RADI: LLMs as World Models for Robotic Action Decomposition and Imagination

Anonymous Author(s)
Affiliation
Address
email

Abstract

In this paper, we introduce the **R**obotic **A**ction **D**ecomposition and **I**magination (RADI) framework, which is a novel framework that leverages LLMs for embodied task planning through three core mechanisms: hierarchical action decomposition, environment imagination, and self-reflective plan correction. Specifically, RADI first gradually decomposes a complex robot task into atomic action sequences, then imagines the execution results of each action based on the environment state, and verifies whether it meets the task expectations through the state changes. If the expectations are not met, it triggers the self-reflective mechanism to re-optimize the action decomposition. We also fine-tuned the action decomposition and imagination modules separately, both of which achieved modest performance gain than using the base model alone. The experiments are conducted based on GPT-4 in the VirtualHome environment, and the results show that RADI significantly improves the success rate of task planning, and verifies the effectiveness of LLM as a world model in robotics.

1 Introduction

Robotics is of irreplaceable importance in promoting social progress, enhancing productivity, and improving human life [Brooks, 1986]. For example, the application of robots in the manufacturing industry has greatly improved productivity and product quality [Gatla et al., 2007a,b]. Robot task planning is a crucial step to ensure that robots can complete complex tasks efficiently and accurately, and it enables robots to maximize their performance in various application scenarios by analyzing, decomposing, and optimizing paths for tasks [Hanheide et al., 2017, Paxton et al., 2019, Galindo et al., 2008, Zhang et al., 2017]. In the field of robot task planning, world models [Ha and Schmidhuber, 2018] play a pivotal role. It is crucial to predict the outcome of actions, reduce the cost of physical trial and error, and improve the safety of decisions. Traditional world models based on physical simulation or rule engines have significant limitations since they rely heavily on accurate environment modeling [Blumenthal et al., 2013, Roth et al., 2003, Zhang and Faugeras, 1990], a process that is costly and difficult to generalize to complex or dynamic scenarios.

Large Language Models (LLMs) [Zhao et al., 2023], represented by the GPT family of models [Floridi and Chiriatti, 2020, Lund and Wang, 2023, Achiam et al., 2023] developed by OpenAI, have achieved performance far beyond that of previous models on a wide range of tasks in natural language processing [Baktash and Dawodi, 2023], and to some extent have shown the potential for artificial general intelligence (AGI) by going beyond the language model itself and understanding the physical world [Bubeck et al., 2023], and even more recently there has been some research to show that LLMs can be effective in generating robot task plans. For example, the PROGPROMPT [Singh et al., 2023] achieves a high success rate in the VirtualHome housework task using LLMs with program-like prompts. In addition, the RoboMatrix [Mao et al., 2024] framework provides a skill-centered hierarchical approach for scalable robot task planning and execution in the open world, demonstrating generalization performance across new objects, scenarios, tasks, and robots.

However, two key research gaps remain unexplored. First, it remains unclear whether LLMs can simulate environment dynamics and anticipate the outcome of actions beyond linguistic reasoning [Yao et al., 2023b, Wu et al., 2023, Chalvatzaki et al., 2023, Wu et al., 2024]. Second, previous work lacks mechanisms for verifying and correcting task plans when failures occur during imagined execution. So, addressing these gaps is crucial for deploying LLM-based planner in real-world dynamic environments.

In this paper, we propose the **R**obotic **A**ction **D**ecomposition and **I**magination (RADI) framework. Specifically, we first utilize LLM to achieve action decomposition by planning and progressively decomposing a complex task into a series of atomic actions. Subsequently, the LLM is employed to simulate environment dynamics, a process we refer to as imagination. Based on the current state of the environment, the LLM is asked to predict state transitions resulting from the execution of each atomic action, and to anticipate whether the resulting states satisfy the task objective. If the predicted outcome indicates failure, the LLM would re-perform the action decomposition, thus improving the success rate of task planning through the reflection of the LLM itself. The experiment conducted in VirturalHome [Puig et al., 2018] shows an improvement in task planning success rate that can be used as a measure of the LLM's ability to act as a world model. The contribution of this paper can be summarized as follows:

- We propose the RADI framework that consists of action decomposition and environmental imagination, allowing the LLMs to break down complex robot tasks and predict the outcomes of actions based on the current environmental state, as well as achieve environmental imagination-driven error correction for action decomposition.
- We provide a systematic way to explore the potential of LLMs as world models in the field of robot task planning, paving the way for more interpretable and reliable applications of LLMs in robotic systems in a variety of environments.
- We conduct experiments on four public datasets in VirtualHome using GPT-4, one of the state-of-the-art LLMs. Experimental results show the effectiveness of the RADI framework to improve robot task planning, and LLMs can serve as the world model for robotics.

2 Related Work

Traditional robotic task planning primarily relies on symbolic and rule-based systems such as Planning Domain Definition Language (PDDL) [Jiang et al., 2019], behavior trees [Iovino et al., 2022], and model predictive control [Lee, 2011, Jing et al., 2023], which require precise modeling of environments and task domains [Shin and Jung, 2024]. These approaches enable structured reasoning and deterministic planning but struggle in dynamic or unstructured environments due to their reliance on handcrafted rules and limited adaptability [Jang et al., 2024].

The emergence of LLMs has introduced new paradigms in robotic planning. Recent studies have explored in-context learning [Dong et al., 2022, Yao et al., 2023a], programmatic prompting [Wu et al., 2024], and reflection-based self-correction [**?**] as techniques to enable LLMs to generate feasible and adaptive plans in dynamic environments. LLMs enable zero-shot task planning and natural language interaction, but their grounding in real-world environments with consistent state perception remains limited [Jang et al., 2024]. To enhance execution fidelity, methods such as DGAP [Qian et al., 2025] use step-wise discriminators to guide LLMs toward actions aligned with expert demonstrations. Similarly, ISR-LLM [Zhou et al., 2024] and Inner Monologue [Huang et al., 2022] propose closed-loop feedback using imagined or retrieved environmental states.

Beyond generating plans, LLMs have also been explored as internal world models capable of simulating environment dynamics and predicting the outcomes of actions [Ge et al., 2024]. For example, MLDT introduces a multi-level decomposition framework that enhances LLM reasoning through goal-task-action hierarchies, enabling more structured and context-aware prediction [Wu et al., 2024]. SayCan [Ahn et al., 2022] uses a pretrained value function to link LLM outputs to the environment. ISR-LLM [Zhou et al., 2024] iteratively refines generated plans via LLMs. However, most of these methods either require significant external supervision or fail to internalize world dynamics within the LLM itself.

RADI leverages the dual capabilities of LLMs in robotic task planning and internal world modeling. Unlike MLDT [Wu et al., 2024], which focuses on hierarchical decomposition to simplify long-horizon planning, RADI explicitly evaluates whether the imagined consequences of each action align with expected environment states. Unlike DGAP Qian et al. [2025], which optimizes plan generation using step-wise scores from a learned discriminator, RADI does not rely on external supervision but instead uses internal imagination to guide plan correction. Unlike approaches such as ReAct [Yao et al., 2023b], Inner Monologue [Huang et al., 2022], and E2WM [Xiang et al., 2023], which depend on environment rollouts or feedback from simulators, RADI assesses the consistency of predicted intermediate states directly from the LLM's own reasoning.

3 Preliminaries

Robot task planning. Robot task planning is the process of allowing a robot, upon receiving a command for a particular task, to generate a detailed executable plan in a given environment to achieve the goal of the task [Tsarouchi et al., 2016, Hanheide et al., 2017, Paxton et al., 2019]. Specifically, given the task goal G, the observation O consisting of objects in the environment E and their relationships, and a set of all possible actions $A = \{a_1, a_2, \dots, a_n\}$ executable for the robot, a task planning algorithm $\mathcal{T}$ aims to find an action sequence π to achieve the goal of the task. In other words, $\mathcal{T} : (G, O, A) \mapsto \pi$. For example, if the goal $G =$*"put one cupcake in microwave and switch on microwave"*, the observation $O =$*"one cupcake is in fridge, one cupcake is in kitchencabinet"*, the possible action set $A = \{walk, open, ..., switchon\}$, then we aim to generate an action sequence $\pi =$ *"walk to fridge, open fridge, grab cupcake,...,switchon microwave"*.

World Models and LLM-based Simulation. Traditional task planners often rely on physics-based simulators or formal domain models to predict environment dynamics. However, such approaches require high-fidelity modeling and often struggle with generalization. Inspired by model-based reinforcement learning [Ha and Schmidhuber, 2018], we treat Large Language Models (LLMs) as abstract world models capable of simulating environment transitions. Instead of accessing the full environment state S, our system operates on partial observations $O \subset S$, assuming that the agent has only a partial, language-based view of the environment. Within this setting, the LLM learns to simulate the effects of executing actions in O, allowing planning through imagined rollouts rather than symbolic inference.

Memory-Augmented Planning. To enable knowledge accumulation and contextual reuse, we augment the planning process with a memory module $\mathcal{M}_{\text{mem}}$ that stores past experiences as triplets (O, π, R), where O is the observed world state, π is the previously executed action sequence, and R contains reasoning traces including verification results and corrections. When facing a new planning task, the system embeds the current observation O' into a semantic vector space and retrieves memory entries with high similarity scores via a function $\texttt{Retrieve}(O') \rightarrow (O_i, \pi_i, R_i)$. This allows the system to transfer knowledge from prior similar scenarios, improving generalization across structurally related but distinct tasks.

Reflection and Self-Correction. To further refine action sequences, we incorporate a reflection mechanism $\mathcal{R}$ that performs post-hoc verification and correction. Given a candidate plan π and an environment description O, the LLM simulates execution through $\mathcal{M}(O, \pi) \mapsto (\hat{O}, \texttt{feasibility})$ and analyzes mismatches between $\hat{O}$ and the goal-satisfying state O^*. When $\texttt{feasibility} = \texttt{False}$, the system identifies failure causes and generates an improved plan $\pi' = \mathcal{R}(O, \pi, \hat{O})$, closing the feedback loop. This mechanism enables the agent to learn from its own planning errors and evolve over time without parameter updates, supporting a form of metacognitive reasoning and self-improvement.

4 Methodology

We propose the RADI framework, as illustrated in Figure 1 where we first let the LLM complete the decomposition of a robot action sequence based on the task description and the observed environment state. Then, the LLM functions as a world model, simulating the change of the environment state after the execution of the action sequence to verify whether the action sequence can accomplish the corresponding task. If LLM determines that the task cannot be accomplished, the system re-invokes

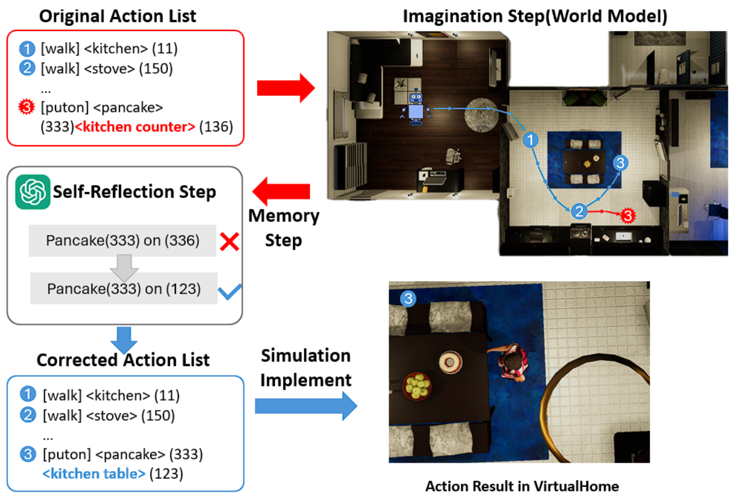

Figure 1: overview of the RADI framework for robotic task planning.

the LLM to the action decomposition. The general pipeline of the proposed framework is shown in Figure 1.

4.1 Action Decomposition

The action decomposition module is independent of the followed imagination module and can be implemented by any decomposition methods. In practice, to improve the efficiency of action decomposition, we adopt a hierarchical task decomposition framework [Wu et al., 2024] that reduces the complexity of planning difficult tasks through an incremental decomposition strategy. Firstly, a goal-oriented decomposition is adopted to decompose the overall task into a number of independent sub-goals based on semantic associations; subsequently, environment observation is introduced at the task execution layer, and each sub-goal is transformed into an actionable sequential task chain through hierarchical prompt templates; and finally, at the action generation layer, each sub-goal is parsed into specific action instructions by combining with the domain knowledge base, and the standardized action sequences are extracted by a pattern-matching algorithm. Formally, given task goal G, observation O and possible action set A, we exploit LLM to obtain the action sequence:

$$\pi = (a_1, a_2, ..., a_m) = LLM(G, O, A). \tag{1}$$

4.2 Imagination via LLM Simulation

The imagination module serves as a critical component in our framework, enabling the system to predict action outcomes through environmental state transition modeling. Rather than relying on traditional simulation-based approaches that require explicit physics models, we leverage the implicit world knowledge embedded within Large Language Models to perform environment state prediction. The structural design of this module focuses on iterative LLM-based state-transition verification mechanisms that maintain consistency between predicted states and environmental constraints.

During the imagination process, the module accepts two primary inputs: (1) the complete action sequence generated by the action decomposition module, and (2) a structured textual representation of the current environmental state, including object relationships, spatial configurations, and physical properties. These inputs are combined into a carefully crafted prompt that instructs the LLM to simulate the execution of the action sequence within the described environment.

In response to this prompt, the LLM produces both stepwise state updates and a binary feasibility flag in one holistic invocation, enabling long-range dependency reasoning and reduced LLM calls.

To enhance the robustness of the imagination-based verification, we implement a K-round iterative correction mechanism. When the LLM determines that an action sequence is infeasible due to physical constraints, logical inconsistencies, or precondition violations, the system automatically

Algorithm 1 Prompt Construction

Require: $world\_model\_str$, $action\_plan\_str$
Ensure: $prompt$
1: $yaml \leftarrow$ `FormatAsYAML`$(world\_model\_str)$
2: $schema \leftarrow$ `LoadActionSchema`$()$
3: $actions \leftarrow$ `LineByLine`$(action\_plan\_str)$
4: $prompt \leftarrow$ [`"WORLD MODEL:"`, $yaml$, `"ACTION SCHEMA:"`, $schema$, `"TASK:"`, $actions$]
5: **return** $prompt$

triggers a Self-Correction Prompt. This prompt encapsulates the identified conflict information and instructs the LLM to generate a revised action sequence that resolves the detected issues. The process continues iteratively until either a viable action sequence is produced or the predefined iteration limit K is reached. In cases where the system fails to converge on a feasible solution after K iterations, we implement an abstention mechanism rather than executing an unreliable action sequence, prioritizing safety and reliability over task completion.

4.3 Memory and Reflection

The memory and reflection components constitute the experiential learning core of our framework, enabling continuous improvement through systematic knowledge accumulation and error analysis. These modules work in concert to transform individual interactions into generalizable knowledge, closely mimicking human cognitive processes of learning from experience.

Memory Module. The memory module maintains a structured repository of previous planning experiences to support decision-making in novel scenarios. Unlike traditional knowledge bases that rely on rule-based representations, our approach encodes experiences in a rich, semantic format that preserves both environmental contexts and reasoning processes. Each memory entry consists of three fundamental components: (1) a structured representation of the world state at the decision points, (2) action verification logs detailing reasoning steps and outcomes, and (3) corrected action plans that represent refined solutions derived from reflection on previous failures.

The architectural organization of the memory module follows a hierarchical structure optimized for efficient retrieval based on situational similarity. At the core of this structure is a vectorized representation of world states that enables semantic similarity matching between current scenarios and past experiences. This embedding-based approach allows the system to retrieve relevant experiences even when scenarios are not identical but share important structural similarities, facilitating transfer learning across related but distinct environmental configurations.

The memory module operates through three distinct phases that form a continuous improvement cycle: initialization, retrieval, and augmentation. During initialization, the module is seeded with a small set of carefully crafted experiences that capture fundamental action-consequence relationships. In the retrieval phase, when faced with a new scenario, the system embeds the current environmental state into a high-dimensional vector representation for similarity-based retrieval. Finally, through augmentation, the system continuously accumulates new experiences as it interacts with the environment, systematically encoding and integrating them into the memory repository.

Reflection Module. The reflection module implements a systematic approach to error analysis and correction, enabling the system to learn from failures rather than simply recording them. This module enables the system to critique its reasoning ability and improve action plans.

When examining action sequences, the reflection module performs a multi-stage analysis consisting of (1) action validation, (2) error categorization, (3) causal analysis, and (4) plan refinement. During action validation, each action is verified against a formal schema that defines preconditions and postconditions. Error categorization classifies detected issues into specific categories such as object existence errors or state inconsistencies. Causal analysis traces errors to specific assumptions, inference patterns, or knowledge gaps. Finally, plan refinement generates corrected action plans that resolve identified issues while preserving the original task objectives.

Module	Inputs	Outputs
Imagination	`scene_desc`	`plan_str`
Verification	unified LLM prompt	(`LLM_out`, *feasibility*)
Memory	triplet (`scene_desc`, updates, plan)	persisted JSON file (`memory.json`)
Reflection	(`LLM_out`, `scene_desc`)	(corrected plan, error diagnosis)

Table 1: Module I/O specifications aligned with our implementation. Here, `LLM_out` is the raw LLM response string; *feasibility* is the binary flag; `scene_desc` is the structured textual scene descriptor.

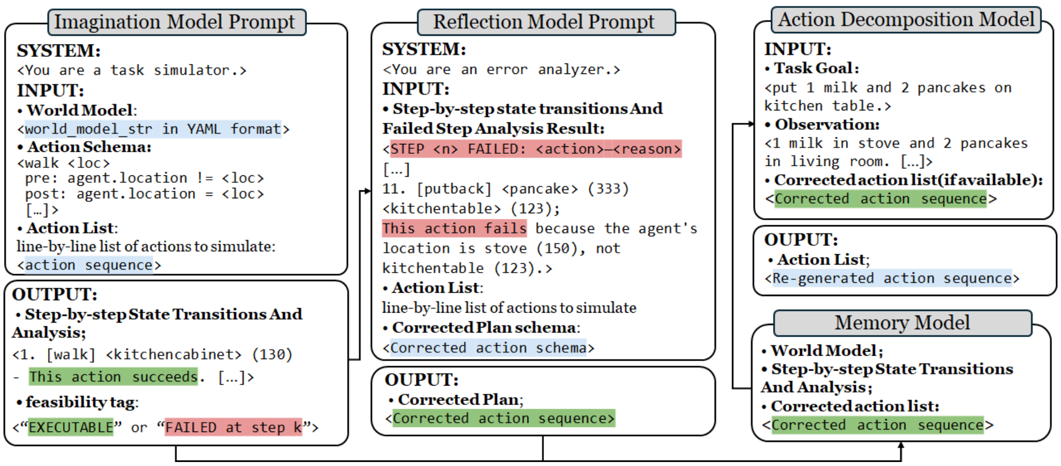

Figure 2: An example with prompt and result of Imagination and Reflection in our RADI framework.

The reflection process is implemented through a specialized verification prompt provided to the LLM, which includes the current world state, action schema definitions, and task requirements. The LLM simulates the execution of each action in sequence, identifying failure points and providing detailed reasoning about constraint violations. When failures are detected, the reflection module leverages the LLM's reasoning capabilities to generate corrected plans that are subsequently stored in the memory module.

Through this cyclical process of imagination, reflection, and memory updating, our framework achieves a form of experiential learning that parallels human cognitive development. The system's planning capabilities naturally improve as it encounters diverse scenarios and refines its understanding of action-consequence relationships, without requiring explicit retraining or parameter updates. This approach enables robust generalization to novel situations by transferring knowledge from similar past experiences, a hallmark of human-like adaptability in complex environments.

4.4 Instruction Tuning

Tune the Action Decomposition Module. To enhance action decomposition performance, we fine-tune our model using the goal-sensitive dataset proposed by MLDT [Wu et al., 2024]. The dataset comprises two types of samples: **task-level** and **action-level** decompositions, both constructed via ChatGPT-based multistep planning and validated in the VirtualHome environment to ensure executability.

Each sample is formatted as an `instruction-output` pair:

- **Task-level format:** The instruction includes a natural language task description incorporating a goal state and object locations. The output is a semantically structured sequence of sub-goals.
  `Instruction:` "from action import grab <obj> in <obj>, put <obj> in <obj>, put <obj> on <obj>, switch on <obj> # key object location: … # task goal: … def task():"
  `Output:` "# the goal means the task is … # Sub-goals "

- **Action-level format:** The instruction describes a single sub-goal. The output contains an action sequence to finish the goal.
  `Instruction:` "from actions import walk <obj>, grab <obj>, switchon <obj>, open <obj>, close <obj>, putin <obj> <obj>, putback <obj> <obj> # task: ..."
  `Output:` Action Plan

This instruction-tuning strategy enables the model to generate goal-sensitive and logically consistent action plans, improving its reasoning and planning capabilities within embodied environments.

Tune the Imagination Module. To fine-tune the imagination module, we collect high-quality samples from the RADI pipeline, where the full action sequence is both executable and successful in the VirtualHome simulator. Each training instance is framed as an `instruction-input-output` triplet:

- **Instruction:**
  `WORLD MODEL`: Object observations and inter-object relationships.
  `AVAILABLE ACTION PRIMITIVES`: A predefined list of allowable atomic actions in the simulation.
  `ACTION SCHEMA`: Descriptions of action syntax and execution rules.
  `TASK`: An instruction to simulate each action step in sequence.
- **Input:**{action_plan_str}
- **Output:**"STEP <n> FAILED: <action> — <reason>" or "EXECUTABLE"

This instruction-tuning setup allows the imagination module to more accurately assess the feasibility of complex, goal-conditioned plans in novel environments, improving downstream task reliability and generalization.

5 Experiments

5.1 Experimental Settings

Environment. We conducted all experiments in VirtualHome [Puig et al., 2018], a 3D household simulation environment comprising diverse indoor scenes (e.g., kitchens, living rooms, bedrooms) and a wide range of interactive objects with configurable physical states. VirtualHome provides a realistic yet controllable testbed for evaluating high-level task planning and action reasoning in embodied settings. This environment supports structured evaluations by enabling textual representations of scene graphs, object functionalities, and action consequences. Our framework interacts with VirtualHome through iterative action decomposition and environment simulation, allowing us to evaluate the ability of language models to generate valid, goal-consistent plans under realistic constraints.

Fine-tuning Datasets. To fine-tune the Action Decomposition module, we used the goal-sensitive corpus proposed by MLDT [Wu et al., 2024], which comprises 16,293 samples, including 2,202 task-level and 14,091 action-level decomposition instances. For tuning the Imagination module, we collected 400 high-quality samples in which the module produced both executable and successful predictions. This core set was further expanded via zero-shot prompting with ChatGPT, resulting in a final dataset of 2,000 imagination instances.

Testing Datasets. Our comprehensive evaluation framework encompasses the entire spectrum of VirtualHome datasets established through the Language-Instructed Decision Transformer (LID) methodology [Li et al., 2022], comprising InDistribution, NovelScenes, and NovelTasks, supplemented by the computationally intensive LongTasks [Wu et al., 2024] corpus. Through systematic application of both Success Rate and Executability Rate metrics across these datasets, we establish a rigorous quantification of LLM capabilities in task decomposition, environmental state prediction, and action validation when guided solely through sophisticated prompt engineering techniques, deliberately eschewing in-context exemplars that might artificially scaffold performance.

Large Language Models. We adopt GPT-4 as the primary large language model for this study. Unlike methods that rely on extensive fine-tuning, our framework remains purely prompt-based,

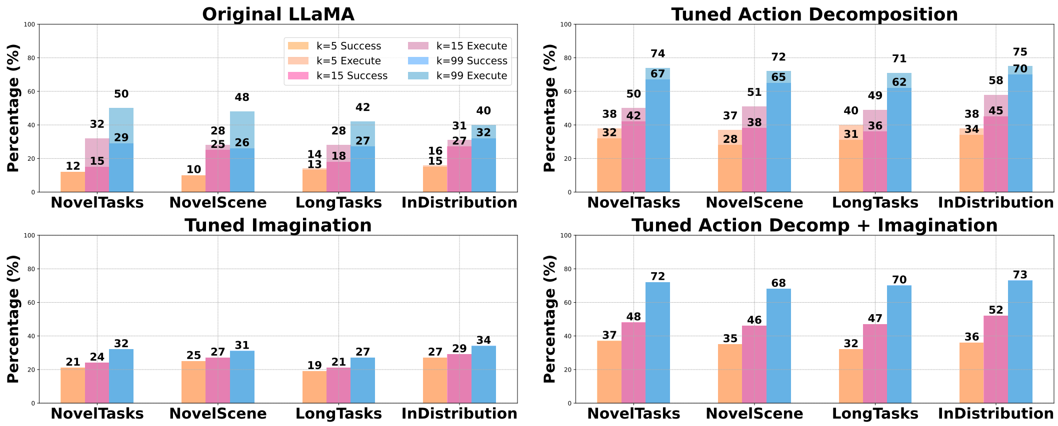

Figure 3: Fine-tuned results

leveraging GPT-4's capacity for iterative task decomposition and environment "imagination." Specifically, GPT-4 breaks down complex robotic tasks into atomic actions and predicts the resulting state transitions without any parameter updates. This design rigorously tests GPT-4's ability to infer physical dynamics and assess the feasibility of each action in a simulated environment.

Evaluation Metrics. We employ three complementary metrics as our exclusive means of evaluation: Success Rate After Abstention (SRA), Abstention Rate (AR), and Overall Success Rate (OSR). SRA is defined as the percentage of tasks successfully executed among those the system does not abstain from, AR measures the proportion of tasks the system opts to skip, and OSR indicates the fraction of all tasks that are ultimately completed. In our framework, a plan is deemed successful only if it satisfies two critical criteria: (1) all actions can be executed in a logically consistent manner, and (2) the resultant state transitions precisely align with the intended outcomes.

5.2 Baselines

We compare our proposed RADI framework with several existing baselines such as Embodied Planning, ReAct, and MLDT, each representing a different perspective on leveraging LLMs for robotic reasoning and execution.

Embodied Planning (Embodied) [Wu et al., 2023] integrates the physical embodiment of robots into the task planning process, often relying on perception-action feedback loops and contextual memory. It treats LLMs as modules to interpret the environment and generate plausible actions based on grounded knowledge. **ReAct** [Yao et al., 2023b] combines reasoning and acting in LLMs, allowing them to think step-by-step and act iteratively in interactive environments. By interleaving natural language reasoning and action generation, ReAct can generate more flexible plans for unseen tasks. **Multi-Level Decomposition Task planning** (MLDT) [Wu et al., 2024] applies hierarchical task decomposition and instruction tuning, but is primarily optimized for open-source LLMs.

5.3 Experimental Results

We evaluated the proposed RADI framework on four benchmark splits of the VirtualHome environment. The quantitative results are presented in Tables 2, 3, and 4.

Effect of Reflection Depth. Table 2 compares three RADI variants with varying maximum reasoning iterations ($K = 10$, 20 and ∞), along with three representative LLM-based planning baselines. As the number of allowed reflection steps increases, both the Success Rate (SR) and Executability Rate (Exe) consistently improve. In particular, the RADI-inf variant (with unbounded reflection) achieves the best performance across all metrics, reaching an average SR of 0.96 and Exe of 0.97, which substantially surpass all baselines. These results suggest that deeper reflective reasoning enables more accurate and feasible action plans.

Table 2: Performance comparison of RADI with different reflection depths versus baseline methods on VirtualHome benchmarks. Success Rate (SR) and Executability Rate (Exe) are reported in decimal form.

Dataset	RADI Framework (Ours)						Existing LLM-based Methods					
	RADI-10		RADI-20		RADI-inf		React		Embodied		MLDT	
	SR	Exe	SR	Exe	SR	Exe	SR	Exe	SR	Exe	SR	Exe
NovelTasks	0.62	0.64	0.82	0.84	**0.92**	**0.94**	0.89	0.91	0.85	0.86	0.33	0.34
NovelScene	0.60	0.60	0.81	0.82	**0.98**	**0.98**	0.87	0.88	0.83	0.87	0.31	0.32
LongTasks	0.58	0.58	0.85	0.86	**0.95**	**0.96**	0.81	0.85	0.74	0.80	0.12	0.12
InDistribution	0.58	0.59	0.87	0.88	**0.99**	**0.99**	0.86	0.89	0.86	0.87	0.34	0.38
Average	0.59	0.60	0.84	0.85	**0.96**	**0.97**	0.86	0.88	0.82	0.85	0.28	0.29

† RADI-10, RADI-20, and RADI-inf refer to our Robotic Action Decomposition and Imagination framework with different maximum repeat generation limits k.

Table 3: Ablation study of RADI framework and the results demonstrate the impact of different components and configuration settings on performance.

Model Variant	NovelTasks		NovelScene		LongTasks		InDistribution		Average	
	SR	Exe	SR	Exe	SR	Exe	SR	Exe	SR	Exe
RADI-inf	0.92	0.94	0.98	0.98	0.95	0.96	0.99	0.99	0.96	0.97
w/o Reflection	0.84	0.85	0.86	0.86	0.86	0.87	0.93	0.94	0.87	0.88
w/o Imagination	0.33	0.34	0.31	0.32	0.12	0.12	0.34	0.38	0.28	0.29
MLDT (Baseline)	0.33	0.34	0.31	0.32	0.12	0.12	0.34	0.38	0.28	0.29

Ablation Study and Relative Contribution. To assess the importance of individual components within the RADI framework, we conducted an ablation study by individually removing the *reflection* and *imagination* modules from RADI-inf. The quantitative results are presented in Table 3. Removing the reflection module leads to a moderate performance drop (average SR: 0.87, Exe: 0.88), whereas eliminating the imagination module causes a dramatic degradation (average SR: 0.28, Exe: 0.29), reducing performance to near-baseline levels.

To better quantify the contribution of each component, we compute the relative performance gains of RADI-inf over its ablated variants, as summarized in Table 4. The relative gain is defined as:

$$\text{Gain} = \frac{\text{RADI}_{\text{inf}} - \text{Variant}}{\text{Variant}} \times 100\%$$

Compared to the no-reflection variant, RADI-inf produces an average gain of +10.3% in SR and +10.2% in Exe. In contrast, the gains over the no-imagination variant increase to +242.9% in SR and +234.5% in Exe. These results highlight that, while reflection introduces steady improvements, the imagination module is the primary driver of generalization and robustness. Overall, the ablation analysis substantiates the hypothesis that effective high-level planning hinges not only on accurate action decomposition but also on iterative simulation and self-correction, which are enabled by the combination of reflection and imagination.

Table 4: Relative performance gain (%) of RADI-inf over ablated variants, across multiple task categories.

Model Variant	Metric	NovelTasks	NovelScene	LongTasks	InDist	Avg
w/o Reflection	SR	+9.5%	+14.0%	+10.5%	+6.5%	+10.3%
	Exe	+10.6%	+14.0%	+10.3%	+5.3%	+10.2%
w/o Imagination	SR	+178.8%	+216.1%	+691.7%	+191.2%	+242.9%
	Exe	+176.5%	+206.3%	+700.0%	+160.5%	+234.5%

6 Conclusion

In this paper, we address the key challenge of utilizing LLM as a world model for robot task planning. We propose the RADI framework, which integrates hierarchical action decomposition, environmental imagination, memory, and reflection to improve the success rate of robotic task planning. By progressively decomposing complex tasks into atomic actions and modeling their outcomes through environment state change prediction, RADI enables LLMs to self-reflect and iteratively improve action sequences. This closed-loop process allows the framework to accumulate experience over time without requiring any parameter updates. Experiments in VirtualHome demonstrated that our framework significantly improves task completion rates while reducing execution failures by abstention and correction mechanism.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Chuyuan Fu, Keerthana Gopalakrishnan, Karol Hausman, et al. Do as i can, not as i say: Grounding language in robotic affordances. *arXiv preprint arXiv:2204.01691*, 04 2022. doi: 10.48550/arXiv.2204.01691.

Jawid Ahmad Baktash and Mursal Dawodi. Gpt-4: A review on advancements and opportunities in natural language processing. *arXiv preprint arXiv:2305.03195*, 2023.

Sebastian Blumenthal, Herman Bruyninckx, Walter Nowak, and Erwin Prassler. A scene graph based shared 3d world model for robotic applications. In *2013 IEEE International Conference on Robotics and Automation*, pages 453–460, 2013. doi: 10.1109/ICRA.2013.6630614.

Rodney Brooks. A robust layered control system for a mobile robot. *IEEE journal on robotics and automation*, 2(1):14–23, 1986.

Sébastien Bubeck, Varun Chadrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, et al. Sparks of artificial general intelligence: Early experiments with gpt-4, 2023.

Georgia Chalvatzaki, Ali Younes, Daljeet Nandha, An Thai Le, Leonardo FR Ribeiro, and Iryna Gurevych. Learning to reason over scene graphs: a case study of finetuning gpt-2 into a robot language model for grounded task planning. *Frontiers in Robotics and AI*, 10:1221739, 2023.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Jingyuan Ma, Rui Li, Heming Xia, Jingjing Xu, Zhiyong Wu, Tianyu Liu, et al. A survey on in-context learning. *arXiv preprint arXiv:2301.00234*, 2022.

Luciano Floridi and Massimo Chiriatti. Gpt-3: Its nature, scope, limits, and consequences. *Minds and Machines*, 30:681–694, 2020.

Cipriano Galindo, Juan-Antonio Fernández-Madrigal, Javier González, and Alessandro Saffiotti. Robot task planning using semantic maps. *Robotics and Autonomous Systems*, 56(11):955–966, 2008. ISSN 0921-8890. doi: https://doi.org/10.1016/j.robot.2008.08.007. Semantic Knowledge in Robotics.

Chandra Sekhar Gatla, Ron Lumia, John Wood, and Greg Starr. An automated method to calibrate industrial robots using a virtual closed kinematic chain. *IEEE Transactions on Robotics*, 23(6): 1105–1116, 2007a. doi: 10.1109/TRO.2007.909765.

Chandra Sekhar Gatla, Ron Lumia, John Wood, and Greg Starr. Calibration of industrial robots by magnifying errors on a distant plane. In *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3834–3841, 2007b. doi: 10.1109/IROS.2007.4398969.

Zhiqi Ge, Hongzhe Huang, Mingze Zhou, Juncheng Li, Guoming Wang, Siliang Tang, and Yueting Zhuang. Worldgpt: Empowering llm as multimodal world model. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 7346–7355, 2024.

David R Ha and Jürgen Schmidhuber. World models. *ArXiv*, abs/1803.10122, 2018. URL `https://api.semanticscholar.org/CorpusID:4807711`.

Marc Hanheide, Moritz Göbelbecker, Graham S. Horn, Andrzej Pronobis, Kristoffer Sjöö, Alper Aydemir, Patric Jensfelt, Charles Gretton, Richard Dearden, Miroslav Janicek, Hendrik Zender, Geert-Jan Kruijff, Nick Hawes, and Jeremy L. Wyatt. Robot task planning and explanation in open and uncertain worlds. *Artificial Intelligence*, 247:119–150, 2017. ISSN 0004-3702. doi: https://doi.org/10.1016/j.artint.2015.08.008. Special Issue on AI and Robotics.

Wenlong Huang, Fei Xia, Ted Xiao, Harris Chan, Jacky Liang, Pete Florence, Andy Zeng, Jonathan James Richard Tompson, Igor Mordatch, Yevgen Chebotar, Pierre Sermanet, Noah Brown, Tomas Jackson, Linda Luu, Sergey Levine, Karol Hausman, and Brian Andrew Ichter. Innermonologue: Embodied reasoning through planning with language models. In *arXiv preprint arXiv:2207.05608*, 2022. URL `https://innermonologue.github.io/`. CoRL 2022 (to appear).

Matteo Iovino, Edvards Scukins, Jonathan Styrud, Petter Ögren, and Christian Smith. A survey of behavior trees in robotics and ai. *Robotics and Autonomous Systems*, 154:104096, 2022.

Dae-Sung Jang, Doo-Hyun Cho, Woo-Cheol Lee, Seung-Keol Ryu, Byeongmin Jeong, Minji Hong, Minjo Jung, Minchae Kim, Minjoon Lee, SeungJae Lee, et al. Unlocking robotic autonomy: A survey on the applications of foundation models. *International Journal of Control, Automation and Systems*, 22(8):2341–2384, 2024.

Yu-qian Jiang, Shi-qi Zhang, Piyush Khandelwal, and Peter Stone. Task planning in robotics: an empirical comparison of pddl-and asp-based systems. *Frontiers of Information Technology & Electronic Engineering*, 20:363–373, 2019.

Changqing Jing, Hongyu Shu, and Yitong Song. Model predictive control for integrated lateral stability and rollover prevention based on a multi-actuator control system. *International Journal of Control, Automation and Systems*, 21(5):1518–1537, 2023.

Jay H Lee. Model predictive control: Review of the three decades of development. *International Journal of Control, Automation and Systems*, 9:415–424, 2011.

Shuang Li, Xavier Puig, Chris Paxton, Yilun Du, Clinton Wang, Linxi Fan, Tao Chen, De-An Huang, Ekin Akyürek, Anima Anandkumar, Jacob Andreas, Igor Mordatch, Antonio Torralba, and Yuke Zhu. Pre-trained language models for interactive decision-making. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 31199–31212. Curran Associates, Inc., 2022.

Brady D Lund and Ting Wang. Chatting about chatgpt: how may ai and gpt impact academia and libraries? *Library hi tech news*, 40(3):26–29, 2023.

Weixin Mao, Weiheng Zhong, Zhou Jiang, Dong Fang, Zhongyue Zhang, Zihan Lan, Fan Jia, Tiancai Wang, Haoqiang Fan, and Osamu Yoshie. Robomatrix: A skill-centric hierarchical framework for scalable robot task planning and execution in open-world. *arXiv preprint arXiv:2412.00171*, 2024.

Chris Paxton, Yotam Barnoy, Kapil Katyal, Raman Arora, and Gregory D. Hager. Visual robot task planning. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 8832–8838, 2019. doi: 10.1109/ICRA.2019.8793736.

Xavier Puig, Kevin Ra, Marko Boben, Jiaman Li, Tingwu Wang, Sanja Fidler, and Antonio Torralba. Virtualhome: Simulating household activities via programs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8494–8502, 2018.

Haofu Qian, Chenjia Bai, Jiatao Zhang, Fei Wu, Wei Song, and Xuelong Li. Discriminator-guided embodied planning for LLM agent. In *The Thirteenth International Conference on Learning Representations*, 2025. URL `https://openreview.net/forum?id=TjP1d8PP8l`.

M. Roth, D. Vail, and M. Veloso. A real-time world model for multi-robot teams with high-latency communication. In *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003) (Cat. No.03CH37453)*, volume 3, pages 2494–2499 vol.3, 2003. doi: 10.1109/IROS.2003.1249244.

Mingyu Shin and Soyi Jung. A survey of behavior tree-based task planning algorithms for autonomous robotic systems. In *2024 15th International Conference on Information and Communication Technology Convergence (ICTC)*, pages 2039–2041. IEEE, 2024.

Ishika Singh, Valts Blukis, Arsalan Mousavian, Ankit Goyal, Danfei Xu, Jonathan Tremblay, Dieter Fox, Jesse Thomason, and Animesh Garg. Progprompt: Generating situated robot task plans using large language models. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11523–11530. IEEE, 2023.

Panagiota Tsarouchi, Sotiris Makris, and George Chryssolouris. Human–robot interaction review and challenges on task planning and programming. *International Journal of Computer Integrated Manufacturing*, 29(8):916–931, 2016.

Yike Wu, Jiatao Zhang, Nan Hu, Lanling Tang, Guilin Qi, Jun Shao, Jie Ren, and Wei Song. Mldt: Multi-level decomposition for complex long-horizon robotic task planning with open-source large language model. In *International Conference on Database Systems for Advanced Applications*, pages 251–267. Springer, 2024.

Zhenyu Wu, Ziwei Wang, Xiuwei Xu, Jiwen Lu, and Haibin Yan. Embodied task planning with large language models. *arXiv preprint arXiv:2307.01848*, 2023.

Jiannan Xiang, Tianhua Tao, Yi Gu, Tianmin Shu, Zirui Wang, Zichao Yang, and Zhiting Hu. Language models meet world models: Embodied experiences enhance language models. *Advances in neural information processing systems*, 36:75392–75412, 2023.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. *Advances in neural information processing systems*, 36:11809–11822, 2023a.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations (ICLR)*, 2023b.

Yu Zhang, Sarath Sreedharan, Anagha Kulkarni, Tathagata Chakraborti, Hankz Hankui Zhuo, and Subbarao Kambhampati. Plan explicability and predictability for robot task planning. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1313–1320, 2017. doi: 10.1109/ICRA.2017.7989155.

Z. Zhang and O. Faugeras. Building a 3d world model with a mobile robot: 3d line segment representation and integration. In *[1990] Proceedings. 10th International Conference on Pattern Recognition*, volume i, pages 38–42 vol.1, 1990. doi: 10.1109/ICPR.1990.118061.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. A survey of large language models. *arXiv preprint arXiv:2303.18223*, 1(2), 2023.

Zhehua Zhou, Jiayang Song, Kunpeng Yao, Zhan Shu, and Lei Ma. Isr-llm: Iterative self-refined large language model for long-horizon sequential task planning. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2081–2088. IEEE, 2024.

A Technical Appendices and Supplementary Material

Technical appendices with additional results, figures, graphs and proofs may be submitted with the paper submission before the full submission deadline (see above), or as a separate PDF in the ZIP file below before the supplementary material deadline. There is no page limit for the technical appendices.

B The detail description of testing dataset

The InDistribution dataset provides baseline performance metrics within familiar environmental configurations, while NovelScenes introduces spatial reconfiguration challenges that assess adaptation to unfamiliar environmental topologies. NovelTasks extends the evaluation paradigm to conceptually novel objectives, testing abstract generalization capabilities rather than spatial adaptability. The LongTasks dataset, containing 1,154 samples with a minimum action threshold of 60 steps per task, represents the apex of complexity with its expanded goal complexity and interaction object diversity.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes] , [No] , or [NA] .
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes] " is generally preferable to "[No] ", it is perfectly acceptable to answer "[No] " provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No] " or "[NA] " is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS Paper Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We list in the abstract and introduction the contributions of this paper.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.

- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We present limitations about the experiment and model in conclusion.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

   Justification: There is no theorem or lemma in this paper.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: The codes are presented in the supplemental material, and the results can be reproduced by running the main program.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: The codes are presented in the supplemental material, and the results can be reproduced by running the main program. The datasets used in this paper are publicly available.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).

- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: The experimental details are presented.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: We set the temperature parameter of the LLMs to 0, so the results are deterministic.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
   - If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: We use one A100-80GB gpu to conduct our experiments.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

Answer: [Yes]

Justification: This paper has no ethics problem.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: We discuss the potential societal impacts in this paper.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: There are no such risks in our paper.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: : We have cited the original papers and included proper license.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [NA]

Justification: We do not release new assets.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.

- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: We do not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: We do not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [Yes]

    Justification: Pretrained LLMs are used as backbones in our method, which is clearly stated in this paper.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.