Warming-up and Shaping Robot Policy Learning with Large Language Model-Based Critic Feedback

Yohei Hayamizu[1†], Tongxu Ai[2†], Ting Hou[2], Guangliang Li[2*], Shiqi Zhang[1]

***Abstract*— Reinforcement Learning (RL) enables robots to learn to perform tasks, but suffers from poor sample efficiency, especially in complex and high-dimensional environments. Due to the rich commonsense knowledge and advanced reasoning capabilities, recent work has explored the integration of robot learning via RL with large language models (LLMs). In this paper, we present Value Initialization and Adaptive Shaping (VIAS), a framework that uses large language models as external critics to provide guidance for value estimation. VIAS is expected to enhance the sample efficiency of robot policy learning via RL by warming-up and shaping its value estimation. We evaluated VIAS in a VirtualHome environment and on two real-world robot platforms in both motion planning and task planning tasks. Experimental results show that VIAS outperforms state-of-the-art methods in both learning speed and task completion. In addition, further analysis shows that VIAS remains effective regardless of the specific LLM model employed and outperforms the LLM directly generated policy, demonstrating its potential for real-world robot applications. Video result and more details are available at https://anonymousname123741.github.io/VIAS.github.io/.**

I. INTRODUCTION

Recent advances in robot learning have enabled robots to perform increasingly complex and long-horizon tasks, bringing them into our daily lives [1], [2], [3]. While reinforcement learning (RL) has been central to this success, it continues to face challenges such as poor sample efficiency and complex reward design. These challenges are particularly pronounced in tasks where state and action information is conveyed in natural language, such as household instruction following. In such scenarios, an agent must learn not only how to act but also how to interpret goal descriptions and justify its behaviors.

Various studies have explored solutions through representation learning, exploration strategies, and the use of prior knowledge [4], [5], [6], [7], [8]. For daily activity tasks, knowledge is often provided in the form of natural language, and prior work has attempted to use this information directly for reward shaping or policy learning [9], [10], [11]. In this context, recent large language models (LLMs) have emerged as powerful candidates for this source of prior knowledge, due to their rich commonsense knowledge and advanced reasoning capabilities [12], [13]. However, attempting to directly integrate LLM reasoning into an agent's policy or reward function leads to a critical problem: its static knowledge cannot keep up with the dynamic changes of the real world [14], [15].

[1] SUNY Binghamton; [2] Ocean University of China

† Equal Contribution. * Corresponding author. This work was supported by Young Taishan Scholars Program of Shandong Province (Grant No. tsqn202408072).

To address these limitations, we propose Value Initialization and Adaptive Shaping (VIAS), a framework that leverages LLMs as external critics to enhance the sample efficiency of robot policy learning via RL. Rather than using LLMs to generate actions or as complex reward-generating code, which solely shapes immediate rewards, we focus on taking LLMs to provide guidance on the cumulative results that states or state-action pairs can bring about. This critic feedback in the form of V-values (state-values) or Q-values (state-action-values), can be used to warm-up robot policy learning and adaptively shape value updates online during the learning process. By anchoring learning with estimates sourced from the robot's own exploration, VIAS is expected to enable the robot to prioritize relevant actions and accelerate policy learning while preserving adaptability.

We validated the effectiveness of our proposed method, VIAS, in language based task planning of a simulated household environment (VirtualHome), which requires commonsense knowledge for everyday activities. Moreover, we conducted real-world evaluations in both motion planning and task planning robotic tasks: a grab and put task with a 6-DOF Ned robot arm, and a pick and place manipulation task with a mobile robot manipulator. The motion-level policies learned via VIAS can also serve as he predefined set of fundamental skills for task planning in VirtualHome and real-world mobile robot manipulation task. The results of our experiments demonstrate that VIAS can improve the sample efficiency of robot policy learning and outperform conventional RL approaches and other LLM-based shaping methods. Further analysis shows that while the choice of LLM can influence the degree of performance enhancement, VIAS remains effective regardless of the specific model employed.

II. RELATED WORK

Incorporating prior knowledge in robotic policy learning is crucial to tackle issues such as sample inefficiency, safety constraints, and the reality gap. Imitation learning [16], [17], behavior cloning [18], and reward shaping [19], [20] are widely used techniques to guide exploration using expert data or domain-specific signals. Human-in-the-loop feedback methods such as TAMER [21] and declarative knowledge-based shaping [22] have been shown to be able to improve the agent's learning speed in constrained environments. However, these methods often require expensive demonstrations or human-delivered reward feedback, which is costly and time-consuming for human expert to provide feedback for

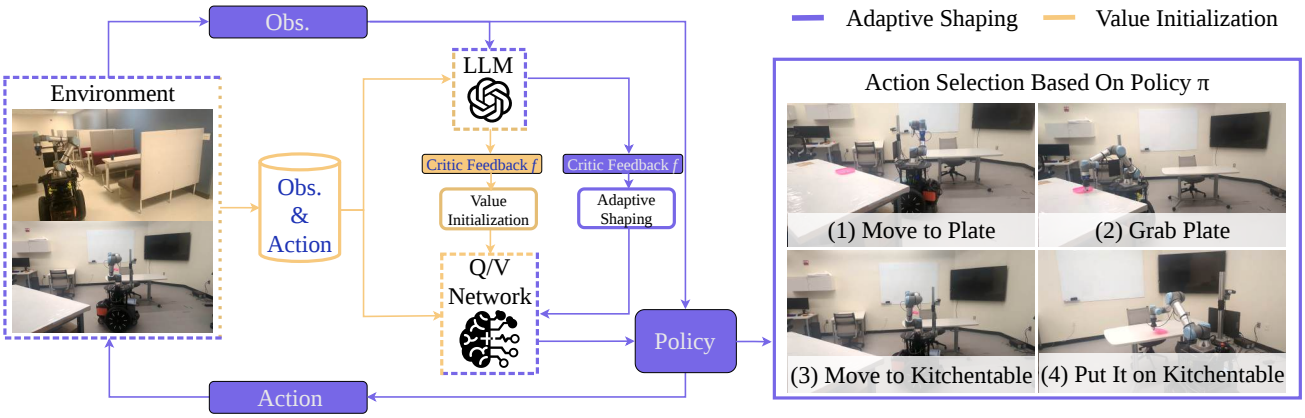

Fig. 1. Overview of the mechanism of our proposed VIAS: The critic feedback from an LLM is used to warm-up and shape the value estimation of robot learning via RL.

each agent behavior and relatively high-quality demonstrations in various tasks, especially in complex ones.

The use of LLMs for decision-making in robotics has garnered increasing attention [23], [24]. Approaches like SayCan [25] integrate LLMs with affordance grounding to generate executable action plans. Frameworks like VOYAGER [26] enable agents to realize general-purpose capabilities using chain-of-thought prompting [27], [28]. LLMs have also been employed as world models that predict future states [29], [30]. While these works use LLMs to select actions or model environments, our approach departs from direct control and instead queries LLMs for value estimates, allowing the RL agent to benefit from informed priors without treating the LLM as a black-box policy.

Recent research has also explored using LLMs to automate reward design in reinforcement learning, aiming to reduce dependence on handcrafted signals and domain expertise. Approaches such as Text2Reward [31] and Eureka [11] translate natural language task descriptions into dense, high-quality rewards, with Eureka demonstrating superior performance across robotic morphologies, including dexterous manipulation. Other methods focus on generating reward functions from high-level instructions [32], [33], or shaping rewards that serve as auxiliary signals to guide learning [34]. EAGER [35] assesses trajectory quality through LLM-driven question-answering, while ELLM [33] incentivizes agents based on LLM-generated suggestions. Similarly, Q-Shaping employs a complex three-phase optimization process by using LLM-generated two categories imprecise Q value (i.e. positive and negative) to update Q network and policy in the early learning stage [36]. Instead of enhance learning by providing synthetic rewards as the above methods, our VIAS bypasses reward construction entirely by querying LLMs for precise critic feedback, and can be used to warm-up robot policy learning and adaptively shape value updates online during the whole learning process.

III. Proposed Method—VIAS

We propose a Value Initialization and Adaptive Shaping (VIAS) framework that leverages LLMs as external critics to enhance the sample efficiency of robot policy learning via RL, as shown in Fig. 1. VIAS consists of two main components: Value Initialization (VI) that uses LLM-generated critic feedback to warm-up the robot's value function before learning, and Adaptive Shaping (AS) that continuously shapes value estimation during learning. By anchoring learning with guidance from LLMs, VIAS is expected to enable the robot to prioritize relevant actions and accelerate policy learning while preserving adaptability.

In the following sections, we present details of these components, including how the critic feedback values are obtained from the LLM and incorporated into the robot learning process.

A. Critic Feedback from LLM

We utilize LLMs (e.g., GPT-4o [37]) to generate critic feedback values f, representing the expected cumulative reward for a given observation-action pair (o, a) or state s alone. That is to say, the critic feedback value generated by LLM can be V-value of states and Q value of state-action pairs. For example, we can prompt the LLM with natural language descriptions of the current observation o along with a set of candidate actions $\{a_0, a_1, \ldots, a_n\}$, asking it to estimate the long-term utility of each action, i.e., the critic Q value.

To promote diversity in the LLM feedback, especially for unseen observations during evaluation, we employ a template-based prompting strategy that adapts to various objects and contexts in the environment. This template allows us to generate a wide range of observation descriptions by varying the entities and spatial relationships described, enabling the robot to receive critic feedback across a rich set of possible states.

To calibrate the model's outputs, we employ few-shot prompting with three annotated examples. We set the temperature to 0.0 and disable top-k sampling to ensure deterministic and stable predictions. Prompts are batched and cached during training to reduce redundancy and minimize computational cost. Crucially, we adopt a numerical feedback scheme rather than a qualitative one. While previous work

has used binary labels or ordinal Likert-style ratings [38], [32], we found such discrete scales inadequate for capturing the fine-grained gradations required in long-horizon tasks. In these settings, it is important to distinguish states or state-action pairs that are close to or far from the goal. A numerical scale allows us to more precisely assign lower values to distant steps and higher values to near-goal states. Accordingly, the LLM is instructed to output a numerical value in the range $[0, 1]$ for each observation or observation-action pair. We discard outputs that are non-numeric or fall outside the valid range, and substitute a fallback heuristic (e.g., zero) in such cases.

B. Robot Learning with LLM Critic Feedback

VIAS allows robot learning from LLM-generated critic feedback in two phases: Value Initialization (VI) that uses LLM-generated critic feedback to warm-up the robot's value function before learning, and Adaptive Shaping (AS) that continuously shapes value estimation during learning.

In the case of querying LLM for critic Q value feedback, in the phase of VI, we construct a buffer $\mathcal{D}_0$ of observation-action pairs (o, a) along with their critic feedback values f before learning. Specifically, we randomly generate a broader set of observations, query the LLM using a template prompt to obtain critic feedback values for a set of actions, and parse the output into tuples (o, a, f). Note that we leave reward signals r or next observations o' empty in $\mathcal{D}_0$, as it is purely used for initialization based on critic Q value estimates. The Q-function is initialized by minimizing the following pre-training loss:

$$\mathcal{L}_0(\theta) = \mathbb{E}_{(o,a)\sim\mathcal{D}_0}\left[(f - Q(o, a; \theta))^2\right]. \tag{1}$$

Since the critic feedback f approximates the expected cumulative reward, supervised learning can be used to warm-start the Q-network. This initialization reduces the likelihood of poor exploration in early stages and helps the robot focus on meaningful behaviors.

In the phase of AS, the robot interacts with the environment, and every M time steps, we query the LLM to obtain an auxiliary critic Q-value f for the executed action. Each experience is stored in a replay buffer $\mathcal{D}$ in the form of (o, a, r, o', f). Unlike $\mathcal{D}_0$, the replay buffer $\mathcal{D}$ contains actual rewards and next observations from the environment. We augment the standard Q-learning update with the LLM-based shaping term:

$$\begin{aligned} TD &= r + \gamma \max_{a'} Q(o', a') - Q(o, a), \\ Q(o, a) &\leftarrow Q(o, a) + \alpha \cdot TD + \beta \cdot (f - Q(o, a)), \end{aligned} \tag{2}$$

where α is the learning rate and β is a heuristic shaping factor that adjusts the influence of LLM guidance. By combining standard temporal-difference updates with auxiliary LLM critic feedback, the robot can generalize to novel states, especially in environments where the reward signal is sparse or delayed. In contrast to the initial critic feedback collection, which uses randomly generated observations, the critic feedback during learning aligns with the robot's actual experiences, providing more relevant and informative guidance. The heuristic factor β is decayed throughout training to lessen the impact of LLM-derived critic feedback, allowing the robot to rely more on environmental signals.

In the case of querying LLM for critic V-value feedback, the VIAS can be implemented with slight differences. For instance, for robot learning via algorithms employing Generalized Advantage Estimation (GAE) [39]), the advantage function is calculated solely on the V-value. Consequently, we query the LLM for V-value feedback using the robot state s rather than a critic Q-value with state-action pair. In the AS phase, instead of shaping the Q-value, we use the same way to query and shape V-value.

Unlike conventional RL methods that gradually propagate sparse reward signals through trial and error, VIAS provides immediate guidance by leveraging LLM-generated critic feedback values. This direct feedback mechanism reduces the time required for reward propagation, allowing the robot to bootstrap learning from LLM-generated meaningful critic feedback signals. In early learning stage, when TD errors are unstable, LLM-generated critic feedback serves as a stabilizing reference. Moreover, for tasks where multiple actions yield indistinguishable short-term rewards but lead to divergent long-term outcomes (e.g., grab vs. put), LLMs offer informed priors that help prioritize actions leading toward the final objective.

IV. Experiments and Results Analysis

We evaluate the effectiveness of the VIAS approach in both motion planning with a robot manipulator, and task planning in simulated VirtualHome environment and on a real mobile robot manipulator.

A. Motion Planning with Robot Manipulator

We evaluated VIAS for motion control of a robotic manipulator in "Grab" and "Put" tasks on a 6-DOF robotic arm platform, which can serve as a critical skill set for task planning task such as "Setup Table" in the simulated VirtualHome environment and on the real mobile robot manipulator. We use PPO as the learning algorithm, which is an on-policy algorithm that requires training data is generated by the current policy to ensure unbiased policy gradient estimates. The VI phase of our VIAS was not implemented, since it typically employs a fully random policy for exploration, which might introduce bias and undermine the theoretical convergence guarantees of on-policy learning. In addition, we query LLM for V-value critic feedback according to state s instead of Q-value, since Generalized Advantage Estimation (GAE) was used to implicitly compute advantage functions, which can progressively refine state-value functions (V-values) without requiring explicit Q-value estimation. GPT-4o was used to provide LLM critic feedback in our implementation.

a) Experimental Setup: The Grab task requires the robot arm to start from a fixed initial pose and drive its end-effector to precisely contact a target object on the workspace, then close the gripper to lift the block up to perform the grab action. In the Put task, the robot arm starts from the pose

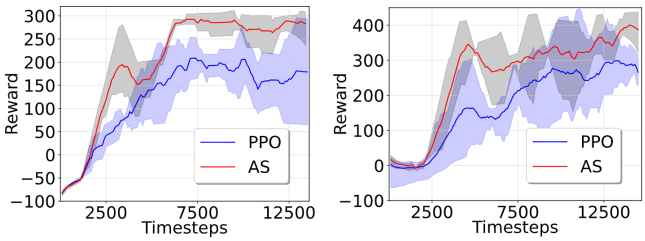

Fig. 2. Learning curves of our robot manipulator via AS in the Grab and Put tasks, in comparison to PPO.

that the robot arm holds the block off the ground, which is the pose at the end of Grab task, and transports the block to the Placement Point in the workspace, where the robot arm will open the gripper to release the block. The robot arm is equipped with a two-finger gripper and a RGB camera. During experiments: (1) The robot arm localizes spatially fixed targets within the workspace via the RGB camera; (2) Controls the end-effector reference point (gripper center point) to reach the target block and Placement Point; (3) The position of the target and Placement point are fixed throughout the task.

The state space is 23-dimentional, and we used the following variables to represent the robot's state: (1) coordinates of the target block position; (2) coordinates of the end-effector position; (3) coordinates of the Placement point position; (3) Gravity-aligned angle of the end-effector orientation. These variables are converted into natural language descriptions via predefined text templates in the prompt to query the LLM for critic feedback.

b) Results: We analyzed the learning curves of our method measured via mean accumulated episode reward over 5 trials every 100 timesteps, and compared to the vanilla PPO algorithm. Fig. 2 shows the learning curves of our method and PPO in the two tasks. Results in Fig. 2 show that our method achieved faster convergence in both tasks than the vanilla PPO agent. Specifically, at the beginning of the learning process, the performance of our method is identical to that of the PPO agent, since there is no warm-start via VI as our VIAS method. However, afterwards, our AS method achieved markedly faster learning speed, higher final performance and more stable policy due to the ability to refine value estimates over time. This indicates the clear advantages of our method for robot learning via RL in low-level motion control tasks.

B. Task Planning in Simulated VirtualHome

VirtualHome is a 3D simulated household environment consisting of interconnected rooms (e.g., kitchen, bathroom) with various interactive objects (e.g., plates, kitchen tables, and chips) to support a wide range of human-like activities. States and actions are represented as natural language descriptions, requiring agents to reason about spatial and semantic relationships to accomplish tasks. VirtualHome originally used a program-based representation for states and actions. To obtain critic feedback from the LLM, we extended the environment to provide all observations and actions in natural languages. At each timestep, the agent receives a textual description of its surroundings and the objects it can interact with. This representation challenges the agent to interpret language, reason over spatial and semantic relationships, and select viable actions to achieve a provided goal. The agent's action space comprises natural language instructions derived from templates. For instance, actions may include simple commands like *"walk to kitchen"* or *"grab plate"*, as well as more complex tasks such as *"put apple on kitchentable"*. The complexity of these actions requires the agent to identify the correct verbs, arguments, and object references from its textual observations.

We evaluate our VIAS on three tasks of increasing complexity. Table I presents descriptions of the three tasks, a sample set of actions, and the objects commonly found in each scenario. As the difficulty of the task increases, the agent encounters a wider variety of objects, potential actions, and longer descriptions, requiring more sophisticated reasoning and sequential decision-making capabilities. The three tasks were used to test the agent's ability to interpret natural language descriptions, comprehend object functions, navigate a realistic virtual setting, and execute multi-step action sequences.

a) Experimental Setup: Our VIAS approach is generally applicable to any value-based RL algorithm. In this paper, we employ Template-based Deep Q-Network (TDQN) [40] for policy learning in language-based path planning task of VirtualHome, a simple but effective architecture that captures the semantic structure of the language. A key feature of TDQN is its use of template-based action representation. Actions are defined as tuples $\langle verb, arg_0, arg_1 \rangle$, where $verb$ denotes a templated action (e.g., $\langle put\ OBJ\ on\ OBJ \rangle$), and arg_0, arg_1 specify the objects to instantiate the template. This design simplifies exploration by constraining the combinatorial action space while preserving the expressiveness of natural language commands. Three baselines are implemented and compared, including the original Template-DQN (TDQN) agent, a R Shaping agent that is a TDQN agent augemented by LLM-generated reward feedback via reward shaping [34], and a Q Shaping agent that is a TDQN agent only using LLM-generated two categories imprecise Q value (i.e. positive and negative) to update Q network and policy in the early learning stage. In addition, a VIAS agent without AS and a VIAS agent without VI were texted in ablation studies to investigate the contribution of the two phases in our method:

- **TDQN**: Naive language-based RL baseline [40].
- **R Shaping**: TDQN augmented by LLM-generated reward feedback [34].
- **Q Shaping**: TDQN augmented with LLM-generated imprecise two categories Q value in the early learning stage [36].
- **VIAS w/o AS**: Value Initialization with LLM-generated critic feedback.
- **VIAS w/o VI**: Adaptive Shaping with LLM-generated critic feedback.

TABLE I

TASK DESCRIPTION, EXAMPLE ACTIONS AND OBJECTS IN THE THREE TASKS OF VIRTUALHOME.

	Task Description	**Action Examples**	**Object Examples**
Setup Table	Place target objects onto specified surfaces.	"grab plate", "put plate on kitchentable," etc	plate, wineglass, cutleryfork, kitchentable, kitchencounter, etc
Prepare Food	Retrieve items, including food and utensils, inside a container.	"grab apple", "open fridge," etc	apple, cupcake, cereal, coffeepot, fridge, cutleryfork, etc
Turn on TV	Locate and activate a TV, and prepare snacks.	"walk to livingroom", "turn on TV," etc	tv, coffeepot, chips, etc

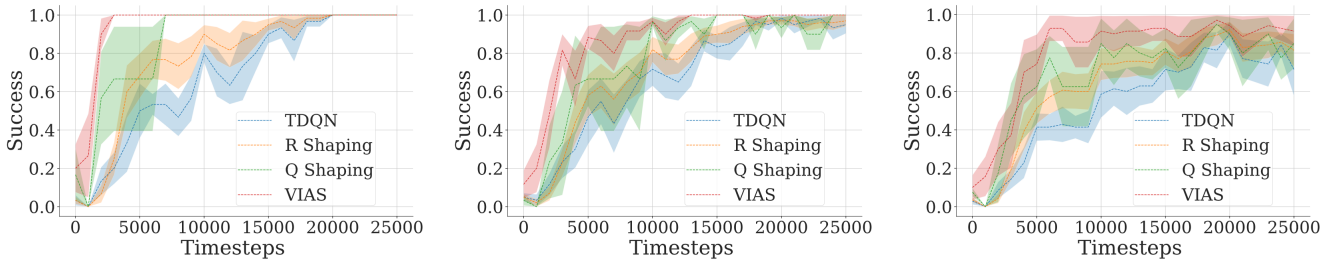

Fig. 3. Performance: The average success rate over 10 runs in three tasks of VirtualHome.

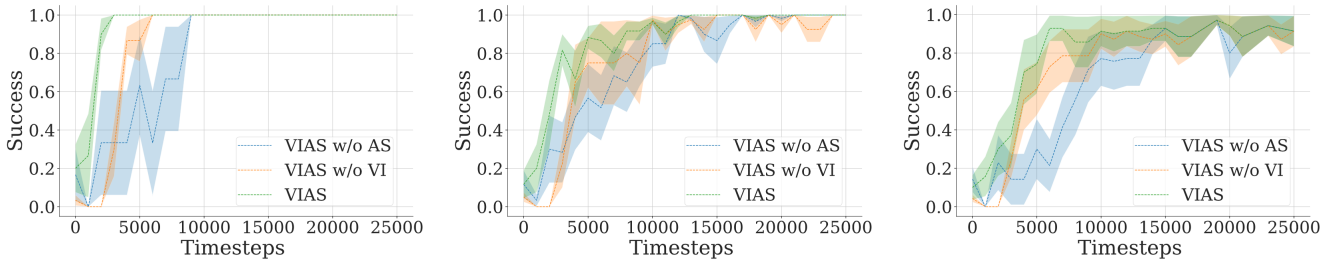

Fig. 4. Ablation Study: The performance of VIAS indicates the complemental effects of VI and AS in the three tasks of VirtualHome.

- **VIAS**: Full approach combining value initialization and adaptive shaping.

LLM feedback was provided by GPT-4o. The five agents were evaluated every $1,000$ timesteps based on the average success rate over 10 episodes. During these evaluations, the exploration rate is set to zero, ensuring the agent to take deterministic actions.

b) Simulation Results: Fig. 3 illustrates the performance of VIAS compared to the three baselines. From Fig. 3 we can see that, VIAS achieved higher success rates and faster convergence across all tasks. Specifically, in the initial stage, VIAS provided a significant "warm-start" advantage over the three baselines, which indicates the VI phase of our VIAS method can improve the agent's initial performance. In addition, throughout the learning process, VIAS learned much faster and obtained a more stable policy than the three baselines, demonstrating the advantage of using LLM-generated critic feedback to refine value functions during the whole learning process. Moreover, our results in Fig. 4 reveal the complemental effects of VI and AS in our proposed VIAS method.

c) Robustness Across LLMs and Boundary of LLM: To assess the robustness of VIAS across LLMs, we also conducted a comprehensive analysis of its performance using three distinct LLMs in the three tasks of VirtualHome. Specifically, we evaluated three versions of the VIAS agent learning from critic feedback provided by the following LLMs: GPT-4o, Gemini-1.5 [41], and Claude-3.5 [42].

The results of this analysis are summarized in Table II, which presents the success rates of the our VIAS learning from critic feedback generated from the three LLMs in the three tasks of VirtualHome. From Table II we can see that, all three versions of VIAS achieved a similar performance in the three tasks, with VIAS learning from Gemini-1.5 performing consistently better than the one from GPT-4o, and VIAS learning from Claude-1.5 performing a bit worse in 'Prepare Food' but a bit better in 'Turn on TV' than GPT-4o. However, it is worth noting that none of the three models caused a significant degradation in the overall performance of VIAS. This indicates that while the choice of LLM can influence the degree of performance enhancement, VIAS remains effective regardless of the specific LLM employed.

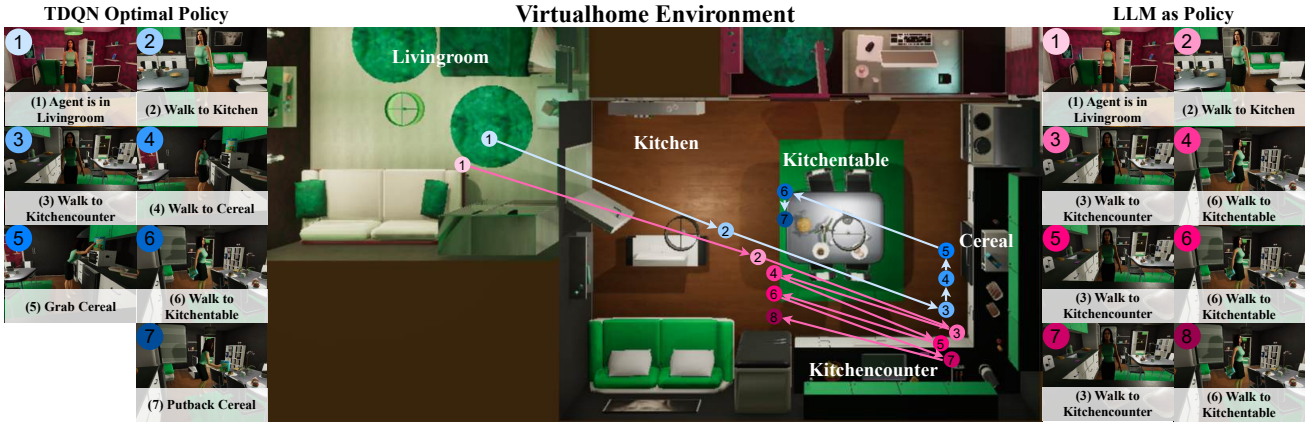

Fig. 5. A comparison of agent trajectories under LLM as Policy versus TDQN-Optimized Policy learned via our VIAS method.

TABLE II
SUCCESS RATES OF THREE IMPLEMENTATIONS OF VIAS USING DIFFERENT OFF-THE-SHELF LLMS

	GPT-4o	**Gemini-1.5**	**Claude-3.5**
Setup Table	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00
Prepare Food	0.99 ± 0.01	$\mathbf{1.00 \pm 0.00}$	0.95 ± 0.06
Turn on TV	0.91 ± 0.09	$\mathbf{0.96 \pm 0.06}$	0.94 ± 0.07

To systematically evaluate the performance boundary of LLM as policy and demonstrate the significance of our VIAS method, we also compare the performance of the agent policy learned via VIAS to the policy directly generated by an LLM in the VirtualHome environment. Using GPT-4o as the policy generator, we adapted the prompt of our VIAS framework by replacing value evaluation outputs with direct action selection, thereby constructing a purely LLM-driven decision-making system. We tested the two policies in the same three tasks with progressive difficulty levels (Setup Table, Prepare Food, and Turn on TV), for 100 independent episodes per task. The primary evaluation metrics include task success rate and average steps per episode.

Our results show that in low-complexity tasks (e.g., 'Table Setup'), the LLM demonstrated near-perfect environmental comprehension, and the generated policy achieved a 100% success rate with an average step count (6 ± 1 steps) approaching the theoretical optimum. This confirms its exceptional performance in semantically unambiguous environments. In contrast, in moderate-to-high complexity tasks (such as 'Prepare Food'), the performance of LLM-gereated policy degraded significantly, with most failures attributed to maximum step limit violations. Action sequence analysis revealed the two main following failure modes: semantic confusion—persistent misidentification of objects with similar names (e.g., walk to kitchentable vs. walk to kitchencounter), and behavioral oscillation—cyclic execution of analogous actions (e.g., $[Walk] \rightarrow kitchentable \rightarrow kitchencounter \rightarrow kitchentable$...). Fig. 5 demonstrates the trajectory of an optimal policy trained by TDQN with our VIAS and the trajectory of LLM as policy which failed because of behavioral oscillation. These results indicate that while LLMs exhibits potential to replace traditional reinforcement learning in simple tasks with discrete state spaces and high semantic distinguishability, they might suffer from lack of temporal memory mechanisms, sensitivity to semantic ambiguities and inability to handle long-range dependencies in complex tasks.

C. Task Planning with Real Mobile Robot Manipulator

To validate the proposed approach in a real-world scenario, we conducted a real-robot experiment using a mobile manipulator tasked with setting up a table according to a specified goal. The mobile manipulator used in this demonstration consists of a Segway base for navigation, a UR5e robotic arm equipped with a Hand-E gripper mounted on the Segway base for manipulation, and an overhead RGB-D camera fixed relative to the robot for perception. This setup provides the robot with the capabilities to perceive its environment, navigate within it, and interact with objects effectively.

The robot was tasked with completing the following goal: "`Goal: put plate on kitchentable and then put cup on kitchentable`". We assume the robot to possess a predefined set of fundamental skills: `grab`, `put`, and `move to`. The `grab` and `put` actions were implemented based on pose estimation using QR markers attached to the target objects, while the `move to` action was realized via a standard navigation stack, which includes standard components such as `move-base`, a global planner, a local planner, and costmap-based obstacle avoidance. The policy deployed during the experiments was pre-trained in a simulated environment. The action `walk` in the observation templates was manually replaced with `move to` to ensure consistency with the available skill set. During real-world deployment, we provide the robot with a textual description of the corresponding situation every timestep.

a) Experimental Setup: The evaluation was designed to assess three key aspects: task success rate, language-action

TABLE III

REAL ROBOT EXPERIMENT RESULTS ACROSS DIFFERENT TRAINING PHASES

Phase	Task Success Rate	Language-Action Consistency	Robustness to Perturbations
Beginning Phase (Timestep 0)	70.00 %	0 / 10 trials	3 / 6 cases
Medium Phase (Timestep 15000)	90.00 %	7 / 10 trials	4 / 5 cases
Final Phase (Timestep 25000)	100.00 %	10 / 10 trials	3 / 3 cases

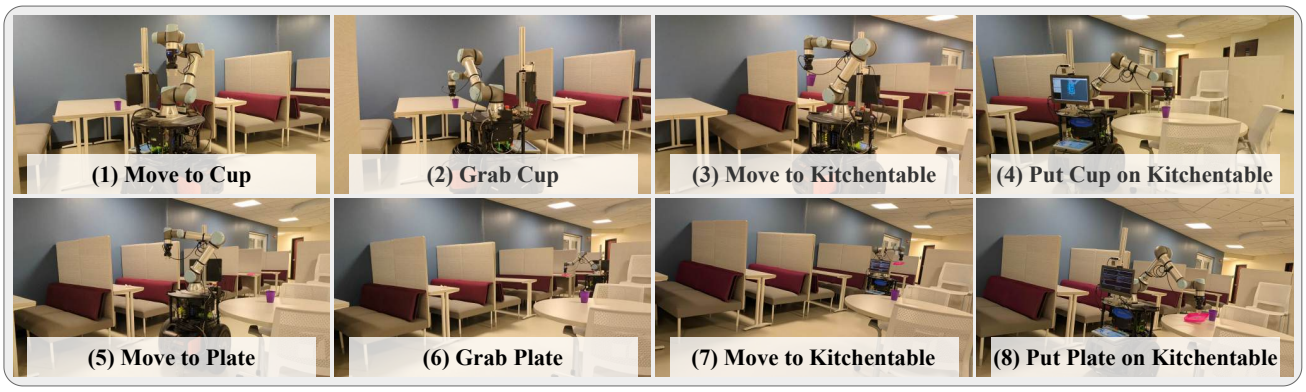

Fig. 6. An action sequence example of the real-robot experiment showcasing a learned policy via our VIAS transferred from a simulator environment. The robot is tasked with setting up a table by placing a plate and a cup onto a kitchentable.

consistency, and robustness to environmental perturbations. To evaluate the **task success rate**, the robot was instructed to complete the same goal over 10 trials with slightly varied initial positions. A trial was considered successful if the robot completed both subtasks, placing the plate and cup, without external intervention. **Language-action consistency** was assessed by examining the correspondence between the semantic content of the textual observations and the sequence of actions executed by the robot. For each trial, we manually inspected whether the robot correctly interpreted observation descriptions and selected appropriate `move to`, `grab`, and `put` actions without redundant behaviors (e.g., taking the same actions or navigating to an incorrect location). To evaluate **robustness to perturbation**, we inspect action failures in motion-level control. We then observe whether the robot could autonomously detect and deal with these disturbances.

b) Results: The experimental results are summarized in Table III and Fig. 6. Fig. 6 shows that the real-robot with a learned policy via our VIAS transferred from a simulator environment can successfully set up a table by placing a plate and a cup onto a kitchentable. As shown in Table III, in the final phase of the training, the robot achieved a 100% task success rate, completing the goal in all 10 trials. Language-action consistency was maintained in 10 out of 10 trials, with two instances of minor misinterpretation where the robot initially attempted to grasp an incorrect object but corrected itself upon failure. Regarding the robustness, there were 6, 5, and 3 action failures during the beginning, medium, and final phases, respectively. We observed that action failures occasionally occurred due to planner instability and an inconsistent end-effector pose. Even when the previous action fails, the policy uses the semantic description of the observation to infer the failure and correct its behavior by re-selecting the proper action. Overall, these results validate that the proposed approach can transfer a policy trained in simulation to a real-world robot while maintaining high task performance, correct semantic grounding, and robustness to unexpected events.

V. CONCLUSION & FUTURE WORK

In this paper, we propose the VIAS framework that integrates LLM-provided critic feedback for robot learning via RL. This critic feedback in the form of V-values (state-values) or Q-values (state-action-values), can be used to warm-up robot policy learning and adaptively shape value updates online during the learning process. Our results in a VirtualHome environment and on two real-world robot platforms in both motion planning and task planning tasks show that VIAS can outperform state-of-the-art methods in both learning speed and task completion. Moreover, further analysis indicates that VIAS remains effective regardless of the specific LLM model employed and outperforms the LLM directly generated policy, demonstrating its potential for real-world robot applications.

Potential directions for future research include developing methods to dynamically adjust the heuristic factor β based on the agent's confidence or the quality of the LLM's feedback, integrating VIAS with reward shaping methods, and leveraging pre-trained vision-language models for visual perception and end-to-end behavior generation.

REFERENCES

[1] J. Gao, W. Ye, J. Guo, and Z. Li, "Deep reinforcement learning for indoor mobile robot path planning," *Sensors*, vol. 20, no. 19, p. 5493, 2020.

[2] S. W. Abeyruwan, L. Graesser, D. B. D'Ambrosio, A. Singh, A. Shankar, A. Bewley, D. Jain, K. M. Choromanski, and P. R. Sanketi, "i-sim2real: Reinforcement learning of robotic policies in tight human-robot interaction loops," in *Proceedings of Annual Conference on Robot Learning*. PMLR, 2023, pp. 212–224.

[3] J. Hu, R. Hendrix, A. Farhadi, A. Kembhavi, R. Martin-Martin, P. Stone, K.-H. Zeng, and K. Ehsan, "Flare: Achieving masterful and adaptive robot policies with large-scale reinforcement learning fine-tuning," *arXiv preprint arXiv:2409.16578*, 2024.

[4] D. Yarats, A. Zhang, I. Kostrikov, B. Amos, J. Pineau, and R. Fergus, "Improving sample efficiency in model-free reinforcement learning from images," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 12, 2021, pp. 10 674–10 681.

[5] J. Huang, J. Hao, R. Juan, R. Gomez, K. Nakamura, and G. Li, "Gan-based interactive reinforcement learning from demonstration and human evaluative feedback," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 4991–4998.

[6] L. Zhang, C. Zheng, H. Wang, R. Gomez, E. Nichols, and G. Li, "Autonomous storytelling for social robot with human-centered reinforcement learning," in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2024, pp. 2450–2456.

[7] F. Zhang, J. Li, Y.-C. Li, Z. Zhang, Y. Yu, and D. Ye, "Improving sample efficiency of reinforcement learning with background knowledge from large language models," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 36, no. 11, pp. 19 681–19 692, 2025.

[8] A. Andres, L. Schäfer, S. V. Albrecht, and J. Del Ser, "Using offline data to speed up reinforcement learning in procedurally generated environments," *Neurocomputing*, vol. 618, p. 129079, 2025.

[9] K. Narasimhan, T. Kulkarni, and R. Barzilay, "Language understanding for text-based games using deep reinforcement learning," *arXiv preprint arXiv:1506.08941*, 2015.

[10] M.-A. Côté, A. Kádár, X. Yuan, B. Kybartas, T. Barnes, E. Fine, J. Moore, M. Hausknecht, L. El Asri, M. Adada *et al.*, "Textworld: A learning environment for text-based games," in *Computer Games: 7th Workshop, CGW 2018, Held in Conjunction with the 27th International Conference on Artificial Intelligence, IJCAI 2018, Stockholm, Sweden, July 13, 2018, Revised Selected Papers 7*. Springer, 2019, pp. 41–75.

[11] Y. J. Ma, W. Liang, G. Wang, D.-A. Huang, O. Bastani, D. Jayaraman, Y. Zhu, L. Fan, and A. Anandkumar, "Eureka: Human-level reward design via coding large language models," *arXiv preprint arXiv:2310.12931*, 2023.

[12] T. Zhang, F. Ladhak, E. Durmus, P. Liang, K. McKeown, and T. B. Hashimoto, "Benchmarking large language models for news summarization," *Transactions of the Association for Computational Linguistics*, vol. 12, pp. 39–57, 2024.

[13] P. Vaithilingam, T. Zhang, and E. L. Glassman, "Expectation vs. experience: Evaluating the usability of code generation tools powered by large language models," in *CHI conference on Human Factors in Computing Systems extended abstracts*, 2022, pp. 1–7.

[14] 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. 24 824–24 837, 2022.

[15] S. Yao, D. Yu, J. Zhao, I. Shafran, T. Griffiths, Y. Cao, and K. Narasimhan, "Tree of thoughts: Deliberate problem solving with large language models," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[16] A. Hussein, M. M. Gaber, E. Elyan, and C. Jayne, "Imitation learning: A survey of learning methods," *ACM Computing Survey*, vol. 50, no. 2, Apr. 2017. [Online]. Available: https://doi.org/10.1145/3054912

[17] B. Hertel, T. Nguyen, M. E. Cabrera, and R. Azadeh, "Automatic estimation and evaluation of multi-objective human preferences for learning from demonstration," *Robot Learning*, vol. 3, no. 1, pp. 1–23, 2026.

[18] F. Torabi, G. Warnell, and P. Stone, "Behavioral cloning from observation," *arXiv preprint arXiv:1805.01954*, 2018.

[19] A. Y. Ng, D. Harada, and S. Russell, "Policy invariance under reward transformations: Theory and application to reward shaping," in *Proceedings of International Conference on Machine Learning (ICML)*, vol. 99, 1999, pp. 278–287.

[20] E. Wiewiora, G. W. Cottrell, and C. Elkan, "Principled methods for advising reinforcement learning agents," in *Proceedings of the 20th International Conference on Machine Learning (ICML)*, 2003, pp. 792–799.

[21] W. B. Knox and P. Stone, "Interactively shaping agents via human reinforcement: The tamer framework," in *Proceedings of the fifth International Conference on Knowledge Capture*, 2009, pp. 9–16.

[22] Y. Hayamizu, S. Amiri, K. Chandan, K. Takadama, and S. Zhang, "Guiding robot exploration in reinforcement learning via automated planning," in *Proceedings of the International Conference on Automated Planning and Scheduling*, vol. 31, 2021, pp. 625–633.

[23] Y. Cao, H. Zhao, Y. Cheng, T. Shu, Y. Chen, G. Liu, G. Liang, J. Zhao, J. Yan, and Y. Li, "Survey on large language model-enhanced reinforcement learning: Concept, taxonomy, and methods," *arXiv preprint arXiv:2404.00282*, 2024.

[24] J. Pei, X. Zheng, Y. Liu, B. Zhu, D. Wang, J. An, R. Voyles, and X. Ma, "The evolution of end-to-end vision-language-action (vla) architectures in robotics," *Robot Learning*, vol. 3, no. 2, pp. 1–36, 2026.

[25] 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.

[26] G. Wang, Y. Xie, Y. Jiang, A. Mandlekar, C. Xiao, Y. Zhu, L. Fan, and A. Anandkumar, "Voyager: An open-ended embodied agent with large language models," *arXiv preprint arXiv:2305.16291*, 2023.

[27] S. Yao, J. Zhao, D. Yu, N. Du, I. Shafran, K. Narasimhan, and Y. Cao, "React: Synergizing reasoning and acting in language models," *arXiv preprint arXiv:2210.03629*, 2022.

[28] N. Shinn, F. Cassano, A. Gopinath, K. Narasimhan, and S. Yao, "Reflexion: Language agents with verbal reinforcement learning," *Advances in Neural Information Processing Systems*, vol. 36, pp. 8634–8652, 2023.

[29] C. Chen, Y.-F. Wu, J. Yoon, and S. Ahn, "li2022pretrainedtransdreamer: Reinforcement learning with transformer world models," *arXiv preprint arXiv:2202.09481*, 2022.

[30] V. Micheli, E. Alonso, and F. Fleuret, "Transformers are sample-efficient world models," *arXiv preprint arXiv:2209.00588*, 2022.

[31] 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.

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

[33] Y. Du, O. Watkins, Z. Wang, C. Colas, T. Darrell, P. Abbeel, A. Gupta, and J. Andreas, "Guiding pretraining in reinforcement learning with large language models," in *International Conference on Machine Learning*. PMLR, 2023, pp. 8657–8677.

[34] K. Chu, X. Zhao, C. Weber, M. Li, and S. Wermter, "Accelerating reinforcement learning of robotic manipulations via feedback from large language models," *arXiv preprint arXiv:2311.02379*, 2023.

[35] T. Carta, P.-Y. Oudeyer, O. Sigaud, and S. Lamprier, "Eager: Asking and answering questions for automatic reward shaping in language-guided rl," *Advances in Neural Information Processing Systems*, vol. 35, pp. 12 478–12 490, 2022.

[36] X. Wu, "From reward shaping to q-shaping: Achieving unbiased learning with llm-guided knowledge," *arXiv preprint arXiv:2410.01458*, 2024.

[37] A. Hurst, A. Lerer, A. P. Goucher, A. Perelman, A. Ramesh, A. Clark, A. Ostrow, A. Welihinda, A. Hayes, A. Radford *et al.*, "Gpt-4o system card," *arXiv preprint arXiv:2410.21276*, 2024.

[38] H. Hu and D. Sadigh, "Language instructed reinforcement learning for human-ai coordination," in *International Conference on Machine Learning*. PMLR, 2023, pp. 13 584–13 598.

[39] J. Schulman, P. Moritz, S. Levine, M. Jordan, and P. Abbeel, "High-dimensional continuous control using generalized advantage estimation," *arXiv preprint arXiv:1506.02438*, 2015.

[40] M. Hausknecht, P. Ammanabrolu, M.-A. Côté, and X. Yuan, "Interactive fiction games: A colossal adventure," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 05, 2020, pp. 7903–7910.

[41] G. Team, P. Georgiev, V. I. Lei, R. Burnell, L. Bai, A. Gulati, G. Tanzer, D. Vincent, Z. Pan, S. Wang *et al.*, "Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context," *arXiv preprint arXiv:2403.05530*, 2024.

[42] Anthropic, "Claude 3.5 sonnet news," https://www.anthropic.com/news/claude-3-5-sonnet, 2024, accessed on 27 June 2024.