2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

LANCAR: Leveraging Language for Context-Aware Robot Locomotion in Unstructured Environments

Chak Lam Shek[1*], Xiyang Wu[1*], Wesley A. Suttle[2], Carl Busart[2], Erin Zaroukian[2], Dinesh Manocha[3], Pratap Tokekar[3], and Amrit Singh Bedi[4]

***Abstract*— Navigating robots through unstructured terrains is challenging, primarily due to the dynamic environmental changes. While humans adeptly navigate such terrains by using context from their observations, creating a similar context-aware navigation system for robots is difficult. The essence of the issue lies in the acquisition and interpretation of context information, a task complicated by the inherent ambiguity of human language. In this work, we introduce LANCAR, which addresses this issue by combining a context translator with reinforcement learning (RL) agents for context-aware locomotion. LANCAR allows robots to comprehend context information through Large Language Models (LLMs) sourced from human observers and convert this information into actionable context embeddings. These embeddings, combined with the robot's sensor data, provide a complete input for the RL agent's policy network. We provide an extensive evaluation of LANCAR under different levels of context ambiguity and compare with alternative methods. The experimental results showcase the superior generalizability and adaptability across different terrains. Notably, LANCAR shows at least a 7.4% increase in episodic reward over the best alternatives, highlighting its potential to enhance robotic navigation in unstructured environments. More details and experiment videos could be found in this link.**

I. Introduction

Designing locomotion for quadruped robots has been a longstanding focus of research [1]. The variability of environmental physical properties heavily influences the robot's movement, making it difficult to create a universal policy that works in all situations. Despite significant progress, devising a universal policy capable of effectively addressing all possible scenarios remains elusive [2], [3]. Previous efforts include using graph-like structures [4] or autoencoders [5] to gather context environment information, but these methods often fall short in complex terrain navigation due to limited reasoning capabilities. Addressing these challenges requires innovative approaches, such as combining human insights with technological solutions. Humans can intuitively understand environmental contexts, like associating wet grass with high damping, a concept difficult for current algorithms to grasp. However, leveraging human feedback is complicated by the ambiguity of natural language [6] and the impracticality of expecting humans to provide detailed quantitative descriptions of environments using physical parameters instead of vague qualitative sentences with descriptive words.

The recent success of Large Language Models (LLMs) and their ability to perform chain-of-thought [7], logical reasoning [8], and common sense reasoning [9] offers a promising solution to these challenges. Techniques such as domain randomization [10] have been applied to prepare RL agents for varied conditions. Although prior work has explored using LLMs for predicting RL reward functions [11] or providing robotic control inputs [12], these approaches do not fully utilize LLMs' reasoning capabilities. Positioning LLMs as intermediaries to translate human language into RL-compatible formats could optimize their effectiveness, preventing RL agents' decision-making from being directly influenced by human instructions.

This study explores the use of LLMs to interpret environmental context, enhancing RL agents' ability to guide robot locomotion, particularly for a quadruped robot navigating diverse terrains with the assistance of human observers. The context, often unperceivable directly by the robot, includes terrain characteristics crucial for navigation. Fig. 1 gives an overview of our approach. In Scenario 1, a robot traverses terrains without any context information. The result shows the robot is struggling to formulate a universal navigation policy. In Scenario 2, the robot navigates the same terrains but receives additional context information from human observers,[1] like *"You are entering a grassland right after the rain"* or *"You are walking on a dry rocky road under the sun."* Robots use an LLM-based translator to convert embeddings representing context information from human interpretation, enhancing their decision-making process alongside their sensory observations.

Our method, LANCAR (**LAN**guage for **C**ontext-**A**ware **R**obot locomotion), capitalizes on the versatility of LLMs to interpret human language, transforming it into indices or context embeddings. This process mitigates the ambiguity inherent in human language, enabling robots to navigate varied terrains with adaptable, generalized control policies through collaboration with human observers. While this paper focuses on leveraging LLMs instead of Vision Language Models (VLMs) for context understanding from images, our approach can readily be extended to incorporate VLMs in practical environments. We validate our approach using the *spot-mini-mini* robot simulator *v.2.1.0* [13], showing that LANCAR enhances performance compared to a no-context baseline. Specifically, LANCAR with context embeddings shows at least 7.3% improvement on episodic rewards in low-level tasks and 7.5% improvement in high-level tasks over the

* Denotes equal contribution
[1]Department of Electrical and Computer Engineering, University of Maryland, College Park, MD, USA {cshek1, wuxiyang}@umd.edu
[2]DEVCOM Army Research Laboratory, Adelphi, MD, USA.
[3]Department of Computer Science, University of Maryland, College Park, MD, USA {dmanocha, tokekar}@umd.edu
[4]Department of Computer Science, University of Central Florida, Orlando, FL, USA {amritbedi}@ucf.edu

[1]In some cases vision-language models may also be a useful surrogate to provide context information that is visual

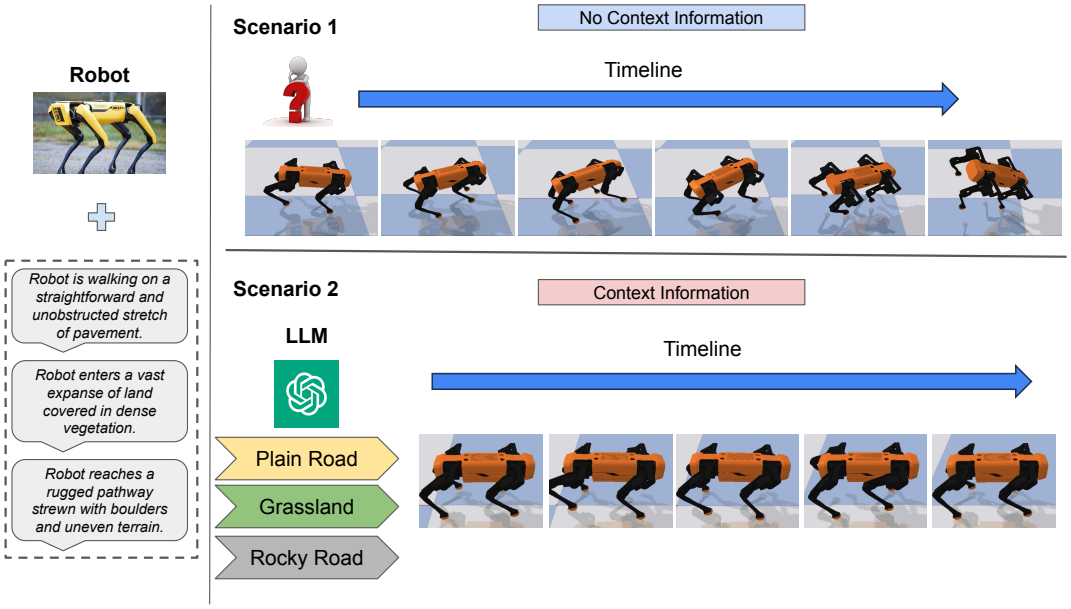

Fig. 1: **Task Description.** We consider two robot learning approaches for locomotion guided by ambiguous human descriptions. The first existing approach (**TOP**) is when the robot moves over diverse terrains with a trained policy without any context information. Given the complexity of the terrains, robots may face difficulties in developing a generalized policy to address all types of terrains, leading to the failure of its ultimate policy. Our proposed approach (**BOTTOM**) has the robot moving over diverse terrains with our trained policy and context information from human observers. Robots convert this interpreted context information into embeddings with LLM. With the extra context information added to robots' own perceptions from their sensors, robots could develop better policies with a better understanding of the environment.

baselines. Additionally, LANCAR using the indexing feature leads to a 1.7% improvement in high-level tasks compared to the no-context approach.

Main Contributions: We summarize our main contributions in this work as follows.

- We propose a novel approach, LANCAR, that incorporates LLMs into RL in robot decision-making that enables robots to understand external context information from human observers and generate a more robust and generalized RL policy.
- We propose an LLM-based context information translator module that interprets *high-level*, ambiguous, human language context information of environments into context information embeddings accessible for RL agents with the reasoning ability of LLMs.
- We evaluate LANCAR with four different backbone RL approaches under 10 case studies using both *low-level* and *high-level* context information. We validate the efficacy of LANCAR in policy generalizability and adaptability across diverse terrains which shows at least 7.3% and 7.5% of performance improvement over established baselines and over 10 times higher episodic reward than ablations using different backbone RL approaches.

II. Related Works

Robot Navigation in Complex Environments. The challenge of reliable robot locomotion and navigation within complex environments requires adaptive policy learning due to the diversity of terrain encountered [14]. The strategies [15], [16], [17] have been developed to address this challenge. NAUTS [18] proposes an approach that makes robots adaptive to off-road diverse terrain with a negotiation process among different navigational policies. VINet [19] uses a novel navigation-based labeling scheme for terrain classification and generalization on both known and unknown surfaces. Ada-Nav [20] presents a novel approach that adaptively tunes policy evaluation trajectory lengths with policy entropy and evaluates this approach in both simulated and real-world outdoor environments. Vision-based approaches for terrain adaptation like ViTAL [21], CMS [22], and RMA [23] use visual observations to generate embeddings that enhance a robot's adaptation capabilities, enabling tasks such as stair climbing or rocky road navigation. However, despite these advances, the current methods are tested within a limited terrain dataset and rely heavily on semantic terrain adaptation strategies, potentially limiting their generalization capabilities in varied real-world terrains.

Human-robot Collaboration. Human-robot collaboration explores how humans and robots can interact effectively to achieve complex tasks, leveraging human cognitive capabilities [24], particularly in unstructured settings [25]. Such human-in-the-loop collaboration with robotics for tasks like trajectory planning [26], and manipulation [27] in challenging environments like surgery [28] and disaster rescue [29]. The integration of LLMs has enhanced human-robot collaboration, enabling robots to draw on human knowledge and reasoning. Ren *et al.* [30] propose an approach that allows robots to seek help from humans with the assistance of LLM. SayTap [31] uses foot contact patterns as the interface between human commands in natural language and a locomotion controller that outputs *low-level* commands. RE-Move [32] uses human-language instructions to help robots avoid obstacles. LM-Nav [33] uses LLM and VLM in object detection for robots' navigation tasks. These developments underscore the utility of LLMs in robot control, but the challenge of achieving policy

generalization across diverse contexts remains unaddressed, marking the primary focus of our investigation.

Language Model for Robotics. The integration of LLMs [34] and Vision Language Models (VLMs) [35] with robotics marks a significant advancement in embodied AI [36], [37]. This fusion allows robots to leverage the commonsense and in-context learning (ICL) of language models [38] in decision-making tasks [39], [40], [41]. Research efforts have enhanced these models' capabilities, such as pre-training for task prioritization [42] and converting complex instructions into detailed tasks with rewards [43]. RT-2 [44], [39] allows manipulators to use the Internet-scale data from the VLMs in their decision-making by taking the action output sequence as another language. Bucker *et al.* [45], [46] use LLMs to allow human language to improve the manipulator trajectories. Mees *et al.* [47] use LLM to decompose the *high-level* tasks into sub-tasks for the robot to execute. Fu *et al.* [48] use LLMs as a driving assistant in autonomous driving tasks. For reinforcement learning, prior works have explored using LLMs in determining reward values [11] and policy explainability in human-AI interaction [49]. Despite these advancements, the specific application of LLMs for interpreting environmental observations and integrating this understanding into RL agents' decision-making processes has not been explored extensively, an area our work aims to address.

III. Methodology

A. Problem Formation

We model the problem as an extension of a Partially Observable Markov Decision Process (POMDP), specifically as an implicit POMDP [50]. An implicit POMDP is specified by a tuple, $\langle \mathcal{S}, \mathcal{A}, \mathcal{O}, \Omega, \mathcal{Z}, \mathcal{F}, \mathcal{T}, \mathcal{R}, \gamma \rangle$, where the state space, $\mathcal{S} = \mathcal{S}_{ex} \cup \mathcal{S}_{im}$, is composed of both explicitly observable states $\mathcal{S}_{ex}$ and implicitly observable states $\mathcal{S}_{im}$. The explicitly observable states are those environmental states directly observable from the agent's onboard sensors. The agent's observation space is $\mathcal{O}$. The observation function is given by $\Omega : \mathcal{S}_{ex} \to \mathcal{O}$. The implicitly observable states are the context information in the environment that cannot be detected directly by the robots but still affect robots' policies. $\mathcal{Z}$ denotes the embedding of the context information from the implicitly observable states $\mathcal{S}_{im}$, while the mapping function between the two is $\mathcal{F} : \mathcal{S}_{im} \to \mathcal{Z}$. Nevertheless, the implicitly observable states (*i.e.* context information) can still be inferred by robots through reasoning over visual perception or tactile sensing or through human language feedback. In this work, our primary focus is to recover $\mathcal{S}_{im}$ using context information given in natural language.

The action space $\mathcal{A}$ represents the agent's feasible actions. The transition function $\mathcal{T} : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$ characterizes the dynamics of the robot within the environment. The reward function $\mathcal{R} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ quantifies the reward of the agent's actions. γ is the discounted factor. The agent's policy π is given by $\pi : \mathcal{O} \times \mathcal{Z} \to \Delta(\mathcal{A})$, while $\Delta(\mathcal{A})$ represents the probability distribution over the action space. We formulate our problem as a finite horizon optimization. The objective is to find an optimal policy π^* that maximizes the expected cumulative reward

$$\pi^* = \arg\max_{\pi} \mathbb{E}_{\pi \sim \{s_t, a_t\}_{t=0}^{H-1}} \sum_{t=0}^{H-1} \gamma^t R(s_t, a_t) \qquad (1)$$

where H is the length of the episode.

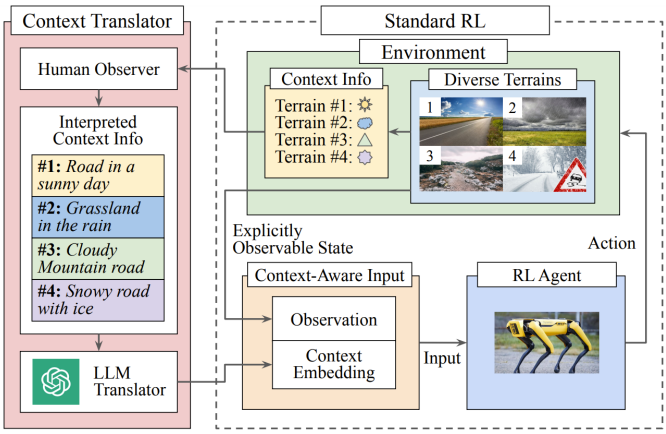

Fig. 2: **Context-Aware Reinforcement Learning Robot Locomotion.** Our framework adds a context translator to the RL setup, enhancing navigation across varied terrains. Agents receive direct observations from the environment, while human observers provide context information, interpreting terrain properties into natural language. The LLM translator processes this into context embeddings, merged with direct observations for RL agent input. The agents then apply their control policies to these enriched inputs to determine and perform actions in the environment.

B. Human-Robot Collaboration Framework

We introduce a human-robot collaboration framework, LANCAR, as depicted in Fig. 2. To recover the context information from environments, we introduce the LLM-based context translator module in addition to the standard RL agent. When robots traverse in environments with diverse terrains at time t, robots observe the environment's explicitly observable states s_{ex}^t, and the human observer interprets the implicitly observable states s_{im} (*i.e.* context information). Here, we assume that the context information is consistent within one episode so that s_{im} is fixed. The human observer provides qualitative descriptions or captions of the context information to the LLM translator. The LLM translator extracts the environmental properties from the context information and generates the context embedding z, which is concatenated with the observations o_t as the input for RL agents. RL agents produce the action a_t using their control policies π given the context-aware inputs and execute the action in the environment. This framework is designed to be compatible with the other RL methods, offering flexibility in its implementation.

The framework is designed delicately to adapt humans' assistance to enhance agents' performance. While it is hypothesized that well-trained agents are better suited to produce a sequence of continuous decisions, direct human control over such well-trained agents may disrupt the decision-making process, potentially leading to degraded performance. On the other hand, human-provided descriptions translated into state estimates over $\mathcal{S}_{im}$ can serve as valuable assistance, enabling the agent to improve its overall performance.

C. LLM-based Context Translator

In our framework, the LLM is pivotal, converting human-interpreted environmental context information into embeddings that RL agents can directly use. We crafted a context translator module leveraging In-context Learning (ICL), enabling the LLM to use its reasoning capabilities through zero-shot or few-shot examples, thus facilitating an interpretable way to interact with the LLM sans explicit training, akin to mimicking human reasoning and decision-making processes [38].

We feed the LLM descriptive sentences about the environment's context information and accompany these with prompts that include examples of potential inputs and outputs the model encounter. An example prompt is presented in Fig. 3. These prompts are structured to guide the LLM in mapping qualitative environmental descriptions to embeddings through a series of multiple-choice questions. Each question pertains to a specific environmental characteristic, with the LLM tasked with selecting from pre-defined qualitative descriptors. The chosen answers are then transformed into concatenated one-hot vectors, creating the context embeddings for RL agents.

By providing in-context examples, we aid the LLM in grasping the task's nature. Upon receiving inputs, the LLM is expected to respond to the questions based on the established format of the in-context examples, ensuring the generation of relevant context embeddings.

Let $\oplus : X \times Y \to [X, Y]$ be an operator concatenating two vectors, and let $\text{onehot}(x)$ denote the function for one-hot encoding. For each property i, it could have different levels in a set P_i which include *Very Low*, *Low*, *Medium*, *High* and *Very High*. Given n properties and their corresponding levels v_{p_i} for property i, the context embedding of this terrain, denoted as C, is obtained by considering the concatenation of the one-hot encoding of the property indexes:

$$C = \text{onehot}(v_{p_1}) \oplus \text{onehot}(v_{p_2}) \oplus \ldots \oplus \text{onehot}(v_{p_n}), \quad (2)$$

For example, if the context information describes the terrain with two properties, saying *This terrain has very low friction and very high damping*. *Very low friction* maps into an one-hot vector $[1, 0, 0, 0, 0]$ and *very high damping* maps into another one-hot vector $[0, 0, 0, 0, 1]$, then the context embedding of this terrain is $[1, 0, 0, 0, 0, 0, 0, 0, 0, 1]$.

We note that our context approach, leveraging human-generated prompts and responses, enables the LLM to effectively bridge the gap between natural language descriptions and actionable state information, a key aspect of our framework's success in recovering context information from unobservable states of environments.

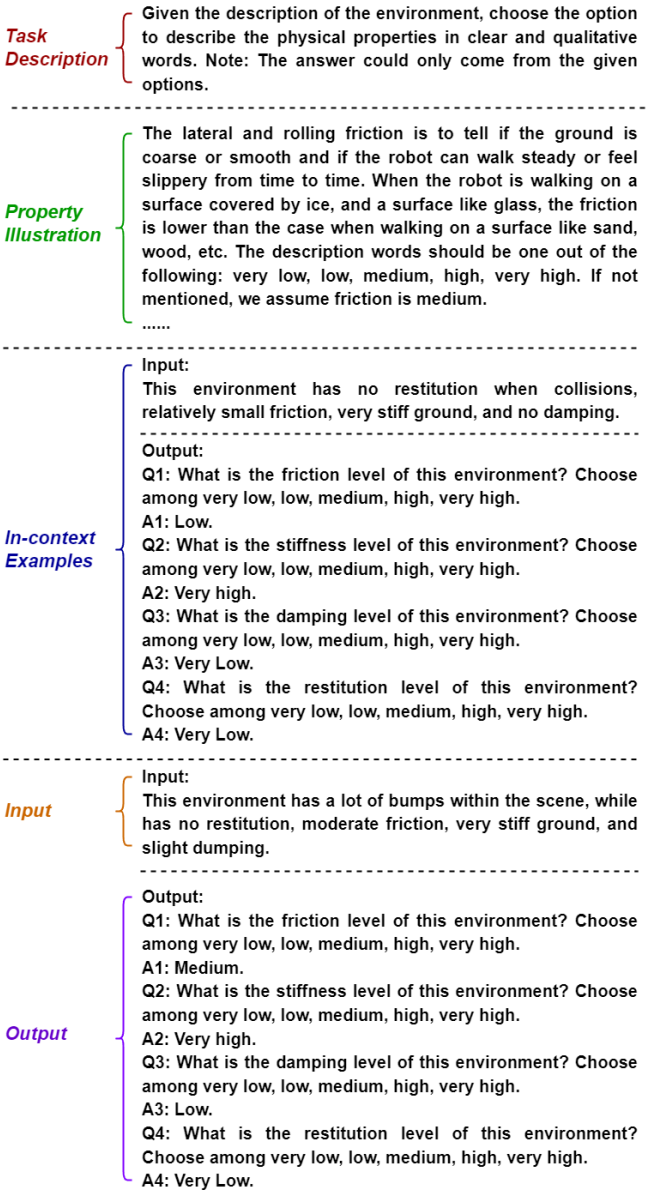

Fig. 3: **An Example Prompt for LANCAR.** The prompt for LANCAR consists of five sections. The first section outlines the *high-level* task for the LLM. The second provides details and examples of relevant terrain properties. The third includes in-context learning examples, featuring *low-level* terrain contexts with outputs derived from multiple-choice question-answering. The final two sections involve presenting inputs to the LLM to generate context embeddings and the corresponding outputs.

D. Reinforcement Learning Agent

In this work, we employ Augmented Random Search (ARS) [51], as the reinforcement learning algorithm for the robot control agent. Both ARS and its ancestor approach, BRS, use the finite difference approach, which approximates the gradient value through derivative sampled in $2N$ directions and updates the network parameters by perturbing policy parameters within the range of $[-\delta, +\delta]$ to assess resulting rewards within that range, while δ is randomly generated from a normal distribution. Compared with BRS, ARS further improves the performance of RL policies by normalization and using top-performing directions to update the network parameters. In addition, ARS uses a linear policy, instead of a non-linear policy like the neural network, to simplify the RL algorithms.

Apart from the ARS approach, we also introduce three widely-used reinforcement learning approaches as our baselines, including SAC [52], PPO [53] and TD3 [54]. More details and discussions are included in Section IV.

IV. Empirical Results and Discussion

In the experiments, we aim to answer the following two questions regarding performance and policy generalization. The first question is: *Does external context information improve the performance of the agent when the agent is operating in diverse conditions?* To answer this question, we designed a series of experiments and compared our framework with alternative approaches that include or exclude context information. In these experiments, each context is a different terrain. The second question we investigate is: *Is the LLM model effective when the given input is* high-level*, open-ended, and ambiguous in retrieving context information and thereby in robot locomotion?*. To answer this question, we use *low-level*, precise, and organized human interpretation of context information in training but also use *high-level*, vague, and unorganized context information from human observers in evaluation, apart from the *low-level* context information evaluation cases.

We use GPT-4 [55] as our LLM model to interpret both *low-level* and *high-level* human instructions for robots into robot-understandable embeddings with a series of formulated multiple-choice questions. All reinforcement learning agents are trained under domain randomization manner with 8 different scenarios, all of which have diverse environmental properties. The episodic reward curve during training is present in Fig. 4. After the training process, agents are evaluated under 10 evaluation cases with 5 cases using *low-level* context information from the environment and 5 cases using *high-level* context information. All evaluation results are averaged over 16 episodes.

A. Environments

We use a quadruped robot locomotion simulator, *spot-mini-mini v.2.1.0* [13], built in PyBullet [56]. The robot's goal is to advance along the x-axis as much as possible within a fixed set timeframe of 5,000 steps, minimizing deviation from this axis. The raw observation state from the environment is 16–dimensional, including the robot's roll, pitch, gyroscopes, and acceleration, as well as a 4-length binary vector denoting the robot's leg contacts with the ground. The extra observation state, depending on the embedding mechanism covered in Section IV-D, is also provided to the agent. The action space is the desired joint angle for each of the 14 joints that are clipped within the maximum allowed velocities. The reward function is given by the combination of multiple terms corresponding to the robot's state, including the robot's traveling distance d_x, denoting the accomplished distance from the origin in x-axis, the penalty d_y that represents the robot's deviation from the y axis, and the penalty r_p that happens when the robot does not keep the desired rate. The reward function is defined as $J = d_x + 0.03d_y + 10r_p$. For the ARS agent using in LANCAR, the learning rate is 0.03. The number of samples for δ is 16. The noise amplitude applied in the exploration is 0.05.

Manually crafting context information for each training instance is unfeasible due to the vast amount of data. Instead, we employ an LLM to automate the generation of detailed, low-level terrain descriptions. This process starts with generating random samples of the terrain's parameters quantitatively describing properties given in Table I, followed by a prompt instructing the LLM to translate these values into qualitative descriptions—ranging from *Very Low* to *Very High*, based on 18 in-context learning examples. Given the actual parameter values, we ask the LLM to generate a *low-level* context description. A sample description generated during training is: *This environment has very low restitution when collision, low friction, very high stiffness level, and very low damping.*

TABLE I: Properties for Training Terrains

Property	Value Range
Restitution	$[0, 0.2]$
Lateral / Horizontal Friction	$[0, 1]$
Rolling Friction	$[2 \times 10^4, 1.6 \times 10^5]$
Stiffness	$[0, 1]$
Damping Coefficient	$[0, 0.5]$

B. Evaluation Cases

In the evaluation phase, we conduct two case study experiments with increasing difficulty levels, *i.e.* increasing vagueness of the context information provided to the LLM, to examine the reasoning ability of our approach. Specifically, we evaluate the following two types of contexts (in increasing order of open-endedness):

1) Low-Level Context: The context information provided by human observers during evaluation gives detailed qualitative descriptions of environmental properties, the same as those given in the training phase. Descriptions for all five evaluation cases we used are provided in Table II.

TABLE II: Low-Level Context Information for Case Study

ID	Context Information
A	"This environment has no restitution when collision, very high friction, and no damping."
B	"This environment has no restitution when collision, very low friction, and no damping."
C	"This environment has high restitution when collision, very high friction, and very high damping."
D	"This environment has medium restitution when collision, low friction, and very high damping."
E	"This environment has high restitution when collision, very high friction, and low stiffness."

2) High-Level Context: The context description provided by human observers is *high-level*, open-ended, vague, and descriptive of the environmental conditions, as opposed to the environmental properties. Descriptions for all five evaluation cases we used are provided in Table III.

TABLE III: High-Level Context Information for Case Study

ID	Name	Context Information
F	Moist Grassland	"The spot is walking on a grassland under a drizzle."
G	Snowy Mountain Road	"The spot is walking on a mountain road covered by ice. It's snowy now."
H	Sunny Beach	"The spot is walking on the beach near the sea under the sun."
I	Rainy Concrete Road	"The spot is walking on a concrete road under heavy rain."
J	Sunny Running Tracks	"The spot is walking on running tracks on a sunny day."

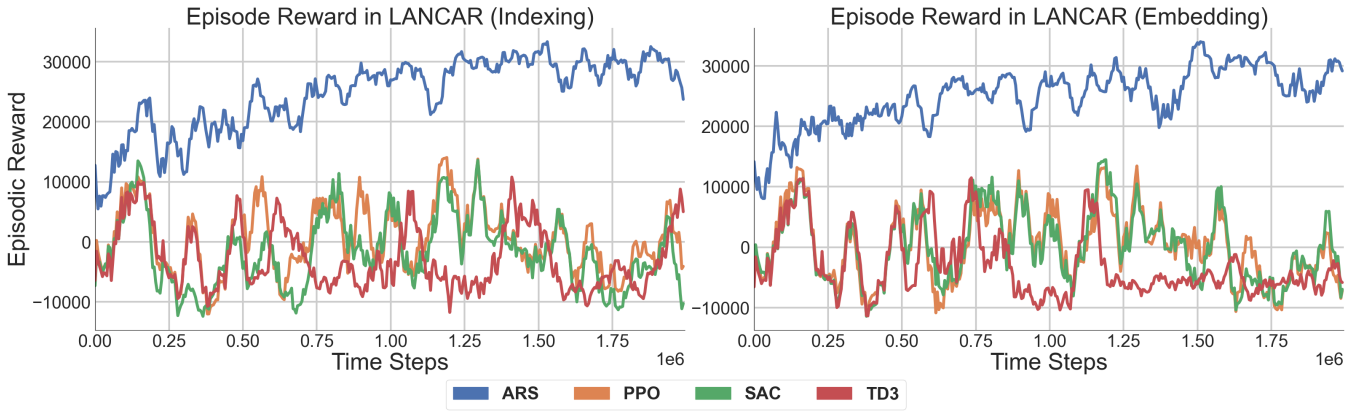

Fig. 4: **Episodic Reward Curve for LANCAR (Indexing) and LANCAR (Embedding) with Different Backbones.** All results are run over 2 million time steps while each episode has 5000 time steps in maximum. **Conclusion.** Both LANCAR (Indexing) and LANCAR (Embedding) have the highest episodic reward when using ARS (blue) backbone than using other backbone approaches.

TABLE IV: **Average Episodic Reward of LANCAR and No-context RL.** We perform evaluation experiments across all baselines and ablation studies over 10 cases (5 *low-level* context cases and 5 *high-level* context cases). **Conclusion.** ARS-based approaches achieve much higher episodic rewards than all other baselines. ARS using LANCAR embeddings for context information have a better performance than all other approaches in most cases.

		Low-Level Context					High-Level Context				
Method	Backbone	A	B	C	D	E	F	G	H	I	J
No-Context ($\times 10^3$)	ARS	36.628	19.698	38.000	28.573	30.744	35.545	13.051	29.819	34.053	33.934
	SAC	24.189	-10.128	15.571	-10.839	-11.457	9.461	-7.123	-10.076	18.252	-3.994
	TD3	25.001	-6.756	17.768	-12.230	-11.726	9.833	-9.445	-12.450	19.352	-3.583
	PPO	7.542	-8.266	-1.249	-10.159	-10.073	4.534	-7.262	-10.637	15.798	-2.181
LANCAR (Indexing) ($\times 10^3$)	ARS	36.659	**23.435**	38.366	20.649	22.952	37.791	**16.265**	22.776	36.676	35.357
	SAC	16.423	-9.695	14.534	-12.199	-12.443	7.521	-7.592	-12.012	16.252	-5.815
	TD3	20.867	-7.665	15.734	-11.672	-11.612	7.955	-7.328	-13.414	17.089	-4.131
	PPO	24.119	-8.343	11.851	-8.520	-9.498	10.937	-10.980	-10.333	19.934	-2.009
LANCAR (Embedding) ($\times 10^3$)	ARS	**41.220**	20.706	**41.725**	**29.545**	**31.595**	**40.563**	12.162	**30.961**	**39.722**	**36.623**
	SAC	12.154	-8.648	17.251	-9.413	-11.159	8.381	-7.197	-12.599	16.176	-5.970
	TD3	20.714	-8.655	17.788	-9.138	-11.022	8.587	-6.465	-12.478	15.772	-16.800
	PPO	12.979	-9.449	5.512	-9.187	-10.314	8.345	-9.533	-9.391	15.607	-8.148

C. Baselines

We have considered the following baseline algorithms: Augmented Random Search (ARS) [51], Soft Actor-Critic (SAC) [52], Proximal Policy Optimization (PPO) [53], and Twin Delayed DDPG (TD3) [54]. These algorithms are commonly applied to solve a variety of control tasks. ARS is a derivative-free optimization algorithm that explores the parameter space through random perturbations to improve policy performance. SAC leverages an off-policy approach to optimize the policy while also estimating the value function. PPO employs a policy gradient method with a clipping mechanism to ensure smooth policy updates and prevent large policy changes. TD3 is an extension of the Deep Deterministic Policy Gradient (DDPG) algorithm [57], integrating twin critics and target policy to address overestimation bias and improve the robustness of the learned policy.

D. Ablation Study

We conduct a series of experiments on our approach, LANCAR, and some baseline approaches, to evaluate the effect of the usage and design of context information embedding strategies. We evaluate the following approaches:

1) No-Context: The RL agent only uses environmental observation in their decision-making. No context information is used by the RL agent. The decision does not rely on the LLM output. It will be used as the baseline of the experiment.

2) LANCAR (Indexing): The context is encoded as a one-hot vector identifying the environment. The RL agent labels all terrains encountered during training with a unique index. The one-hot vector sets the i-th element of the vector as 1 and is used as the embedding for the RL agent, denoting the i-th training terrain. During the evaluation, the indexing embedding vector is replaced with an all-zero vector as padding for the RL agent.

3) LANCAR (Embedding): This is the approach we propose in this work. The LLM generates context embeddings by interpreting human language instruction in the way presented in Section III-C, and the RL agent incorporates context embeddings with environmental observation in their decision-making. The context embeddings are represented with a combination of multiple one-hot vectors. Each one-hot vector quantifies properties in Table I into five intervals.

E. Results

1) Case Study: Low-Level Context Information: The evaluation involved testing with low-level context information across five different terrain cases: normal terrain (Case A), low friction (Case B), high damping (Case C), medium restitution with very high damping (Case D), and high restitution and damping with low stiffness (Case E). Generally, terrains with low friction, low stiffness, and high damping present greater challenges for RL-controlled robots, with restitution level variations increasing locomotion task uncertainty.

Table IV (Case A-E) shows the evaluation results of episodic reward over all cases using *low-level* context information. We find that LANCAR (Embedding) using an ARS backbone and Embedding method outperformed the other two approaches. LANCAR (Embedding) achieved 16.0% higher episodic rewards than LANCAR (Indexing) and 7.3% higher episodic rewards than the No-Context baseline. The LANCAR (Indexing) approach, while slightly underperforming compared to the No-Context baseline by 7.5% in episodic reward, showed variable performance across terrains, outperforming LANCAR (Embedding) in the low friction scenario (Case B) but falling behind in all other cases. This variability suggests a limited adaptation capability of the Indexing method to different terrains and context inputs. Notably, all methods employing an ARS backbone demonstrated better performance (approximately $17.0\times$ higher episodic reward) than those with other backbones across all evaluation scenarios, underscoring the superior adaptability of approaches with ARS backbone in this context.

2) Case Study: High-Level Context Description: In our second case study, we explored *high-level* context scenarios: Moist Grassland (Case F), Snowy Mountain Road (Case G), Sunny Beach (Case H), Rainy Concrete Road (Case I), and Sunny Running Tracks (Case J). These terrains, characterized by complex combinations of surface properties like stiffness, damping, and friction, present more challenging conditions than those in the *low-level* context study.

Table IV (Case F-J) shows the evaluation results of episodic reward over 5 *high-level* evaluation cases. Evaluation results show that LANCAR (Embedding) with an ARS backbone performed better than both LANCAR (Indexing) with 9.3% higher episodic rewards and the No-Context setups with 7.5% higher episodic rewards. LANCAR (Indexing) surpassed LANCAR (Embedding) in Case G (Snowy Mountain Road), indicating particular adaptability to low-friction conditions, but LANCAR (Embedding) excelled in all other scenarios. Approaches using the ARS backbone consistently outperformed those with different backbones across all tested terrains. Furthermore, the heightened challenge of these high-level context terrains accentuated the performance gap between ARS-backed methods and other approaches.

F. Discussion

Our experiments showed that context-aware strategies, LANCAR (Indexing) and LANCAR (Embedding), consistently surpassed the baseline no-context approach in performance. LANCAR (Embedding) generally yielded superior results across diverse scenarios, showcasing its adaptability, whereas LANCAR (Indexing) excelled in specific situations. This discrepancy is attributed to the limited range of training scenarios, which may not encompass a broad spectrum of physical properties, leading to some evaluation cases falling outside the training domain. Expanding the variety of training terrains could address this issue by enhancing the model's exposure to different environments. Notably, LANCAR (Embedding) demonstrated a more significant performance boost over no-context baselines with high-level contexts, suggesting that the LLM's interpretation of context information effectively mitigates environmental ambiguity and the complexities of locomotion tasks. Additionally, methods utilizing the ARS backbone outperformed all alternatives across every scenario, affirming their superior adaptability across a wide array of evaluation conditions.

V. Conclusion

This paper introduces a method allowing human observers to use natural language for conveying environmental context to robots, with LLMs translating this into context embeddings for RL agents. These embeddings, combined with the agents' observations, enhance navigation strategies.

Looking ahead, we aim to evolve our methodology by incorporating visual sensors and foundation models for interpreting environmental context, aiming for more accurate object captions within robot-perceived images. Another potential extension of our work is to leverage multi-modal foundation models for context understanding through different sensors, directly creating embeddings understandable by robots. Besides, we plan to explore mechanisms for enhancing robot adaptability across different contexts within the same episode, particularly for outdoor navigation tasks, aiming to improve the robustness and adaptiveness of real-world robot locomotion strategies.

References

[1] M. H. Raibert, *Legged robots that balance*. USA: Massachusetts Institute of Technology, 1986.

[2] S. Josef and A. Degani, "Deep reinforcement learning for safe local planning of a ground vehicle in unknown rough terrain," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 6748–6755, 2020.

[3] X. Xiao, Z. Wang, Z. Xu, B. Liu, G. Warnell, G. Dhamankar, A. Nair, and P. Stone, "Appl: Adaptive planner parameter learning," *Robotics and Autonomous Systems*, vol. 154, p. 104132, 2022.

[4] J. Liang, Z. Wang, Y. Cao, J. Chiun, M. Zhang, and G. A. Sartoretti, "Context-aware deep reinforcement learning for autonomous robotic navigation in unknown area," in *7th Annual Conference on Robot Learning*, 2023.

[5] H. Karnan, E. Yang, D. Farkash, G. Warnell, J. Biswas, and P. Stone, "Self-supervised terrain representation learning from unconstrained robot experience," in *ICRA2023 Workshop on Pretraining for Robotics (PT4R)*, 2023.

[6] H. Ahn, S. Choi, N. Kim, G. Cha, and S. Oh, "Interactive text2pickup networks for natural language-based human–robot collaboration," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3308–3315, 2018.

[7] J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, D. Zhou, *et al.*, "Chain-of-thought prompting elicits reasoning in large language models," *Advances in Neural Information Processing Systems*, vol. 35, pp. 24824–24837, 2022.

[8] A. Creswell, M. Shanahan, and I. Higgins, "Selection-inference: Exploiting large language models for interpretable logical reasoning," *arXiv preprint arXiv:2205.09712*, 2022.

[9] M. Geva, D. Khashabi, E. Segal, T. Khot, D. Roth, and J. Berant, "Did aristotle use a laptop? a question answering benchmark with implicit reasoning strategies," *Transactions of the Association for Computational Linguistics*, vol. 9, pp. 346–361, 2021.

[10] 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.

[11] M. Kwon, S. M. Xie, K. Bullard, and D. Sadigh, "Reward design with language models," *arXiv preprint arXiv:2303.00001*, 2023.

[12] S. Mirchandani, F. Xia, P. Florence, B. Ichter, D. Driess, M. G. Arenas, K. Rao, D. Sadigh, and A. Zeng, "Large language models as general pattern machines," *arXiv preprint arXiv:2307.04721*, 2023.

[13] M. Rahme, I. Abraham, M. Elwin, and T. Murphey, "Spotminimini: Pybullet gym environment for gait modulation with bezier curves," 2020.

[14] M. Elnoor, A. J. Sathyamoorthy, K. Weerakoon, and D. Manocha, "Pronav: Proprioceptive traversability estimation for autonomous legged robot navigation in outdoor environments," *arXiv preprint arXiv:2307.09754*, 2023.

[15] S. LaValle, "Planning algorithms," *Cambridge University Press google schola*, vol. 2, pp. 3671–3678, 2006.

[16] J. Canny, *The complexity of robot motion planning*. MIT press, 1988.

[17] D. Manocha, *Algebraic and numeric techniques in modeling and robotics*. University of California, Berkeley, 1992.

[18] S. Siva, M. Wigness, J. G. Rogers, L. Quang, and H. Zhang, "Nauts: Negotiation for adaptation to unstructured terrain surfaces," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1733–1740, IEEE, 2022.

[19] T. Guan, R. Song, Z. Ye, and L. Zhang, "Vinet: Visual and inertial-based terrain classification and adaptive navigation over unknown terrain," in *2023 IEEE international conference on robotics and automation (ICRA)*, pp. 4106–4112, IEEE, 2023.

[20] B. Patel, K. Weerakoon, W. A. Suttle, A. Koppel, B. M. Sadler, A. S. Bedi, and D. Manocha, "Ada-nav: Adaptive trajectory-based sample efficient policy learning for robotic navigation," *arXiv preprint arXiv:2306.06192*, 2023.

[21] S. Fahmi, V. Barasuol, D. Esteban, O. Villarreal, and C. Semini, "Vital: Vision-based terrain-aware locomotion for legged robots," *IEEE Transactions on Robotics*, vol. 39, no. 2, pp. 885–904, 2022.

[22] A. Loquercio, A. Kumar, and J. Malik, "Learning visual locomotion with cross-modal supervision," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 7295–7302, IEEE, 2023.

[23] A. Kumar, Z. Fu, D. Pathak, and J. Malik, "Rma: Rapid motor adaptation for legged robots," *arXiv preprint arXiv:2107.04034*, 2021.

[24] A. E. Leeper, K. Hsiao, M. Ciocarlie, L. Takayama, and D. Gossow, "Strategies for human-in-the-loop robotic grasping," in *Proceedings of the seventh annual ACM/IEEE international conference on Human-Robot Interaction*, pp. 1–8, 2012.

[25] A. Ajoudani, A. M. Zanchettin, S. Ivaldi, A. Albu-Schäffer, K. Kosuge, and O. Khatib, "Progress and prospects of the human–robot collaboration," *Autonomous Robots*, vol. 42, pp. 957–975, 2018.

[26] A. P. Dani, I. Salehi, G. Rotithor, D. Trombetta, and H. Ravichandar, "Human-in-the-loop robot control for human-robot collaboration: Human intention estimation and safe trajectory tracking control for collaborative tasks," *IEEE Control Systems Magazine*, vol. 40, no. 6, pp. 29–56, 2020.

[27] M. Raessa, J. C. Y. Chen, W. Wan, and K. Harada, "Human-in-the-loop robotic manipulation planning for collaborative assembly," *IEEE Transactions on Automation Science and Engineering*, vol. 17, no. 4, pp. 1800–1813, 2020.

[28] E. Fosch-Villaronga, P. Khanna, H. Drukarch, and B. H. Custers, "A human in the loop in surgery automation," *Nature Machine Intelligence*, vol. 3, no. 5, pp. 368–369, 2021.

[29] M. DeDonato, V. Dimitrov, R. Du, R. Giovacchini, K. Knoedler, X. Long, F. Polido, M. A. Gennert, T. Padır, S. Feng, *et al.*, "Human-in-the-loop control of a humanoid robot for disaster response: a report from the darpa robotics challenge trials," *Journal of Field Robotics*, vol. 32, no. 2, pp. 275–292, 2015.

[30] A. Z. Ren, A. Dixit, A. Bodrova, S. Singh, S. Tu, N. Brown, P. Xu, L. Takayama, F. Xia, J. Varley, *et al.*, "Robots that ask for help: Uncertainty alignment for large language model planners," *arXiv preprint arXiv:2307.01928*, 2023.

[31] Y. Tang, W. Yu, J. Tan, H. Zen, A. Faust, and T. Harada, "Saytap: Language to quadrupedal locomotion," *arXiv preprint arXiv:2306.07580*, 2023.

[32] 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.

[33] D. Shah, B. Osiński, S. Levine, *et al.*, "Lm-nav: Robotic navigation with large pre-trained models of language, vision, and action," in *Conference on Robot Learning*, pp. 492–504, PMLR, 2023.

[34] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, *et al.*, "Language models are few-shot learners," *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.

[35] K. Zhou, J. Yang, C. C. Loy, and Z. Liu, "Learning to prompt for vision-language models," *International Journal of Computer Vision*, vol. 130, no. 9, pp. 2337–2348, 2022.

[36] H. Fan, X. Liu, J. Y. H. Fuh, W. F. Lu, and B. Li, "Embodied intelligence in manufacturing: leveraging large language models for autonomous industrial robotics," *Journal of Intelligent Manufacturing*, pp. 1–17, 2024.

[37] V. S. Dorbala, J. F. Mullen Jr, and D. Manocha, "Can an embodied agent find your "cat-shaped mug"? llm-based zero-shot object navigation," *IEEE Robotics and Automation Letters*, 2023.

[38] Q. Dong, L. Li, D. Dai, C. Zheng, Z. Wu, B. Chang, X. Sun, J. Xu, and Z. Sui, "A survey for in-context learning," *arXiv preprint arXiv:2301.00234*, 2022.

[39] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, X. Chen, K. Choromanski, T. Ding, D. Driess, A. Dubey, C. Finn, *et al.*, "Rt-2: Vision-language-action models transfer web knowledge to robotic control," *arXiv preprint arXiv:2307.15818*, 2023.

[40] J. Liang, P. Gao, X. Xiao, A. J. Sathyamoorthy, M. Elnoor, M. C. Lin, and D. Manocha, "Mtg: Mapless trajectory generator with traversability coverage for outdoor navigation," 2024.

[41] A. Padalkar, A. Pooley, A. Jain, A. Bewley, A. Herzog, A. Irpan, A. Khazatsky, A. Rai, A. Singh, A. Brohan, *et al.*, "Open x-embodiment: Robotic learning datasets and rt-x models," *arXiv preprint arXiv:2310.08864*, 2023.

[42] M. Ahn, A. Brohan, N. Brown, Y. Chebotar, O. Cortes, B. David, C. Finn, C. Fu, K. Gopalakrishnan, K. Hausman, *et al.*, "Do as i can, not as i say: Grounding language in robotic affordances," *arXiv preprint arXiv:2204.01691*, 2022.

[43] W. Yu, N. Gileadi, C. Fu, S. Kirmani, K.-H. Lee, M. G. Arenas, H.-T. L. Chiang, T. Erez, L. Hasenclever, J. Humplik, *et al.*, "Language to rewards for robotic skill synthesis," *arXiv preprint arXiv:2306.08647*, 2023.

[44] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, J. Dabis, C. Finn, K. Gopalakrishnan, K. Hausman, A. Herzog, J. Hsu, *et al.*, "Rt-1: Robotics transformer for real-world control at scale," *arXiv preprint arXiv:2212.06817*, 2022.

[45] A. Bucker, L. Figueredo, S. Haddadinl, A. Kapoor, S. Ma, and R. Bonatti, "Reshaping robot trajectories using natural language commands: A study of multi-modal data alignment using transformers," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 978–984, IEEE, 2022.

[46] A. Bucker, L. Figueredo, S. Haddadin, A. Kapoor, S. Ma, S. Vemprala, and R. Bonatti, "Latte: Language trajectory transformer," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 7287–7294, IEEE, 2023.

[47] O. Mees, J. Borja-Diaz, and W. Burgard, "Grounding language with visual affordances over unstructured data," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 11576–11582, IEEE, 2023.

[48] D. Fu, X. Li, L. Wen, M. Dou, P. Cai, B. Shi, and Y. Qiao, "Drive like a human: Rethinking autonomous driving with large language models," *arXiv preprint arXiv:2307.07162*, 2023.

[49] H. Hu and D. Sadigh, "Language instructed reinforcement learning for human-ai coordination," *arXiv preprint arXiv:2304.07297*, 2023.

[50] D. Ghosh, J. Rahme, A. Kumar, A. Zhang, R. P. Adams, and S. Levine, "Why generalization in rl is difficult: Epistemic pomdps and implicit partial observability," *Advances in Neural Information Processing Systems*, vol. 34, pp. 25502–25515, 2021.

[51] H. Mania, A. Guy, and B. Recht, "Simple random search provides a competitive approach to reinforcement learning," *arXiv preprint arXiv:1803.07055*, 2018.

[52] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *International conference on machine learning*, pp. 1861–1870, PMLR, 2018.

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

[54] S. Fujimoto, H. Hoof, and D. Meger, "Addressing function approximation error in actor-critic methods," in *International conference on machine learning*, pp. 1587–1596, PMLR, 2018.

[55] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat, *et al.*, "Gpt-4 technical report," *arXiv preprint arXiv:2303.08774*, 2023.

[56] E. Coumans and Y. Bai, "Pybullet, a python module for physics simulation for games, robotics and machine learning." http://pybullet.org, 2016–2021.

[57] T. Lillicrap, "Continuous control with deep reinforcement learning," *arXiv preprint arXiv:1509.02971*, 2015.