Adaptive Policy Design in Dynamic Environments with LLM Planning and Human Feedback

Souradip Chakraborty, Bhrij Patel, Mohamed Elnoor, Ehaab Basil, Man Liang, Xiangyu Liu

***Abstract*— Reinforcement learning (RL) algorithms often struggle when faced with test time shifts and dynamic barriers, leading to failures in real-world scenarios. One significant challenge is the sim-to-real gap, where RL agents trained in simulations fail to perform in the real world. This research focuses on improving RL agent adaptability through just-in-time (JIT) querying and communication inspired by human behavior. We address two questions: when to ask for feedback and how to adapt the policy using it. We explore efficient quantification of uncertainty during test time and incorporate language-based feedback using few-shot learning in NLP. Our approach aims to bridge the sim-to-real gap and enhance RL agent reliability in practical settings. Finally, we explored the possibilities of replacing human with large language models like ChatGPT for planning and observed several failure modes of the same, where it failed to plan under several extremely simple scenarios. This exposes a critical vulnerability of LLMs in planning.**

***Index Terms*— Reinforcement learning, sim-to-real gap, adaptability, uncertainty quantification, just-in-time querying, language-based feedback, NLP.**

I. INTRODUCTION

Reinforcement learning (RL) has emerged as a powerful approach for tackling complex sequential decision-making tasks across various domains. RL algorithms have shown remarkable success in controlled settings, but they often face challenges when deployed in real-world scenarios. One significant issue is the sim-to-real gap, where RL agents trained in simulated environments fail to generalize their learning to the real world, leading to critical failures and suboptimal performance.

The sim-to-real gap becomes particularly evident in scenarios such as robotic navigation, where RL agents trained in simulated environments may collide or malfunction when deployed in the physical world. The gap arises due to the inherent differences between simulators and real-world environments, including variations in perception, dynamics, and uncertainty. Training RL agents to handle every possible variation in simulator design becomes impractical and limits their usability in real-world applications.

Enhancing the adaptability of RL agents to dynamic and unpredictable scenarios is crucial for their successful deployment in real-world settings. The ability to navigate unknown obstacles and handle novel situations is a key characteristic of human behavior that can serve as inspiration for improving RL agents' performance. By addressing the sim-to-real gap and improving adaptability, RL agents can achieve better safety and reliability in critical scenarios.

This research aims to develop a human-inspired approach to RL that focuses on improving test-time adaptability through efficient just-in-time (JIT) querying and communication. We investigate the optimal timing for RL agents to seek feedback and the most effective form of feedback to enhance their policies. Additionally, we explore uncertainty quantification during test-time scenarios to provide agents with a measure of confidence or uncertainty in their predictions.

Furthermore, we incorporate language-based feedback from humans or experts into the RL agents' inference paradigm without requiring policy retraining. By leveraging few-shot learning techniques in Natural Language Processing (NLP) and state-of-the-art language models, we aim to develop an efficient communication protocol for interpreting and adapting to human feedback.

Through experimental evaluations on simulated environments and real-world scenarios, we compare the performance of our proposed approach with existing RL algorithms, including those without human feedback or uncertainty-driven techniques. The ultimate goal is to bridge the sim-to-real gap, enhance the reliability of RL agents, and enable their effective deployment in practical settings.

In the following sections, we present our research methodology, describe the experiments conducted, and analyze the results obtained. We discuss the implications of our findings and highlight future research directions to further improve RL agents' adaptability and address the challenges associated with the sim-to-real gap.

II. RELATED WORKS

To address the challenges faced in complex and dynamic environments, there is an urgent need for innovative solutions that improve the adaptability of reinforcement learning (RL) policies for robot navigation. Previous research has investigated a variety of methods to bridge the sim-to-real gap and improve the robustness of RL agents, such as transfer learning [1], meta-learning [2], and domain randomization [3, 4].

In this section, we provide a review of the existing literature on robot navigation with human feedback and natural language in perceptually challenging environments.

Human Feedback and Natural Language Processing in RL-based Robot Navigation. Human feedback is advantageous for robot navigation because it allows an agent to learn effectively from valuable feedback provided by a human expert. The use of human feedback has led

to improvements in the performance of RL in real-life problems [5]. For example, navigating robots in perceptually challenging environments is a difficult task without human feedback [6]. However, natural language processing is needed to leverage the benefits of human feedback [7, 8]. A recent work proposed in [9] uses language-based feedback when the trained policies are uncertain of real-time changes in the environment.

Autonomous Robot Navigation in Environments with Noisy Sensor Data. Navigating robots in real-world environments with noisy sensory data can be quite challenging due to varying conditions such as lighting changes, occlusions, and motion blur. Erroneous state estimations caused by inaccurate perception models are often the main cause of robot navigation failures. Although some algorithms incorporate a single sensor modality for uncertainty modeling [10], these methods require a controlled environmental setting. On the contrary, although some methods incorporate sensor fusion for utilizing multiple sensors [11], they often come at the cost of increased computational complexity.

III. Methodology

A. Research questions

This research draws inspiration from human behaviour, where individuals encounter unknown obstacles in the real world and seek assistance to overcome them. The initial step involves recognizing the presence of an unknown obstacle and subsequently seeking help from a supervisor or expert, demonstrating the ability to discern the type of assistance required.

The primary objective of this project is to develop a human-inspired approach to reinforcement learning (RL), specifically focusing on enhancing test-time adaptability through efficient just-in-time (JIT) querying and communication. To accomplish this, two fundamental research questions are addressed. Firstly, when is the optimal time for an RL agent to request feedback? Secondly, what is the most effective form for the RL agent to solicit feedback, and how can the received feedback be used to adapt the policy?

We hypothesize that an accurate quantification of epistemic uncertainty, considering the design characteristics of the robot's input space, can provide insights into the optimal timing for seeking feedback. While uncertainty-driven learning approaches have been explored in real-world RL problems, previous research has predominantly concentrated on the training phase, with limited consideration given to the interplay between observation design and uncertainty quantification. Additionally, we aim to address the challenge of incorporating language-based feedback into the RL inference paradigm without necessitating policy retraining. Human feedback is often conveyed through natural language, which presents challenges for an RL agent in interpretation and can result in miscommunication. Thus, our project aims to develop an efficient language-based communication protocol to seamlessly integrate human feedback into the RL inference paradigm.

By answering these research questions, we endeavour to advance the field by improving RL agents' adaptability and addressing the limitations posed by the sim-to-real gap. Furthermore, we aim to enable the effective deployment of RL agents in practical settings, enhancing their reliability and applicability in real-world scenarios.

B. Environment Description: Baseline Environmental Design

For the initial proof of concept of our approach, we constructed a 2D Maze Environment, both with and without obstacles, to evaluate the performance of the RL agent. The objective of the agent in this environment is to navigate through the maze and reach the final goal position using discrete steps. The basic structure of the environment, with and without obstacles, is visually depicted in Figure 1.

Specifically, the maze environment is represented as an $N \times N$ grid, where each cell corresponds to a state. The state space is a vector of dimension N^2, denoted as $s_t = [s_t^{1,1}, s_t^{1,2}, \cdots, s_t^{N,N}]^T$, where $s_t^{i,j}$ represents the value of the cell at the i^{th} row and j^{th} column of the maze environment at time t. The state s_t serves as the observation or state space of the RL agent.

In the maze environment, the following cell values are assigned: $s_t^{i,j} = 0$ if the cell is unoccupied and the agent is free to move into it, $s_t^{i,j} = 2$ if the agent occupies the cell, $s_t^{i,j} = 1$ if the cell contains an obstacle that terminates the game if the agent collides with it, and $s_t^{i,j} = 3$ if the cell represents the goal position.

Additionally, we introduce a local observation space by considering a smaller region around the agent's current location. This local observation space is obtained by selecting a sub-grid of size $N' \times N'$, where $N' << N$, ensuring a localized view of the environment around the agent.

By employing these maze environments, we evaluate the performance and adaptability of our RL approach in navigating through obstacles and reaching the goal position. This allows us to gain insights into the effectiveness of our proposed methods and their potential for generalization to continuous settings.

C. Motivation & Problem Formulation

The primary aim of this study is to develop an efficient and safe approach for adapting to test-time unseen dynamic barriers by leveraging uncertainty quantification and language feedback. Drawing inspiration from the success of Language Model-based Models (LLMs) in various language-guided tasks, such as language translation, text summarization, information retrieval, recommendation engines, and language-grounded robotics, we propose to harness their capabilities to efficiently handle such scenarios.

To address this objective, we focus on two key questions: 1) determining the appropriate time for an RL agent to request feedback when encountering dynamic barriers, and 2) establishing effective methods for interaction and communication between the RL agent and ChatGPT in

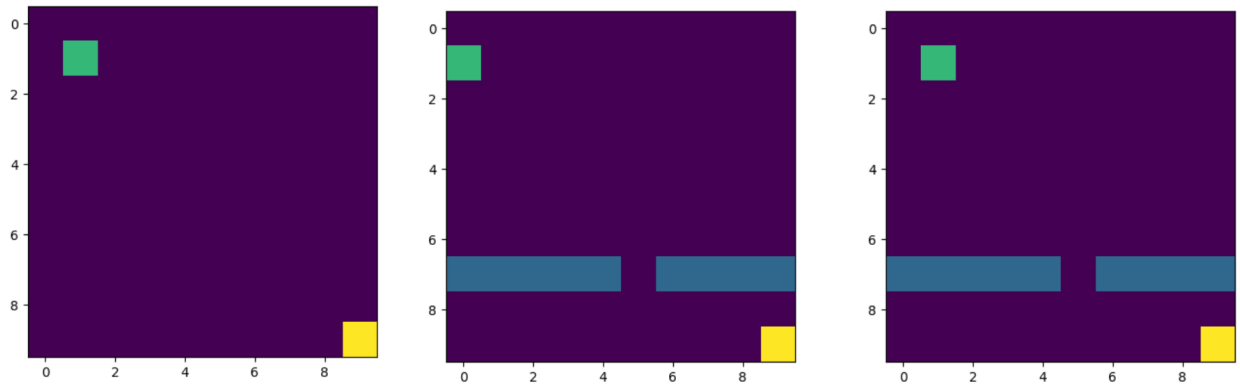


Fig. 1: Visual representation of the Basic Maze environment without and with Obstacles.

order to address the aforementioned scenario. To tackle the first question, we explore uncertainty quantification methods, with an emphasis on accurately estimating epistemic uncertainty within the specific design considerations of the robot's input space. We underscore the significance of quantifying epistemic uncertainty in these scenarios, as monitoring predictive uncertainty can be challenging. For example, in a grid cell, multiple optimal paths may exist towards the goal, leading to high policy entropy and creating uncertainty. Hence, as an initial step in our methodology, we conduct a simple experiment using imitation learning, training an agent's policy in an obstacle-free environment and evaluating its performance when obstacles or unknown barriers are introduced at test time.

Hence, as the first step of our methodology, we designed a simple experiment in the imitation learning paradigm where we train the policy of an agent in an obstacle-free goal-reaching environment and test the performance by adding an obstacle or unknown obstacle at the test time. Next, we introduce the setup and the imitation learning paradigm in a more formal manner.

We first collect Expert demonstrations, denoted by $\mathcal{D}$ i.e. a collection of optimal trajectories τ, where $\mathcal{D} := \{\tau_1, \tau_2, \cdots, \tau_N\}$, where τ_i represents the i^{th} trajectory given by $\tau_i = \{s_0^i, a_0^i, s_1^i, a_1^i, \cdots, s_T^i, a_T^i\}$ where (s_t^i, a_t^i) represents the state-action pair from i^{th} trajectory at timestep t.

The objective of learning behavioural policy boils down to maximizing the log-likelihood of the occurrence of the trajectories under the demonstration as

$$\log L(\theta) = \log\left(\prod_{i=1}^{n} P_\theta(\tau_i)\right) = \sum_{i=1}^{n} \log P_\theta(\tau_i), \qquad (1)$$

where $P_\theta(\tau_i)$ denotes the probability of the i^{th} trajectory of the expert. We note that $P_\theta(\tau_i) = P(s_0^i)\prod_{t=1}^{T} \pi_\theta(a_t^i|s_t^i)P(s_{t+1}^i|s_t^i, a_t^i)$. Using $P_\theta(\tau_i)$ into (1), we can obtain the gradient of the objective $\log L(\theta)$ with respect to θ is given by

$$\nabla_\theta \log L(\theta) = \sum_{i=1}^{n}\sum_{t=1}^{T} \nabla_\theta \log \pi_\theta(a_t^i|s_t^i). \qquad (2)$$

Thus with the imitation learning-based formulation, we can estimate the gradient w.r.t. to the loss function $\nabla_\theta \log L(\theta)$ using (2). However, the trained policy π_θ works well in training scenarios but fails to adapt to unseen test scenarios. Hence, an efficient representation of uncertainty is critical such that once the uncertainty reaches a certain threshold, the agent can stop and ask for feedback

With the above formulation, we run several experiments varying the shape and purpose of the dynamic barriers/obstacles (need not always obstruct the path) and monitor the posterior variance with a bootstrap ensemble-based method. Very interestingly, we observe that there is a close dependence between the observation space and the downstream uncertainty evaluation whereas global observation gives an imprecise characterization of uncertainty, local observations are more effective in giving a precise characterization of uncertainty, which is beneficial in answering the question of *When to Stop and ask for Help? We performed several experiments by varying the shape and size of the obstacle and the detection work and the agent can stop almost 90% of the time.*

Figure 2 illustrates the monitoring of posterior variance in both training and testing environments. In the training environments, where no unknown components exist, the posterior variance remains consistently low. In contrast, the testing environments, where the policy is learned with global observation, exhibit imprecise uncertainty characterization. On the other hand, the use of local observation during policy learning results in precise uncertainty characterization.

D. Prompt Design

In this research, we introduce an advanced interpretation and adaptation module to augment the performance of reinforcement learning agents through the effective integration of human language feedback. Our approach leverages the cutting-edge ChatGPT model and employs a zero-shot learning methodology to refine the prompt design, thereby aiming to substantially enhance the quality and efficacy of human-agent interactions.

The fundamental principle of our proposed method involves the conversion of natural language input from

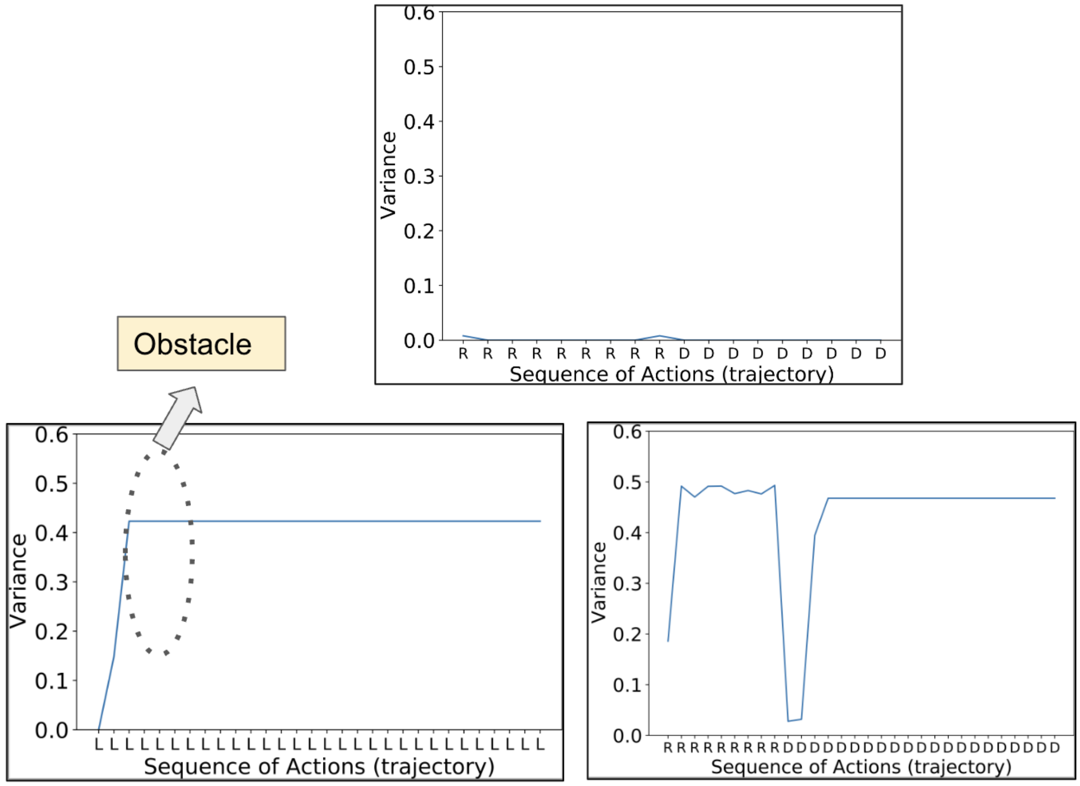


Fig. 2: Posterior Variance being monitored over the episodes in the training(top figure) and testing environments (bottom). As it can be clearly seen that in the training environments, there are no unknown components, and hence the posterior variance is low consistently. The figure in the bottom-right represents the test environment where policy is learned with global observation resulting in an imprecise characterization of uncertainty. On the other hand, the figure in the bottom left represents the test environment where policy is learned with the local observation which results in a precise characterization of uncertainty

humans into a structured sequence of actions that can seamlessly integrate into the decision-making process of the reinforcement learning agent. As depicted in Figure 3, the human user provides a comprehensive and intuitive description of the desired next steps, which the agent subsequently processes to extract the corresponding sequence of actions.

```
Enter 0 if human interference is not needed else 1: 1
Taking human input
Describe the next steps: move left 7 times and then down two times
```

Fig. 3: Human natural feedback

To accomplish this, we harness the powerful capabilities of the ChatGPT model, renowned for its exceptional performance in understanding and generating human-like text. By combining this model with crafted prompts, we attempted to exploit the benefits of zero-shot learning, empowering the reinforcement learning agent to accurately comprehend and execute the commands provided by the user without the requirement for explicit instruction or additional training.

E. Architectural Design for Visual Policy Training

Within this subsection, we explore the development and implementation of low-dimensional representations derived from high-dimensional visual images for reinforcement learning agents operating in Goal-Conditioned and Global Observation scenarios. Our primary objective is to establish state-of-the-art architectures that effectively process inputs and enable informed decision-making in these environments. This approach is pivotal for addressing the unique challenges encountered in complex real-world scenarios where reinforcement learning agents must make decisions based on incomplete or partially observable information.

1) Goal-Conditioned Scenario: To accommodate the distinct inputs in the Goal-Conditioned scenario, we have devised a sophisticated Convolutional Neural Network (CNN) architecture comprising two branches. The first branch is responsible for processing the image input corresponding to local observations and employs convolutional layers to extract meaningful features and patterns from the high-dimensional visual data. The second branch, which consists of a dense layer, handles the distance-to-goal input and encodes the spatial relationship between the agent and its objective. Figure 4 visually illustrates the reinforcement learning architecture of our model, meticulously designed to promote efficient information integration.

By employing this advanced hybrid visual representation

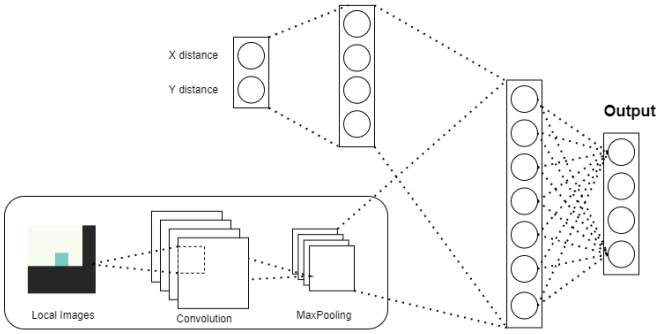


Fig. 4: The architecture of the $\text{Goal}_{C}onditionedscenario.$

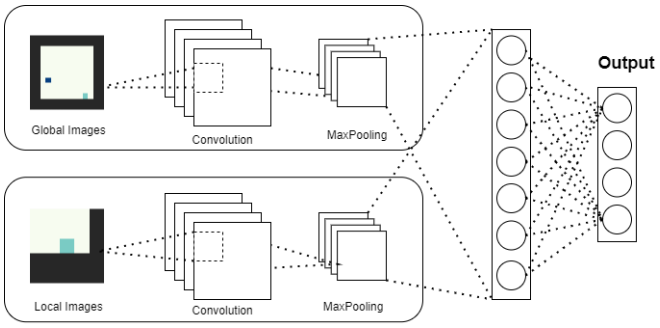


Fig. 5: The architecture of the local global model scenario.

technique for the Goal-Conditioned scenrio, the reinforcement learning agent can efficiently integrate local observations and distance-to-goal information, enabling informed decision-making within the given environment.

2) Global Observation Scenario: In the Global Observation scenario, we have devised a convolutional neural network (CNN) architecture that effectively processes both the global map and local observations of the agent as inputs. This architecture consists of two distinct branches, each dedicated to handling one of the input types. The first branch is specifically designed for processing the global map, while the second branch focuses on the local observation images.

To enable the reinforcement learning agent to integrate insights from both the global map and local observations, these two branches are merged or concatenated. By combining the features extracted from both inputs, the integrated representation is fed through fully connected layers, resulting in an integer output representing one of the potential actions. The reinforcement learning architecture of our model, carefully constructed to incorporate this merging of inputs, is depicted in Figure 5. This dual-input visual representation technique empowers the reinforcement learning agent to effectively fuse global and local observations, thereby facilitating informed decision-making within the given environment.

IV. Experiments

This subsection presents the training results of our reinforcement learning agents in the Goal-Conditioned and Global Observation scenarios, highlighting the effectiveness

Fig. 6: Training loss for the images-only model.

of the proposed architectures in leveraging the respective inputs to make informed decisions. The agents were trained using imitation learning techniques with expert data generated from a grid world environment specifically designed to encourage the agent to find the shortest path to the goal. The expert demonstrations were obtained using Dijkstra's algorithm to ensure high-quality training data.

A. Local Observation vs. Local Observation + Distance to Goal Observation

To further validate the importance of incorporating distance-to-goal information, we conducted experiments using a variant of the model that only utilized local observation images as input, excluding the distance input. Interestingly, the model failed to converge in this scenario. This outcome is deemed reasonable, as the agent would struggle to navigate the environment with only local observations and no information regarding the distance to the goal or global observations. Figure 6 illustrates the training loss for the images-only model.

These training findings underscore the significance of combining both local observation and distance-to-goal information in the proposed architecture for the Goal-Conditioned scenrio. By effectively integrating these two inputs, our reinforcement learning agent can adeptly navigate the environment and make informed decisions, thus validating the efficacy of the advanced hybrid visual representation technique in goal-conditioned reinforcement learning tasks.

The experiments conducted demonstrate the value of the proposed architectures and highlight their potential in facilitating the development of more sophisticated reinforcement learning agents. These agents are capable of efficiently navigating complex environments by incorporating multiple sources of information for decision-making. Figure 7shows the convergence of the local_global model.

B. Visual Uncertainty Estimation

Out-of-distribution detection is a critical component of ensuring the safe deployment of machine learning models, as it helps in identifying whether a model is making reliable predictions within its training data distribution, or making

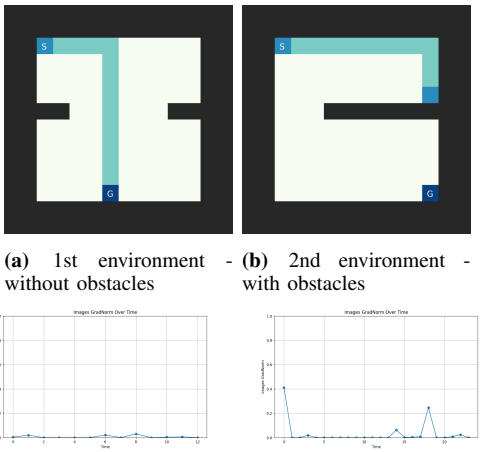
Fig. 7: Convergence for the local_global model.

uncertain predictions for inputs it has not encountered during training. In our efforts to improve the robustness of our obstacle avoidance model, we attempted two OOD detection methods: action probability and entropy measurement, and the GradNorm method. However, both methods did not yield the expected results in our context, prompting us to employ traditional similarity measurement methods.

1) Action Probabilities and Entropy Measurement: The first method involved analyzing the action probabilities and entropy of the model's output. The fundamental idea was that if the model's output probabilities were uniform across the four actions, indicated by high entropy, the model's confidence in its prediction would be low and vice versa.

However, this method did not work effectively in our case. One possible reason for this is that our model might be overly confident, even when encountering novel or ambiguous scenarios. This phenomenon, known as overconfidence, is a common issue in deep learning models and can lead to low entropy values even for out-of-distribution inputs, thus rendering this method ineffective for our use-case.

2) GradNorm Method: The second method we employed was GradNorm [12], a gradient-based OOD detection approach. This method leverages the vector norm of gradients, backpropagated from the Kullback-Leibler (KL) divergence between the softmax output and a uniform probability distribution. The core idea is that the magnitude of gradients is higher for in-distribution (ID) data than for OOD data, making it informative for OOD detection. Despite the theoretical soundness of this method, it did not perform as expected in our scenario. We hypothesize that in the case of high-dimensional images, the network acquires implicit low-dimensional representations through learning and since the network lacks exposure to the new test images, it may be unable to detect at test time due to projecting in the known space. This might also be due to the complexity of our specific task, where there might not be a clear distinction between in-distribution and out-of-distribution samples in the gradient space. This could potentially occur if the obstacles introduced in the test data are not significantly different from the training data, hence causing the gradients for both ID and OOD data to be similar. Figure 8 shows the performance of GradNorm in detecting OOD. The GradNorm score is low in the environments, even though there is an obstacle in the second one.



(a) 1st environment - without obstacles **(b)** 2nd environment - with obstacles

(c) GradNorm for the 1st 1st environment **(d)** GradNorm for the 2nd environment

Fig. 8: GradNorm performance in different environment

3) Traditional Similarity Measurement Methods: In light of the unsatisfactory performance of the two OOD detection methods, we turned to traditional methods to estimate the uncertainty of our model. Specifically, we calculated the similarity between the training images (without obstacles) and the test/implementation images (with obstacles) using the structural similarity index measure (SSIM) [13]. This approach, while seemingly straightforward, is a time-tested method that can effectively differentiate between in-distribution and out-of-distribution samples by measuring the degree of similarity or difference from the training data. Figure 9 shows that SSIM was able to successfully detect the obstacle as OOD (higher density).

V. Challenges and Future Steps

High Dimensional Representation and Uncertainty: In the vectorized Maze environment with local observation, the posterior variance estimation using bootstrapped ensembles demonstrates promising results. However, real-world

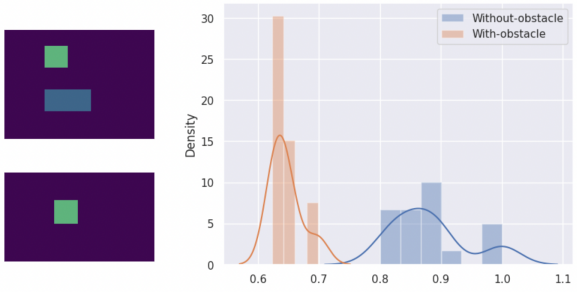


Fig. 9: OOD detection using SSIM. Local feature-based SSIM was able to efficiently detect OOD as shown in Figure 9. Tradition gradient norm-based approaches failed due to the implicit low-dimensional projection of NNs parametrizing the policy which ignore new objects at test time.

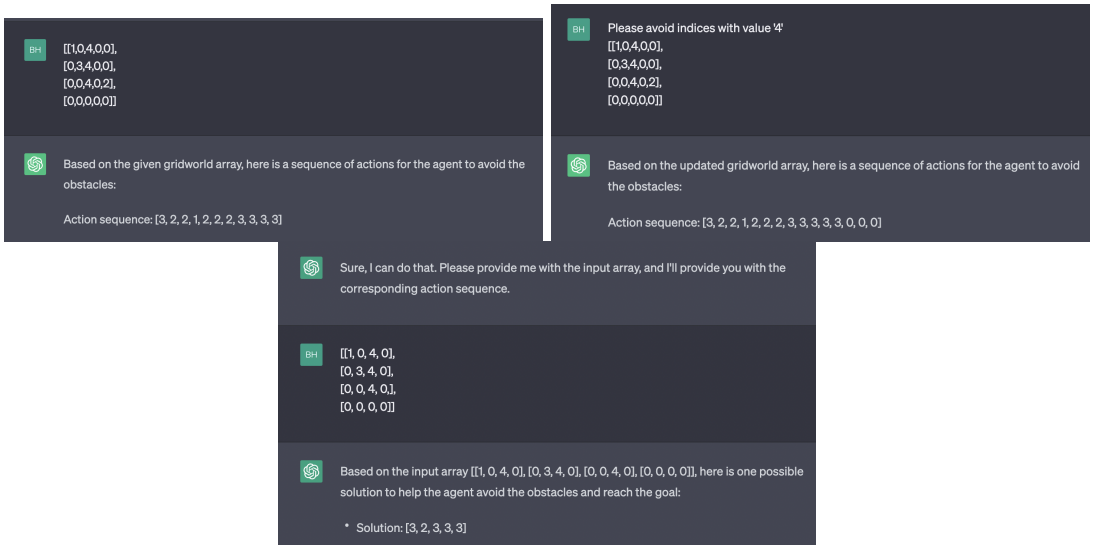


Fig. 10: The prompting methods used to achieve a valid path after entering the domain description: (a) shows solely inputting a 2D array, (b) adds in a reminder to avoid indices with the value '4', and (c) shows inputting an array after giving two example problems and solution pairs. In all cases, ChatGPT failed to avoid obstacles. In (a) and (b), the paths are also unreasonably long. In (c) we are only asking it to give a path that avoids obstacles, not even indicating a goal.

scenarios often involve high-dimensional visual images, requiring the policy to be conditioned on low-dimensional representations for efficient inference during test time. Generalizing to unknown objects with low-dimensional representations poses a significant challenge. Specifically, understanding the relationship between downstream uncertainty and learned representations with provable guarantees remains largely unexplored in the literature, making it a crucial question to address in our scenario.

LLM as a Prior and Efficient Prompt Design: The next crucial step involves designing an effective interaction or communication module for the agent to interact with Language Model-based Models (LLMs) like ChatGPT when encountering dynamic barriers. Humans find this task relatively straightforward, as adaptability to human feedback is reasonably simple. However, the main challenge lies in efficiently conveying the visual/vectorized scenario to the language model. During test time, when we input the vectorized/visual representation to the LLM, it should efficiently generate a description and provide the corresponding solution. We evaluate the zero-shot performance of ChatGPT with basic prompts describing the current configurations and rules for the vectorized input and observe the results in Figure 10.

Large Language Models (LLMs) increase their importance to society every day. Their massive appeal stems from their remarkable capabilities in zero-shot generalization, making them extremely versatile in various tasks, such as editing, web searching, and coding. For our purposes to remove any need for human supervision during test-time deployment, we would like to test one LLM, ChatGPT's, performance in short-term path planning. As a parallel task to the visual recognition and uncertainty component, we tested ChatGPT by giving it in 2D grid world environments for potential future work in extending to raw images as input.

To initialize all experiments, we describe the domain of the problem to ChatGPT:

Large Language Models (LLMs) increase their importance to society every day. Their massive appeal stems from their remarkable capabilities in zero-shot generalization, making them extremely versatile in various tasks, such as editing, web searching, and coding. For our purposes to remove any need for human supervision during test-time deployment, we would like to test one LLM, ChatGPT's, performance in short-term path planning. As a parallel task to the visual recognition and uncertainty component, we tested ChatGPT by giving it in 2D grid world environments for potential future work in extending to raw images as input.

To initialize all experiments, we describe the domain of the problem to ChatGPT:

Human: An agent is navigating in a 2-D gridworld represented as a 2-D array. In the array, the value '0' represents an open space, '1' represents the start position, '2' represents the goal, '3' represents where the agent is, and '4' represents an obstacle.
As input, you will be given the 2-D gridworld array, and we want you to output a sequence of actions that makes the agent avoid the obstacles. You only need to give enough actions to avoid the obstacles, do not worry about instructing on how to reach the goal once the obstacles are avoided.
The actions should be represented by integers as listed below:
'0' represents moving left, meaning the column index decreases by 1

'1' represents moving up, meaning the row index decreases by 1
'2' represents moving right, meaning the column index increases by 1
'3' represents moving down, meaning the row index increases by 1
I am not asking for code or general approaches. I am going to give you arrays in the chat that represent the 2D gridworld and I want you to output a sequence of actions as specified.
ChatGPT: Sure, I can help you with that! Please provide me with the 2D gridworld array.

Below are the types of prompts we tried after giving domain setup:

- Input arrays with reminders to ChatGPT to avoid the obstacle ("Please avoid indices with value 4").
- Input arrays that represent local observation that does not contain the goal.
- Another approach we tried was *in-context learning* where given example of input problems and solutions, then model can then generalize and output valid solutions to new input problems.

Figure 10 shows the results of these different prompt constructions. Throughout all these experiments, ChatGPT consistently fails to provide valid paths, suggesting actions that would make the agent collide with the obstacle.

As another test, we asked ChatGPT to list the indices with the value '4' in an array. Figure 11 shows that ChatGPT adds a fourth index even though there are only three indices with a '4', thus showing it cannot reliably determine what's an obstacle.

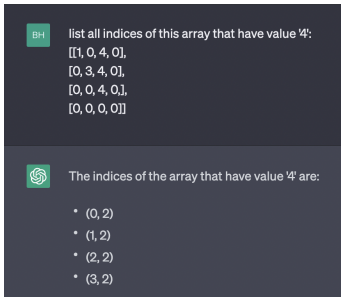


Fig. 11: Performance of ChatGPT to list indices with the value '4'. It lists a fourth and incorrect index, suggesting that it cannot reliably determine obstacles.

From these results, we see that ChatGPT has little generalization ability for short-term planning. Even after pointing out errors in the path, ChatGPT still failed to give a valid path, making the same mistakes again. As evident from Figures 10 and 11, the descriptions generated by ChatGPT are sub-optimal and require further research in prompt designs to obtain optimal descriptions. Therefore, one of our research directions is to explore advancements in prompt-tuning to efficiently describe visual or vectorized representations. Further evaluation with metrics used in classic path planning such as path length and path smoothness may give us more insights than just the binary metric of whether the generated path is valid or not. Another experiment to test ChatGPT or other LLMs, like BARD, could be to find paths in weighted environments.

TABLE I: Algorithm Performance with Different Policy Parametrization on Vectorized Maze Environment. Clearly shows the performance of tree-based parameterizations work better in tabular/vectorized inputs even better than MLPs.

Algorithms	Train Acc	Test Acc
Random Forest	96%	92%
Logistic Regression	88%	85%
XgBoost	91%	87%
MLP	87%	83%

VI. Conclusions

In this project, we proposed a framework that aims to improve the adaptability of autonomous agents by incorporating feedback into the design process. The comparison with baselines is conducted at different stages to evaluate the effectiveness of the proposed framework. In the initial phase, we compared with ML algorithms for policy parametrization and in the next stage for OOD detection and uncertainty quantifications. We observed that bootstrapped uncertainty works reasonably well for vectorized environments and traditional OOD for visual environments. This is primarily due to the implicit representation of NN projecting into the lower dimensional space characterized by the training environment and hence failing to test-time new shifts. We then formulate the communication module where the agent interacts with humans via an efficient language-action module which is achieved through efficient prompt tuning. Finally, we explored the possibilities of replacing human with large language models like ChatGPT for planning and observed several failure modes of the same, where it failed to plan under several extremely simple scenarios. This exposes a critical vulnerability of LLMs in planning and is a point of future research

VII. Source Code

Our code is publicly available here:
https://github.com/MohamedElnoor/adaptiveLLMplanning

References

[1] W. Zhao, J. P. Queralta, and T. Westerlund, "Sim-to-real transfer in deep reinforcement learning for robotics: a survey," in *2020 IEEE symposium series on computational intelligence (SSCI)*, pp. 737–744, IEEE, 2020.

[2] K. Li, A. Gupta, A. Reddy, V. H. Pong, A. Zhou, J. Yu, and S. Levine, "Mural: Meta-learning uncertainty-aware rewards for outcome-driven reinforcement learning," in *International conference on machine learning*, pp. 6346–6356, PMLR, 2021.

[3] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, "Domain randomization for transferring deep neural networks from simulation to the real world," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 23–30, IEEE, 2017.

[4] M. Sheckells, G. Garimella, S. Mishra, and M. Kobilarov, "Using data-driven domain randomization to transfer robust control policies to mobile robots," in *2019 International Conference on Robotics and Automation (ICRA)*, pp. 3224–3230, IEEE, 2019.

[5] G. Li, R. Gomez, K. Nakamura, and B. He, "Human-centered reinforcement learning: A survey," *IEEE Transactions on Human-Machine Systems*, vol. 49, no. 4, pp. 337–349, 2019.

[6] S. Agnisarman, S. Lopes, K. C. Madathil, K. Piratla, and A. Gramopadhye, "A survey of automation-enabled human-in-the-loop systems for infrastructure visual inspection," *Automation in Construction*, vol. 97, pp. 52–76, 2019.

[7] K. X. Nguyen, Y. Bisk, and H. D. Iii, "A framework for learning to request rich and contextually useful information from humans," in *International Conference on Machine Learning*, pp. 16553–16568, PMLR, 2022.

[8] B. Peng, J. MacGlashan, R. Loftin, M. L. Littman, D. L. Roberts, and M. E. Taylor, "A need for speed: Adapting agent action speed to improve task learning from non-expert humans," in *Proceedings of the international joint conference on autonomous agents and multiagent systems*, 2016.

[9] S. Chakraborty, K. Weerakoon, P. Poddar, P. Tokekar, A. S. Bedi, and D. Manocha, "Re-move: An adaptive policy design approach for dynamic environments via language-based feedback," *arXiv preprint arXiv:2303.07622*, 2023.

[10] K. Katyal, K. Popek, C. Paxton, P. Burlina, and G. D. Hager, "Uncertainty-aware occupancy map prediction using generative networks for robot navigation," in *2019 International Conference on Robotics and Automation (ICRA)*, pp. 5453–5459, IEEE, 2019.

[11] J.-w. Hu, B.-y. Zheng, C. Wang, C.-h. Zhao, X.-l. Hou, Q. Pan, and Z. Xu, "A survey on multi-sensor fusion based obstacle detection for intelligent ground vehicles in off-road environments," *Frontiers of Information Technology & Electronic Engineering*, vol. 21, no. 5, pp. 675–692, 2020.

[12] R. Huang, A. Geng, and Y. Li, "On the importance of gradients for detecting distributional shifts in the wild," *Advances in Neural Information Processing Systems*, vol. 34, pp. 677–689, 2021.

[13] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004.