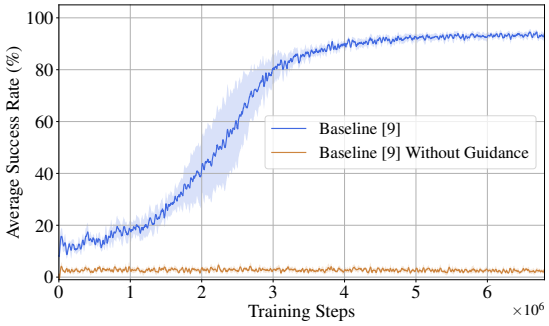


Fig. 3: Training performance of the baseline approach [9] (blue), as well as a variant without global path guidance (orange).

(LSTM) (size 256) layer and an MLP (size 128) layer. Using LSTMs for the policy and value networks enables to capture the hidden temporal dynamics of the environment, including friction and inertia. The final output of the value network is of size 1, corresponding to the state value estimate, while the policy network returns a vector of size 22, corresponding to logits that define the two categorical distributions for the velocities on the x and y axes.

III. Experimental Results

In this section, we evaluate our approach by first describing the experimental setup used for training and testing. We then assess the performance of current state-of-the-art work by Dengler *et al.* [9], analyzing the impact of global path guidance on task success. Furthermore, we investigate the role of the location-based attention mechanism by comparing it with alternative feature extraction methods and conduct a quantitative evaluation across various unseen obstacle configurations to validate the generalization capabilities of our approach in terms of success and collision rate. Finally, we demonstrate the effectiveness of our method in a physical hardware setup, highlighting its robustness in real-world scenarios, including dynamic environments.

We train agents in the Isaac Sim physics simulator [17], using a custom environment for pushing in clutter. To speed up training, we use a single rectangular obstacle as the default setup, with fine-tuning on two-obstacle scenarios. Each episode samples random poses for the pusher, object, obstacle, and target, ensuring the obstacle lies between object and target.

For training, we employ PPO with a training episode limit of 160 steps, that is extended to 200 during evaluation for increased complexity (e.g., novel obstacle shapes and multiple obstacles). The reward includes a termination bonus ($r_{term} = 50$ for success, -10 for boundary violations), a collision penalty ($r_{coll} = -5$), and distance-based terms weighted by $k_1 = 0.1$, $k_2 = 0.02$. To enable sim-to-real transfer, we apply dynamics randomization and synthetic observation noise during training.

A. Baseline and Influence of Path Guidance

Since the work of Dengler *et al.* [9] is most closely related to ours, we re-implemented their method in PyBullet [18] for comparison on our obstacle-avoidance pushing task. PyBullet was chosen due to their method's reliance on precomputed global paths, which limits GPU-parallelized training in Isaac Sim.

Following their setup, we ignore target orientation in this baseline. Attempts to include orientation, as in our method, led to convergence failure.

We trained their baseline without access to global path information—i.e., no sub-goal input. As shown in Fig. 3, the guided version converges, but removing global guidance leads to failure, highlighting its dependence on predefined paths and its limitations in handling our guidance-free, orientation-aware task.

Experimental Setup	Location Based Attention (Ours)		CNN Feature Extraction	
	Success Rate %	Collision Rate %	Success Rate %	Collision Rate %
Training	97.1	1.26	88.5	4.83
Circular	95.6	2.66	84.7	0.56
Cross-Shape	94.1	2.90	84.5	1.75
T-Shape	93.5	4.72	85.3	0.97
L-Shape	90.2	7.75	83.8	2.47
Dual Obstacles	48.1	50.7	57.9	34.3
Dual fine-tuned (DFT)	91.2	3.54	61.1	3.22
Circular (DFT)	96.4	0.20	72.1	0.34
Cross-Shape (DFT)	96.7	0.33	73.8	0.54
T-Shape (DFT)	96.3	1.32	71.9	1.01
L-Shape (DFT)	94.9	1.58	71.2	1.22

TABLE I: Performance comparison between location-based attention (Ours) and CNN feature extraction for different obstacle configurations varying in size, shape, and quantity.

B. Impact of Location-Based Attention on the Training

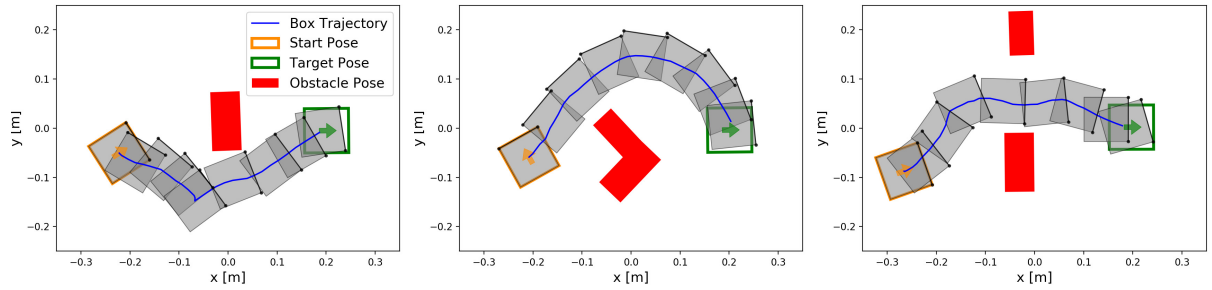
To assess the impact of location-based attention, we compare it to alternative occupancy grid processing methods during training and rollout. Specifically, we implement a standard CNN with three layers and an ablation of our method that omits the weighted attention sum, replacing it with feature concatenation and an MLP ([2048, 512, 64]). All models have similar parameter counts to ensure a fair comparison.

We also tested a multi-headed self-attention (MHA) module but excluded it due to its high memory demands, which made training infeasible on an NVIDIA A6000 (48GB VRAM) and severely slowed down parallel simulations.

Fig. 5 shows the resulting training curves for our proposed framework as well as the CNN and MLP modified approaches for processing the occupancy map. We report mean and standard deviation across three training seeds. We find that our approach with location-based attention achieves the highest final success rate (96%). On the other hand, while convergence is faster with the CNN structure, its asymptotic performance is noticeably lower (87%). Furthermore, the CNN has a 70% higher GPU memory consumption, due to the computational overhead from convolutional operations storing multiple large intermediate feature maps, making our method more efficient and with a better performance. Finally, the MLP ablation of our method, removing the weighted sum computation with attention scores, fails to converge, highlighting the critical role of selectively attending to spatial features.

C. Quantitative Evaluation

We quantitatively evaluate our method against the baseline CNN feature extraction described in Sec. III-B across various



(a) Training setup (b) L-shape obstacle (c) Dual obstacle

Fig. 4: Different obstacle configurations and the corresponding trajectories resulting from executing the push actions generated by our RL policy in the physical hardware setup. The three experiments show (a) pushing behavior with contact surface switching, (b) a smooth trajectory around an L-shaped obstacle, and (c) a precise pushing maneuver to fit the object through a narrow gap between two obstacles.

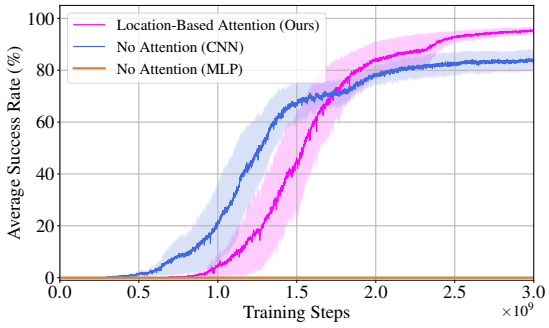


Fig. 5: Training performance on our obstacle avoidance pushing task, with (Ours) and without (CNN) attention for feature extraction.

unseen obstacle configurations, including circular, cross, T-, L-shaped, and dual obstacle setups (Fig. 4). We evaluate each trained policy for $2,000$ episodes per environment, with randomized start and target poses, as well as varying obstacle poses and sizes. We consider an episode successful when the pusher and the manipulated object avoid collisions, the object remains within the workspace boundaries, it is placed within $1.5\,\mathrm{cm}$ and $\pi/6$ rad of the target pose, and the task completes in no more than 200 steps.

As shown in Table I, our method consistently outperforms the CNN baseline in single-obstacle scenarios, achieving higher success rates and maintaining low collision rates. While the CNN shows slightly fewer collisions, this is reasoned by frequent inaction, leading to a high number of time-limit failures. In contrast, our agent remains active and successfully completes more tasks, even with unfamiliar obstacle shapes.

In the dual obstacle setup, the CNN initially performs better, achieving 57.9% success. However, after fine-tuning on the dual obstacle environment (DFT) for $5 \cdot 10^8$ steps, our method significantly improves to 91.2% success with only 3.54% collisions, demonstrating strong adaptability. The CNN baseline improves only marginally with fine-tuning, reaching 61.1% success.

Additionally, fine-tuning our method on the dual obstacle environment improves its performance on all single-obstacle cases, showing strong generalization. The CNN, however, performs worse after fine-tuning, indicating its poor generalization and limited adaptability through fine-tuning. These results highlight the effectiveness of our location-based attention mechanism in enabling both robustness and adaptability in cluttered environments.

D. Hardware Experiments

Our physical setup (Fig.1) uses a KUKA iiwa arm with OpTaS[19] to map task-space actions to joint commands. We evaluate generalization using two scene detection pipelines: MoCap (Vicon-based, high precision) and 3Cam (three RealSense D435s with AprilTags, more flexible). Note that for the hardware experiments, we decided to fix the target pose to simplify the setup, but our simulation experiments fully randomize it.

To quantitatively evaluate our system's performance, we tested 10 random initial configurations across three MoCap scenarios: (a) a standard setup with a single rectangular obstacle, (b) a single obstacle of an unseen shape, and (c) dual separated obstacles. Fig. 4 shows smooth pushing sample trajectories generated by the physical robot in these scenarios. The learned policy achieved a 100% success rate in (a) and (b), while in (c), it attained a 90% success rate due to a single collision.

Our supplemental video[1] provides further qualitative demonstrations of our system's performance in various real-world scenarios with both MoCap and 3Cam setups, that include e.g., the adaptability to dynamic changes.

IV. Conclusion

We presented a model-free RL framework for non-prehensile planar pushing with obstacle avoidance in cluttered environments. Our method combines categorical exploration with a lightweight location-based attention mechanism for efficient spatial feature extraction. Unlike prior work, it operates without global path guidance and accounts for target object orientation. Using an occupancy grid to represent clutter, the system adapts well to diverse and dynamic scenes. Experiments show high success rates and low collisions, even with unseen obstacle shapes, and effective fine-tuning in more complex multi-obstacle setups. Real-world tests further validate its robustness and precision under challenging conditions.

[1] https://www.youtube.com/watch?v=Ef0_oQiDq2E

REFERENCES

[1] A. Efendi, Y.-H. Shao, and C.-Y. Huang, "Technological development and optimization of pushing and grasping functions in robot arms: A review," *Measurement*, 2024.

[2] J. Stüber, C. Zito, and R. Stolkin, "Let's push things forward: A survey on robot pushing," *Frontiers in Robotics and AI*, 2020.

[3] W. Zhou and D. Held, "Learning to grasp the ungraspable with emergent extrinsic dexterity," in *Proc. of the Conference on Robot Learning (CORL)*, 2023.

[4] J. Moura, T. Stouraitis, and S. Vijayakumar, "Non-prehensile planar manipulation via trajectory optimization with complementarity constraints," in *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, IEEE, 2022.

[5] S. Wang, L. Sun, F. Zha, W. Guo, and P. Wang, "Learning adaptive reaching and pushing skills using contact information," *Frontiers in Neurorobotics*, 2023.

[6] J. Del Aguila Ferrandis, J. Moura, and S. Vijayakumar, "Learning Visuotactile Estimation and Control for Non-prehensile Manipulation under Occlusions," in *Proc. of the Conference on Robot Learning (CORL)*, 2024.

[7] L. Wu, Y. Chen, Z. Li, and Z. Liu, "Efficient push-grasping for multiple target objects in clutter environments," *Frontiers in Neurorobotics*, 2023.

[8] W. Bejjani, M. Leonetti, and M. R. Dogar, "Learning image-based receding horizon planning for manipulation in clutter," *Journal on Robotics and Autonomous Systems (RAS)*, 2021. DOI: https://doi.org/10.1016/j.robot.2021.103730.

[9] N. Dengler, D. Großklaus, and M. Bennewitz, "Learning goal-oriented non-prehensile pushing in cluttered scenes," in *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, IEEE, 2022. DOI: 10.1109/IROS47612.2022.9981873.

[10] A. Pasricha, Y.-S. Tung, B. Hayes, and A. Roncone, "Pokerrt: Poking as a skill and failure recovery tactic for planar non-prehensile manipulation," *IEEE Robotics and Automation Letters (RA-L)*, 2022. DOI: 10.1109/LRA.2022.3148442.

[11] V. Levé, J. Moura, N. Saito, S. Tonneau, and S. Vijayakumar, "Explicit contact optimization in whole-body contact-rich manipulation," in *Proc. of the IEEE-RAS Intl. Conf. on Humanoid Robots*, 2024.

[12] J. Del Aguila Ferrandis, J. Moura, and S. Vijayakumar, "Nonprehensile planar manipulation through reinforcement learning with multimodal categorical exploration," in *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2023. DOI: 10.1109/IROS55552.2023.10341629.

[13] M.-T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," *Proc. of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2015.

[14] "An image is worth 16x16 words: Transformers for image recognition at scale, author=Dosovitskiy, Alexey," *Proc. of the Intl. Conf. on Learning Representations (ICLR)*, 2021.

[15] F. R. Hogan and A. Rodriguez, "Reactive planar non-prehensile manipulation with hybrid model predictive control," *The International Journal of Robotics Research*, 2020.

[16] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[17] M. Mittal, C. Yu, Q. Yu, J. Liu, N. Rudin, D. Hoeller, J. L. Yuan, R. Singh, Y. Guo, H. Mazhar, A. Mandlekar, B. Babich, G. State, M. Hutter, and A. Garg, "Orbit: A unified simulation framework for interactive robot learning environments," *IEEE Robotics and Automation Letters (RA-L)*, 2023. DOI: 10.1109/LRA.2023.3270034.

[18] E. Coumans and Y. Bai, *PyBullet, a Python module for physics simulation for games, robotics and machine learning*, http://pybullet.org, 2016–2021.

[19] C. E. Mower, J. Moura, N. Z. Behabadi, S. Vijayakumar, T. Vercauteren, and C. Bergeles, "OpTaS: An Optimization-based Task Specification Library for Trajectory Optimization and Model Predictive Control," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 9118–9124. DOI: 10.1109/ICRA48891.2023.10161272.