Counterfactual Planning for Generalizable Agents' Actions

Jiarun Fu[1], Lizhong Ding[1*], Qiuning Wei[1], Yuhan Guo[1], Yurong Cheng[1], Junyu Zhang[1],

[1]School of Computer Science and Technology, Beijing Institute of Technology
{jrfu, weiqiuning, guoyuhan, yrcheng, zhangjunyu}@bit.edu.cn, lizhong.ding@outlook.com.

Abstract

Large language models have revolutionized agent planning by serving as the engine of heuristic guidance. However, LLM-based agents often struggle to generalize across complex environments and to adapt to stochastic feedback arising from environment–action interactions. We propose Counterfactual Planning—a method designed to improve the generalizability and adaptability of agents' actions by inferring causal representations of environmental confounders and performing counterfactual reasoning over planned actions. We formalize the agent planning process as a structural causal model, providing a mathematical formulation for causal analysis of how environmental states influence action generation and how actions affect future state transitions. To support generalizable action planning, we introduce the State Causality Evaluator (SCE), which dynamically infers task-conditioned causal representations from complex environment states; and to enhance adaptability under stochastic feedback, we propose the What-If-Not (WIN) reward, which performs counterfactual interventions to refine actions through causal evaluation. We validate our framework in an open-world environment, where experiments demonstrate improvements in both action generalization and planning adaptability.

Introduction

Large language models (LLMs)—such as DeepSeek (DeepSeek-AI 2024, 2025), GPT (OpenAI 2024a,b), and LLaMA (Touvron et al. 2023)—have changed agent planning by serving as the heuristic engine of agents in embodied intelligence (Wu et al. 2023b; Feng et al. 2025) and multi-agent systems (Tan et al. 2025). Unlike conventional reasoning tasks, such as language understanding (Elazar et al. 2021), knowledge inference (Saikh et al. 2022; Fu et al. 2024), or mathematical logic (Ding et al. 2019a, 2020; Cobbe et al. 2021)—which typically involve static inputs and deterministic reasoning, agent planning introduces unique challenges characterized by long-horizon decision-making under dynamic, partially observable, and stochastic environments. These environments often feature dynamically evolving structures and interactive feedback (Qian et al. 2025). As a result, even minor planning errors can rapidly compound across sequential steps, leading to significant deviations from intended plans (Liu et al. 2024) and eventual task failures (Luo et al. 2024; Zhang et al. 2024a). Therefore, an open challenge is to effectively harness the reasoning capabilities of LLMs to support precise and generalizable agent planning in complex, dynamic environments (Bai et al. 2025; Feng et al. 2025).

From both philosophical and cognitive perspectives, understanding causality is fundamental-if not the ultimate objective—for understanding how humans learn from and interact with the physical world (Pearl 2009; Ding et al. 2018, 2019b; Kaddour et al. 2022; Fu et al. 2025). Inspired by this view, causal learning has been extensively integrated into reinforcement learning (RL) and agent planning, as it enables reasoning about the consequences of actions in dynamic environments (Gupta et al. 2024), as shown in Figure 1(a). Recent advances explore the incorporation of causal reasoning into LLM-based agent planning (Zhang et al. 2024b; Chen et al. 2024; Ashwani et al. 2024), often by equipping LLMs with *static causal knowledge* derived from external priors or manually annotated structures. Representative approaches include Causal-aware LLMs (Chen et al. 2025), which construct task-specific causal templates by assuming access to predefined environment causal graphs. Similarly, CausalFM (Ma et al. 2025) adopts Bayesian causal inference based on static factorization structures. While these methods demonstrate the value of causal priors, they remain fundamentally limited by their reliance on immutable or externally provided causal knowledge. We refer to them as Static-Causal LLM Planning (as shown in Figure 1(b)). Such limitations hinder the ability to update causal representations during planning, making it difficult to capture latent or dynamic confounders—environmental factors that simultaneously influence both observations and action outcomes—especially in open-ended, partially observable domains (Feng et al. 2025). As a result, these approaches lack the generalizability to dynamically infer causal representations or to adapt agent actions based on corresponding stochastic feedback. These limitations reveal a gap in existing LLM-based agent planning methods, giving rise to two key challenges:

1. How can we enable generalizable agent planning across complex, dynamic environments by causally representing environmental confounders?
2. How can we achieve adaptive action planning under stochastic feedback resulting from environment-action interactions through causal reasoning?

*Corresponding Author

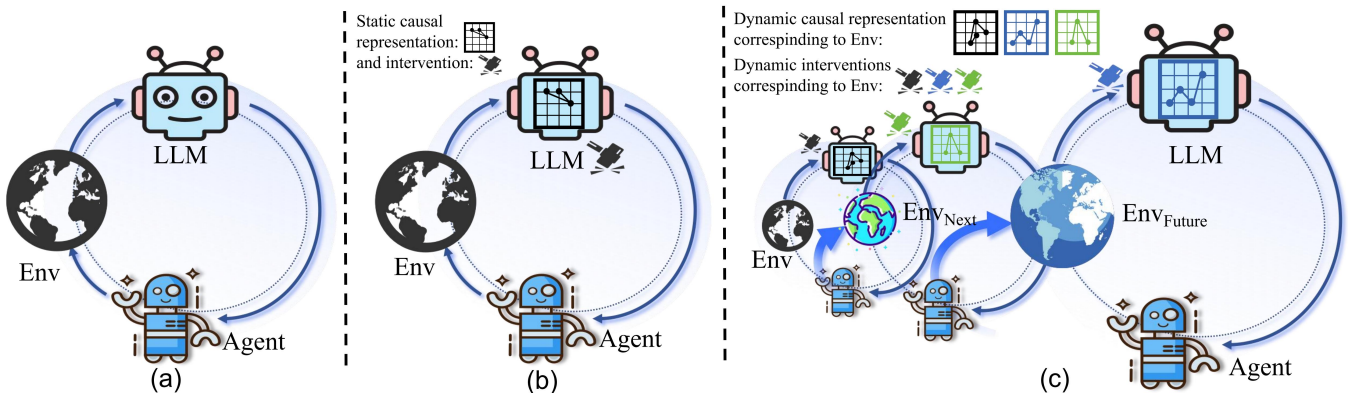

Figure 1: Comparison of Existing Approaches and Our Method: (a) **Reactive LLM Planning**: LLMs assist RL agents in interacting with the environment. (b) **Static-Causal LLM Planning**: LLMs leverage fixed, externally defined causal priors to guide planning. (c) **Dynamic-Causal LLM Planning (Ours)**: LLMs dynamically infer causal representations and perform counterfactual reasoning, enabling generalization and adaptive action planning in complex, dynamic environments (Env).

To address these challenges, we propose Counterfactual Planning—a causal reasoning framework that enhances LLM-based agent planning by incorporating causal representations of environmental confounders and counterfactual reasoning over planned actions, thereby enabling Dynamic-Causal LLM Planning (as illustrated in Figure 1(c)). We formalize the planning process as a Structural Causal Model (SCM), enabling causal analysis over how environmental states influence action generation and how actions affect subsequent state transitions. Building on this formulation, we introduce the State Causality Evaluator (SCE) to improve planning generalization. SCE targets the action generation component of the SCM, dynamically inferring task-specific causal representations from complex environment states to identify latent confounders and inject interpretable causality into the agent's decision process. To enhance adaptability under stochastic feedback, we design the What-If-Not (WIN) reward, which focuses on the state transition component of the SCM. WIN leverages counterfactual interventions to assess the causal advantage of actions in influencing future states, enabling real-time adaptation of planning based on fine-grained causal feedback. We validate Counterfactual Planning across a diverse set of 22 tasks in the open-world Crafter environment. Empirical results demonstrate consistent improvements in both generalization and adaptability over standard LLM-based planning methods, highlighting the benefit of integrating dynamic causal representation and counterfactual reasoning into LLM-based agent planning. Our main contributions are:

1. We formulate the LLM-based agent planning process as a Structural Causal Model (SCM), enabling explicit causal analysis over both action generation and state transitions in complex, dynamic environments.
2. To support generalization across diverse environments, we design the State Causality Evaluator (SCE), which operates on the action generation component of the SCM by dynamically injecting interpretable causality into the action planning.
3. To improve planning adaptability, we introduce the What-If-Not (WIN) reward, which targets the transition function component of the SCM, enabling agents' adaptation to environmental stochastic feedback.

Preliminaries

To provide a formal basis for the subsequent development of our framework, we describe the agent–environment interaction using the Factored Markov Decision Processes (FMDPs), which formalize how agents make sequential decisions under task-conditioned, partially observable environments. We then present Structural Causal Models (SCMs) as a mathematical model for causal analysis of the planning process.

FMDP Formulation for LLM-Based Agent Planning

LLM-based agents operate in partially observable, stochastic environments over multiple time steps. We formalize this process as a Factored Markov Decision Process (FMDP), where each task is specified by a natural language instruction that conditions both policy decisions and environment dynamics.

Formally, an FMDP is defined as a tuple $\langle \mathcal{L}, \mathcal{S}, \mathcal{A}, \mathcal{F}, \mathcal{R} \rangle$, where $\mathcal{L}$ denotes the space of natural language tasks, and each task $l \in \mathcal{L}$ induces a grounded MDP with: a state space $\mathcal{S}$, action set $\mathcal{A}$, transition function $F_l : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$, and reward function $R_l : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$. Here, $F_l(s, a)$ denotes the task-specific stochastic transition function, capturing the uncertainty in state transitions resulting from executing action a in state s under task l.

We assume that the LLM-based agent encodes prior context through its evolving hidden state, serving as an implicit memory across time steps. At time t, the policy π_θ generates an action conditioned on the current state s_t and the task instruction l:

$$a_t \sim \pi_\theta(\cdot \mid s_t, l), \quad r_t = R_l(s_t, a_t), \quad s_{t+1} \sim F_l(\cdot \mid s_t, a_t). \tag{1}$$

Structural Causal Model (SCM)

To model the underlying decision and transition mechanisms in agent planning, we adopt the formalism of a Structural Causal Model (SCM) (Pearl 2009). An SCM $\mathcal{M}$ is defined as a 3-tuple $\langle \mathbb{V}, \mathbb{U}, \mathbb{F} \rangle$, where $\mathbb{V}$ denotes a set of *endogenous variables*, $\mathbb{U}$ a set of *exogenous variables*, and $\mathbb{F}$ a set of *deterministic structural functions*. Each function $f_i \in \mathbb{F}$ maps a subset of variables to an endogenous variable $v_i \in \mathbb{V}$:

$$f_i : \mathbb{V}_i^{pa} \times \mathbb{U}_i \mapsto v_i, \quad \text{where } \mathbb{V}_i^{pa} \subseteq \mathbb{V},\ \mathbb{U}_i \subseteq \mathbb{U}.$$

The SCM induces a causal graph—typically a directed acyclic graph (DAG)—where each variable v_i is causally influenced by its parents $\mathbb{V}_i^{pa}$. Given a causal graph induced by the SCM, the joint observational distribution factorizes along its structure as:

$$f(\mathbb{V} \mid \mathbb{U}) = \prod_{v_i \in \mathbb{V}} f_i(v_i \mid \mathbb{V}_i^{pa}, \mathbb{U}_i).$$

Mathematical Formalization: Structural Causal Modeling for Agent Planning

We represent the LLM-based agent planning process as *Structural Causal Models (SCMs)*. The reward function is excluded from the structural causal model, as it serves as a post hoc evaluation signal that does not causally influence state transitions or decision processes.

We define the agent planning SCM $\mathcal{M}_{\text{AP}}$ as a composition of two sub-models that capture the agent's decision and the environment's response mechanisms, respectively:

- **Action SCM:** $\mathcal{M}^{(a)} = \langle \mathbb{V}^{(a)}, \mathbb{U}^{(a)}, \mathbb{F}^{(a)} \rangle$

$$\mathbb{V}^{(a)} = \{a_t \mid t = 0, \ldots, K-1\},$$
$$\mathbb{U}^{(a)} = \{s_t, l \mid t = 0, \ldots, K-1\},$$
$$\mathbb{F}^{(a)} = \{\pi_\theta\}, \quad a_t \sim \pi_\theta(\cdot \mid s_t, l).$$

  Here, the LLM-based policy π_θ maps the current state and task to an action, where K represents the total number of decision steps considered in a single trajectory and a_t has no parent variables.

- **Transition SCM:** $\mathcal{M}^{(s)} = \langle \mathbb{V}^{(s)}, \mathbb{U}^{(s)}, \mathbb{F}^{(s)} \rangle$

$$\mathbb{V}^{(s)} = \{s_t \mid t = 0, \ldots, K-1\},$$
$$\mathbb{U}^{(s)} = \{a_t \mid t = 0, \ldots, K-1\},$$
$$\mathbb{F}^{(s)} = \{f_{s_{t+1}}\}, \quad s_{t+1} \sim f_{s_{t+1}}(\cdot \mid s_t, a_t).$$

  The function $f_{s_{t+1}}$ models the environment's dynamic transition. For brevity, we omit the condition l in $f_{s_{t+1}}$.

Together, the action SCM $\mathcal{M}^{(a)}$ and transition SCM $\mathcal{M}^{(s)}$ jointly define the agent planning model $\mathcal{M}_{\text{AP}}$. Under this formulation, the observational distribution of each sub-model factorizes according to its internal causality:

$$f(\mathbb{V}^{(\cdot)} \mid \mathbb{U}^{(\cdot)}) = \prod_{v_t \in \mathbb{V}^{(\cdot)}} f_t^{(\cdot)}(v_t \mid \mathbb{V}_t^{pa}, \mathbb{U}_t), \quad (\cdot) \in \{a, s\},$$

where $f_t^{(a)}$ corresponds to the agent's action distribution (e.g., $\pi_\theta(a_t \mid s_t, l)$), and $f_t^{(s)}$ models the environment's stochastic transition dynamics (e.g., $f_{s_{t+1}}(\cdot \mid s_t, a_t)$). This completes the structural definition of $\mathcal{M}_{\text{AP}}$, providing a modular causal representation of both the agent's decision process and the environment's dynamics.

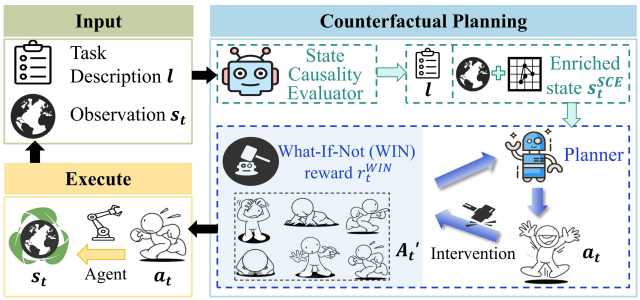

Figure 2: Counterfactual Planning begins with the task description and current observation as inputs. The SCE enhances the state with causal representations, which is then used by the planner to generate actions. Meanwhile, the WIN module simulates counterfactual actions and evaluates their outcomes to refine planning. The selected action is executed in the environment, and the resulting feedback closes the loop for continual causal reasoning and adaptation.

Method: Counterfactual Planning

Grounded in the formulation $\mathcal{M}_{\text{AP}}$, we now present Counterfactual Planning, a framework designed to enhance both generalization and adaptability in LLM-based agent planning by combining causal representation inference with counterfactual reasoning. This framework (as shown in Figure 2) comprises two complementary modules—each corresponding to one sub-SCM in $\mathcal{M}_{\text{AP}}$. State Causality Evaluator (SCE) enhances generalization under $\mathcal{M}^{(a)}$, while What-If-Not (WIN) reward enables adaptive planning under $\mathcal{M}^{(s)}$.

Problem Formulation

Within the SCM formalism $\mathcal{M}_{\text{AP}} = \mathcal{M}^{(a)} \cup \mathcal{M}^{(s)}$, LLM-based planning can be causally decomposed into two components: *action generation* governed by $\mathcal{M}^{(a)}$, and *state transition* governed by $\mathcal{M}^{(s)}$. Despite the expressive capacity of LLMs, agents still encounter two fundamental planning challenges in dynamic environments:

- **Causal Generalization under Latent Confounders:** In $\mathcal{M}^{(a)}$, actions are generated by $a_t \sim \pi_\theta(\cdot \mid s_t, l)$, where both the task l and state s_t are exogenous inputs. Since interventions cannot be directly performed on l and s_t, generalization relies on uncovering and leveraging latent causal factors within s_t. Particularly, as l remains fixed during each episode, enriching the causal representation of s_t becomes essential for producing generalizable actions.
- **Causal Adaptation under Stochastic Feedback:** In $\mathcal{M}^{(s)}$, transitions follow $s_{t+1} \sim f_{s_{t+1}}(\cdot \mid s_t, a_t)$, where stochasticity in the transition dynamics makes it difficult to assess the effectiveness of actions based solely on observed outcomes. To adapt effectively, the agent must reason over the counterfactual interventions to a_t.

We now formalize these challenges as planning problems:

Definition 1 (Causal Generalization Problem)**.** *The Causal Generalization Problem refers to the challenge of generating generalizable actions under the action SCM $\mathcal{M}^{(a)}$ by inferring and utilizing latent causal representations over confounder variables embedded in the environment state s_t.*

Definition 2 (Causal Adaptation Problem)**.** *The Causal Adaptation Problem refers to the challenge of adapting actions under the transition SCM $\mathcal{M}^{(s)}$ by leveraging counterfactual interventions to assess the causal advantage of actions a_t on stochastic transition outcomes $s_{t+1} \sim f_{s_{t+1}}(s_t, a_t)$.*

Inferring Causal Representations via State Causality Evaluator

To solve the Causal Generalization Problem (Definition 1), we introduce the **State Causality Evaluator (SCE)**—a module that enhances the causal informativeness of s_t by inferring task-conditioned causal representations from complex environment states conditioned on the task l. By embedding task-conditioned causal representations into the input, SCE enables generalized action generation under $\mathcal{M}^{(a)}$.

As illustrated in Figure 3, the SCE includes three steps: *symbolic decomposition*, *task-conditioned grounding*, and *causal representation inference*.

1. Symbolic Decomposition. We first extract symbolic representations from the environment state and task instruction:

$$f_{\text{decomp}} : \mathcal{L} \times \mathcal{S} \to 2^{\mathcal{C}} \times 2^{\mathcal{V}}, \quad (\mathcal{C}_t, \mathcal{V}_t) = f_{\text{decomp}}(l, s_t),$$

where $\mathcal{C}$ and $\mathcal{V}$ denote the global concept and variable universes, and $2^{\mathcal{C}}$, $2^{\mathcal{V}}$ are their power sets. Here $\mathcal{C}_t = \{c_i\} \subseteq \mathcal{C}$ and $\mathcal{V}_t = \{v_j\} \subseteq \mathcal{V}$ are the subsets extracted at time t.

2. Task-Conditioned Grounding. This step refines the symbolic representations by aligning them with task semantics. It is composed of two sub-steps:

- **(a) Concept Identification:** Given the initial concept set $\mathcal{C}_t$ and task instruction l, we define an identifying function:
$$f_{\text{concept}} : \mathcal{L} \times 2^{\mathcal{C}} \to 2^{\mathcal{C}}, \quad \mathcal{C}'_t = f_{\text{concept}}(l, \mathcal{C}_t),$$
where $\mathcal{C}'_t \subseteq \mathcal{C}_t$ retains only those identified symbolic concepts that are causally relevant to l.
- **(b) Variable Characterization:** Using the identified concept set $\mathcal{C}'_t$ and the candidate variable set $\mathcal{V}_t$, we define a characterization function:
$$f_{\text{var}} : 2^{\mathcal{V}} \times 2^{\mathcal{C}} \to 2^{\mathcal{V}}, \quad \mathcal{V}'_t = f_{\text{var}}(\mathcal{V}_t, \mathcal{C}'_t),$$
where $\mathcal{V}'_t \subseteq \mathcal{V}_t$ contains task-relevant variables that are semantically aligned with some $c_i \in \mathcal{C}'_t$.

3. Causal Representation Inference. Given the task-conditioned sets $\mathcal{V}'_t$ and $\mathcal{C}'_t$, SCE infers a binary causal adjacency matrix:

$$\begin{aligned} &f_{\text{infer}} : 2^{\mathcal{V}} \times 2^{\mathcal{C}} \times \mathcal{L} \times \mathcal{S} \to \{0,1\}^{|\mathcal{V}'_t| \times |\mathcal{V}'_t|}, \\ &\text{CR}_t = f_{\text{infer}}(\mathcal{V}'_t, \mathcal{C}'_t, l, s_t), \end{aligned} \tag{2}$$

where $\text{CR}_t[i, j] = 1$ indicates a directed causal edge $v'_i \to v'_j$ under context (l, s_t). The adjacency matrix CR_t defines the symbolic causal representation of confounder variables' causal graph in s_t conditioned on the current task.

Causal Representation Injection into Policy Context To inject the inferred causal representations into the agent's decision process, we augment the input state to the policy model as follows:

$$s_t^{\text{SCE}} = \texttt{Augment}(s_t, \text{CR}_t), \quad a_t \sim \pi_\theta(\cdot \mid s_t^{\text{SCE}}, l),$$

where `Augment` is a function that concatenates the original state s_t with the causal representation CR_t inferred by the SCE. The causally-Enriched state representation s_t^{SCE} enables π_θ to reason over learned causal dependencies, thereby supporting generalizable action generation under $\mathcal{M}^{(a)}$.

Counterfactual Reasoning via What-If-Not Reward

To address the Causal Adaptation Problem (Definition 2), we propose the What-If-Not (WIN) reward, enabling counterfactual evaluation of actions a_t under the transition SCM $\mathcal{M}^{(s)}$. WIN estimates the causal influence of each action through counterfactual simulations, facilitating more adaptive planning in the presence of stochastic environmental dynamics.

Counterfactual Interventions under Transition SCM We formalize the structured counterfactual interventions in WIN over the action input a_t to the transition mechanism $f_{s_{t+1}}$ within the Transition SCM $\mathcal{M}^{(s)}$. To operationalize this, we define the intervention function:

$$f_{\text{intv}} : \mathcal{S} \times \mathcal{A} \times \mathcal{L} \times \Psi \to \mathcal{A}, \quad \tilde{a}_t^{(\psi)} = f_{\text{intv}}(s_t, a_t, l, \psi)$$

where $\psi \in \Psi$ denotes the intervention type from a discrete set $\Psi = \{\texttt{wait}, \texttt{opp}, \texttt{ran}\}$, and $\tilde{\mathcal{A}}_t = \{\tilde{a}_t^{(\psi)}\}_{\psi \in \Psi}$ denotes the set of counterfactual actions induced by applying f_{intv} over different ψ.

We define three canonical intervention types:

- **Wait** ($\psi = \texttt{wait}$): A null operation that simulates inaction:
$$\tilde{a}_t^{(\texttt{wait})} = f_{\text{intv}}(s_t, a_t, l, \texttt{wait}) = a^{\text{noop}} \in \mathcal{A},$$
where a^{noop} denotes a special no-op action in the action space. In practice, this corresponds to skipping an action at time t, resulting in no interaction with the environment.
- **Opposite** ($\psi = \texttt{opp}$): An inverse behavior defined over a symbolic action space:
$$\tilde{a}_t^{(\texttt{opp})} = f_{\text{intv}}(s_t, a_t, l, \texttt{opp}) = f_{\text{inv}}(a_t),$$
where f_{inv} applies a symbolic inversion using a predefined mapping (e.g., $\texttt{left} \mapsto \texttt{right}$, $\texttt{build} \mapsto \texttt{destroy}$).
- **Random** ($\psi = \texttt{ran}$): A random alternative sampled from the task-conditioned action set:
$$\tilde{a}_t^{(\texttt{ran})} = f_{\text{intv}}(s_t, a_t, l, \texttt{ran}) \in \mathcal{A}_{\text{valid}}(l) \setminus \{a_t\}.$$
In practice, a random action is sampled from the valid action set without regard to its utility for the current task.

Each counterfactual action $\tilde{a}_t^{(\psi)}$ is executed in the transition function to yield a corresponding counterfactual state:

$$\tilde{s}_{t+1}^{(\psi)} \sim f_{s_{t+1}}(\cdot \mid s_t, \tilde{a}_t^{(\psi)}).$$

Note that $\tilde{s}_{t+1}^{(\psi)}$ is not used to compute $\tilde{r}_t^{(\psi)}$ directly; it is used to validate each counterfactual transition.

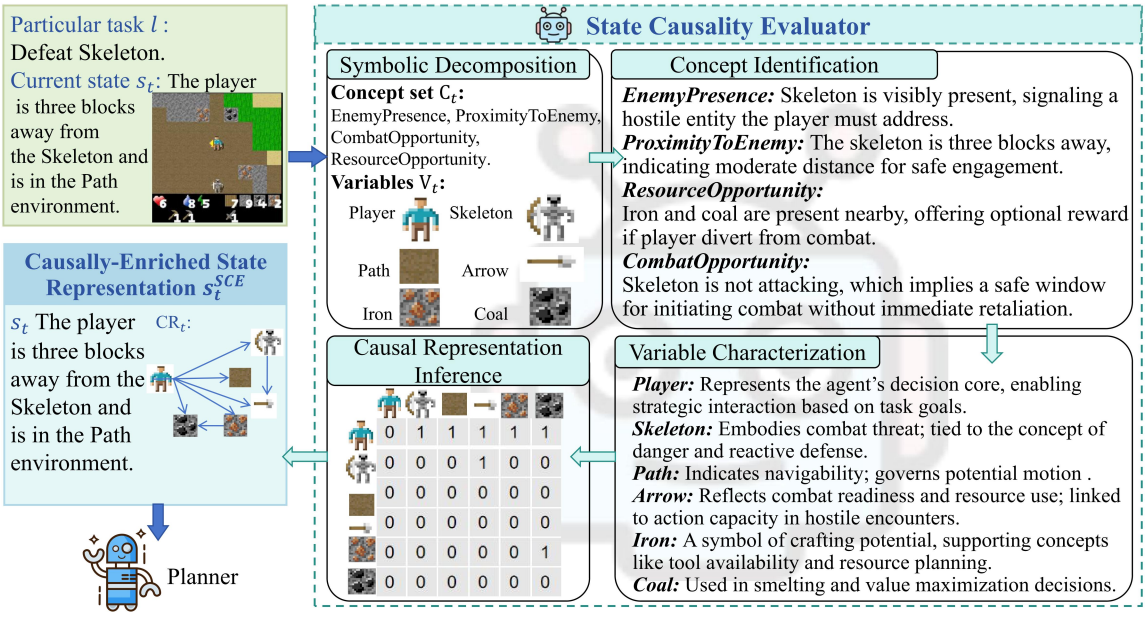

Figure 3: **State Causality Evaluator (SCE)** injects task-conditioned causality into the agent's decision context. Given a task instruction (e.g., *"Defeat Skeleton"*) and the current environment state, SCE performs symbolic decomposition to extract concepts such as `EnemyPresence`, `ProximityToEnemy`, and `ResourceOpportunity`, which are grounded to entities like `Skeleton`, `Arrow`, `Iron`, and `Coal`. Through variable characterization and causal representation inference, SCE infers a symbolic causal representation (e.g., `Skeleton` $\rightarrow$ `PlayerHealth`, `Arrow` $\rightarrow$ `CombatOpportunity`) that reflects confounding relationships under $\mathcal{M}^{(a)}$. This graph is then injected into the policy input as an enriched causal state s_t^{SCE}, enabling π_θ to reason over structural dependencies and generate actions that generalize across diverse tasks and contexts

Counterfactual Causal Effect Estimation Let $r_t = f_R(s_t, a_t)$ denote the factual reward. For each counterfactual action $\tilde{a}_t^{(\psi)} \in \tilde{\mathcal{A}}_t$, we compute the counterfactual reward:

$$\tilde{r}_t^{(\psi)} := f_R(s_t, \tilde{a}_t^{(\psi)})$$

Finally, we define the **WIN score** as the sigmoid-based causal advantage:

$$r_t^{\text{WIN}} := \frac{1}{|\Psi|} \sum_{\psi \in \Psi} \sigma(r_t - \tilde{r}_t^{(\psi)}), \quad \text{where } \sigma(x) = \frac{1}{1+e^{-x}}.$$

The sigmoid function σ provides a smooth measure of causal advantages over each counterfactual intervention, ensuring bounded comparison in stochastic environments. It avoids false positives in action revision and reflects whether a_t is causally optimal within the realistic intervention space $\tilde{\mathcal{A}}_t$.

Action Adaptation via Counterfactual Feedback If the causal advantage score $r_t^{\text{WIN}} \leq 0.5$, the agent initiates an action adaptation step, as this suggests that the "what-if-not" actions could potentially achieve higher outcomes, inducing the agent to revise its plan in favor of more effective alternatives.

$$a_t^{\text{new}} \sim \pi_\theta \left(\cdot \mid s_t,\ (a_t, r_t, \tilde{\mathcal{A}}_t, r_t^{\text{WIN}}),\ l \right).$$

Counterfactual Planning Algorithm

Counterfactual Planning operates over the structural causal model $\mathcal{M}_{\text{AP}} = \mathcal{M}^{(a)} \cup \mathcal{M}^{(s)}$, which provides the causal foundations for action generation and transition evaluation. The full process is summarized in Algorithm 1.

Note: In this work, the functions f_{decomp}, f_{concept}, f_{var}, f_{infer}, and f_{intv} are implemented via LLM-based prompt querying. To minimize performance variability arising from model capacity, each function is explicitly embedded into a well-scoped prompt with fixed structure, output format, and role instruction, serving as a functional abstraction over LLM behavior. Alternative implementations—such as symbolic parsing (Sheth, Roy, and Gaur 2023) or neural extractors with structural priors (Karpas et al. 2022)—are also possible.

Experiment

To rigorously evaluate the effectiveness of our Counterfactual Planning framework, we conduct experiments in the open-ended Crafter environment. Our goal is to assess whether causal modeling and counterfactual reasoning can enhance the generalization and adaptability of LLM-based agents in dynamic, partially observable, and multi-task settings.

Our agent is built upon Qwen2.5-72B-Instruct (Team 2024), integrated with the proposed SCE and WIN modules. All experiments are run on an 8×RTX A6000 GPU cluster. We implement counterfactual planning using an LLM-based prompt query; see Appendix for details.

Experimental Environment: Crafter

Crafter (Hafner 2022) is a 2D Minecraft-inspired environment characterized by partial observability, stochastic feed-

Algorithm 1: Counterfactual Planning

Require: Policy π_θ, reward function f_R, transition model $f_{s_{t+1}}$, task instruction l, planning horizon K

Initialize state s_0
for $t = 0, 1, \ldots, K-1$ **do**
  // Causal Representation Inference under $\mathcal{M}^{(a)}$
  $(\mathcal{C}_t, \mathcal{V}_t) \leftarrow f_{\text{decomp}}(l, s_t)$
  $\mathcal{C}'_t \leftarrow f_{\text{concept}}(l, \mathcal{C}_t)$
  $\mathcal{V}'_t \leftarrow f_{\text{var}}(\mathcal{V}_t, \mathcal{C}'_t)$
  $\text{CR}_t \leftarrow f_{\text{infer}}(\mathcal{V}'_t, \mathcal{C}'_t, l, s_t)$
  $s_t^{\text{SCE}} \leftarrow \texttt{Augment}(s_t, \text{CR}_t)$
  $a_t \sim \pi_\theta(\cdot \mid s_t^{\text{SCE}}, l)$
  $r_t \leftarrow f_R(s_t, a_t)$
  // Simulated Counterfactual Evaluation via WIN
  $\tilde{\mathcal{A}}_t = \{\tilde{a}_t^{(\psi)} \leftarrow f_{\text{intv}}(s_t, a_t, l, \psi) \mid \psi \in \Psi\}$
  for each $\tilde{a}_t^{(\psi)} \in \tilde{\mathcal{A}}_t$ **do**
    $\tilde{r}_t^{(\psi)} \leftarrow f_R(s_t, \tilde{a}_t^{(\psi)})$
  end for
  $r_t^{\text{WIN}} \leftarrow \frac{1}{|\Psi|} \sum_{\psi \in \Psi} \sigma(r_t - \tilde{r}_t^{(\psi)})$
  if $r_t^{\text{WIN}} \leq 0.5$ **then**
    $a_t^{\text{new}} \sim \pi_\theta(\cdot \mid s_t,\ (a_t, r_t, \tilde{\mathcal{A}}_t, r_t^{\text{WIN}}),\ l)$
    $a_t \leftarrow a_t^{\text{new}}$
  end if
  $s_{t+1} \sim f_{s_{t+1}}(\cdot \mid s_t, a_t)$
end for

back, and a structured achievement system with 22 predefined goals. Agents operate on a 64×64 map with a 9×7 field of view, interacting with natural resources (e.g., trees, coal, iron), hostile entities (e.g., zombies), and craftable items. Solving tasks requires long-horizon planning involving sequential exploration, resource collection, crafting, and combat. These properties make Crafter a challenging testbed for planning under uncertainty. For counterfactual evaluation, we implement WIN using a cloned simulation environment that replicates the current state and evaluates each counterfactual action independently. This ensures factual trajectory integrity while enabling offline estimation of counterfactual outcomes.

Evaluation Metrics

We evaluate agent performance using two key metrics: the **success rate** of individual achievements and the **overall achievement score**. The latter is computed as the geometric mean of success rates across the 22 tasks:

$$\text{Score} = \exp\left(\frac{1}{N}\sum_{i=1}^{N} \ln(1 + s_i)\right) - 1,$$

where s_i is the success rate for the i-th achievement. This metric penalizes uneven performance and rewards agents that generalize across all tasks.

Baselines

To evaluate the effectiveness of our framework, we compare it against a diverse set of baselines covering RL, LLM-based agents, human heuristics, and causality-aware methods. All baselines are chosen for strong performance under limited training budgets (e.g., 1M environment steps) and represent key paradigms in Crafter planning.

(1) Reinforcement Learning Agents. We include widely used RL methods in Crafter, such as the value-based agent Rainbow (Hessel et al. 2018), model-based planners DreamerV3 (Hafner 2022) and Curious Replay (Kauvar et al. 2023), and policy-gradient methods like PPO (ResNet) (Moon et al. 2024) and LSTM-SPCNN (Stanić et al. 2022), which introduce spatial or recurrent inductive biases.

(2) LLM-based and Causality-Aware Methods. We compare against AdaRefiner (Zhang and Lu 2024), a prompt-based LLM agent with adaptive refinement, and Causal-aware LLMs (Chen et al. 2025), which use static causal reasoning to assess the impact of explicit counterfactual modeling.

(3) Human and Rule-Based References. We include scores from Human Experts (Hafner 2022) as performance upper bounds, and rule-based systems such as SPRING (Wu et al. 2023a) and Achievement Distillation (Moon et al. 2024), which exploit domain knowledge or structured rewards.

Results and Analysis

The experimental results in Table 1 and Figure 4 demonstrate the effectiveness of our **Counterfactual Planning (CP)** framework across a wide range of tasks in the Crafter environment. Compared with RL-based agents and LLM-based planners, our method achieves consistently higher task success rates, especially on complex, multi-step goals that require long-term causal reasoning. This confirms the advantage of explicitly integrating structural causal modeling into the planning process. In contrast to *Causal-aware LLMs* (Causal LLMs), which inject static causal priors or heuristic symbolic prompts without adapting to feedback, our framework performs *dynamic causal inference* during interaction. The SCE extracts task-specific causal representations from the current state, while the WIN conducts stepwise counterfactual evaluations to quantify causal advantage. Together, these components enable the agent to discover latent confounders, revise actions, and adaptively refine its decision-making process under stochastic transitions. As shown in Figure 4, this leads to superior performance in semantically challenging tasks where prior methods often fail due to limited causal representation.

Compared to approaches like Achievement Distill or SPRING, which rely on fixed domain priors, our method maintains performance gains even on semantically complex goals, such as multi-step crafting and exploration. This reflects its ability to iteratively refine structural understanding and adapt to task uncertainty over time.

Ablation Study

We conduct ablation experiments to evaluate the contributions of the SCE and the WIN. Results in Table 2 show that removing either component causes notable performance drops, validating their respective roles: SCE enhances generalization via task-conditioned causal representation, while WIN enables adaptive refinement via counterfactual evaluation. When both modules are removed, performance degrades to the level of standard LLM-based RL agents, indicating

Metric	Ours		RL-based				
	CP	CP	Rainbow	PPO (ResNet)	DreamerV3	Curious Replay	LSTM-SPCNN
Score (%)	20.4 ± 0.72	36.1 ± 0.4	4.3 ± 0.2	15.6 ± 1.6	14.77 ± 1.6	19.4 ± 1.6	12.1 ± 0.8
Steps	1M	5M	1M	1M	1M	1M	1M

Metric	LLM-based / Causal-aware				Human / Rule-based		
	AdaRefiner	AdaRefiner	Causal LLMs	Causal LLMs	Achievement Distill.	Human Expert	SPRING
Score (%)	15.8 ± 1.4	28.2 ± 1.8	18.9 ± 0.53	33.6 ± 0.02	21.8 ± 1.4	50.5 ± 6.8	27.3 ± 1.2
Steps	1M	5M	1M	5M	1M	0	0

Table 1: We report the average success rate (%) across 22 diverse tasks at two training steps (1M and 5M) to evaluate generalization and efficiency. The table is divided into three major categories: (i) our method (**CP**, short for Counterfactual Planning), (ii) reinforcement learning baselines, and (iii) language-model-based and human/rule-based planners. Our method consistently outperforms RL-based agents, such as PPO and DreamerV3, at both early (1M) and later (5M) stages, demonstrating superior sample efficiency and planning quality. Compared to LLM-based and causal-aware baselines (e.g., AdaRefiner, Causal LLMs), CP yields significant gains. Notably, human experts still outperform all methods, but CP narrows the gap more than any neural baseline. The scores are reported as mean ± standard deviation over 5 random seeds. Our results highlight the effectiveness of integrating causal modeling and counterfactual reasoning into LLM-based action planning.

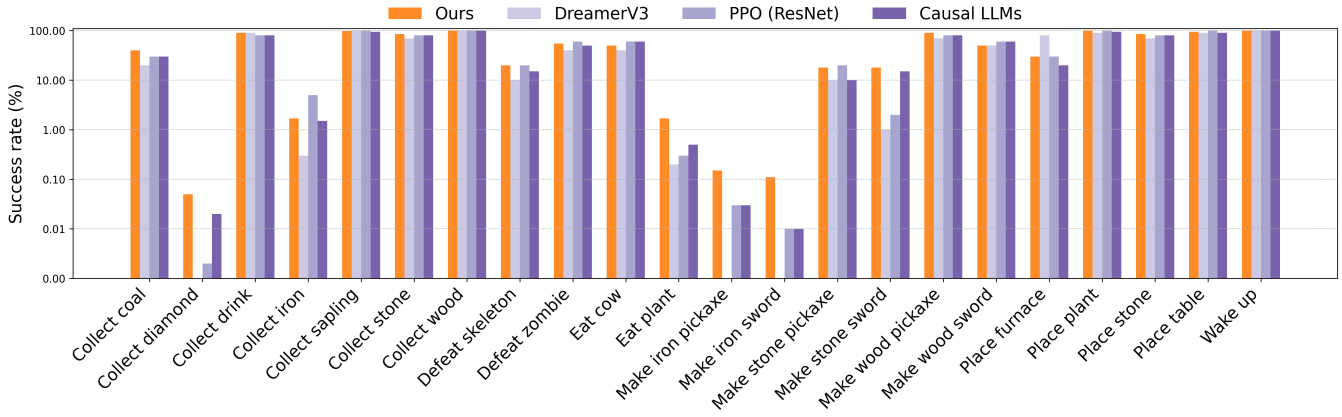

Figure 4: Per-task success rates across 22 Crafter achievements under different methods.

Method (@1M)	Score (%)
Counterfactual Planning	**20.4 ± 0.72**
Ours w/o SCE	16.44 ± 0.08
Ours w/o WIN	17.82 ± 0.13
Ours w/o SCE and WIN	15.6 ± 1.66

Table 2: Ablation study at 1M steps showing the impact of removing the SCE and WIN modules. Both components contribute to performance improvements.

that SCE and WIN are not only effective in isolation but also structurally complementary in supporting causal planning. Together, SCE and WIN provide a structurally integrated mechanism for both generalization and adaptability.

Conclusion

We presented Counterfactual Planning, a dynamic causal reasoning framework for enhancing LLM-based agent planning in complex, stochastic environments. We define a Structural Causal Model formulation to enable principled modeling of how environment states influence action generation and how actions affect subsequent state transitions. To instantiate this framework, we introduced two key components: (1) the State Causality Evaluator, which constructs task-conditioned causal representations from environment observations to support generalizable planning, and (2) the What-If-Not (WIN) reward, which performs structured counterfactual interventions to refine the actions based on stochastic transitions. Empirical results on the Crafter confirm that our method outperforms RL- and LLM-based baselines across a range of tasks and training regimes.

We hope this work provides useful insights for explorations into incorporating causality into foundation model–based agent planning systems, not only by demonstrating empirical gains in challenging open-world environments, but also by offering a general causal formulation that can be extended to other tasks, architectures, and domains where robust, interpretable, and adaptive decision-making is required.

Acknowledgements

This work was supported by the National Key Research and Development Program of China under (Grant No. 2022YFB2703100), the Joint Funds of the National Natural Science Foundation of China under (Grant No. U22A2099), the General Program of the National Natural Science Foundation of China under (Grant No. 62376028), the Excellent Young Scientists Fund (Overseas) of the National Natural Science Foundation of China, and the National Key Scientific Instruments and Equipment Development Project under (Grant No. 62427808).

References

Ashwani, S.; Hegde, K.; Mannuru, N. R.; Sengar, D. S.; Jindal, M.; Kathala, K. C. R.; Banga, D.; Jain, V.; and Chadha, A. 2024. Cause and effect: can large language models truly understand causality? In *AAAI*.

Bai, C.; Zhang, Y.; Qiu, S.; Zhang, Q.; Xu, K.; and Li, X. 2025. Online preference alignment for language models via count-based exploration. *arXiv:2501.12735*.

Chen, S.; Xu, M.; Wang, K.; Zeng, X.; Zhao, R.; Zhao, S.; and Lu, C. 2024. CLEAR: Can Language Models Really Understand Causal Graphs? *In Findings of EMNLP*.

Chen, W.; Zhang, J.; Zhu, H.; Xu, B.; Hao, Z.; Zhang, K.; Ye, J.; and Cai, R. 2025. Causal-aware Large Language Models: Enhancing Decision-Making Through Learning, Adapting and Acting. *In IJCAI*.

Cobbe, K.; Kosaraju, V.; Bavarian, M.; Chen, M.; Jun, H.; Kaiser, L.; Plappert, M.; Tworek, J.; Hilton, J.; Nakano, R.; Hesse, C.; and Schulman, J. 2021. Training Verifiers to Solve Math Word Problems. *arXiv:2110.14168*.

DeepSeek-AI. 2024. DeepSeek-V3 Technical Report. *arXiv:2412.19437*.

DeepSeek-AI. 2025. DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning. *arXiv:2501.12948*.

Ding, L.; Liao, S.; Liu, Y.; Liu, L.; Zhu, F.; Yao, Y.; Shao, L.; and Gao, X. 2020. Approximate kernel selection via matrix approximation. *IEEE Transactions on Neural Networks and Learning Systems*, 31.

Ding, L.; Liao, S.; Liu, Y.; Yang, P.; and Gao, X. 2018. Randomized kernel selection with spectra of multilevel circulant matrices. In *AAAI*.

Ding, L.; Liu, Z.; Li, Y.; Liao, S.; Liu, Y.; Yang, P.; Yu, G.; Shao, L.; and Gao, X. 2019a. Linear kernel tests via empirical likelihood for high-dimensional data. In *AAAI*.

Ding, L.; Yu, M.; Liu, L.; Zhu, F.; Liu, Y.; Li, Y.; and Shao, L. 2019b. Two generator game: Learning to sample via linear goodness-of-fit test. *In NeurIPS*.

Elazar, Y.; Kassner, N.; Ravfogel, S.; Ravichander, A.; Hovy, E.; Schütze, H.; and Goldberg, Y. 2021. Measuring and improving consistency in pretrained language models. *Transactions of the Association for Computational Linguistics*, 9.

Feng, Z.; Xue, R.; Yuan, L.; Yu, Y.; Ding, N.; Liu, M.; Gao, B.; Sun, J.; Zheng, X.; and Wang, G. 2025. Multi-agent Embodied AI: Advances and Future Directions. *arXiv:2505.05108*.

Fu, J.; Ding, L.; Li, H.; Li, P.; Wei, Q.; and Chen, X. 2025. Unveiling and Causalizing CoT: A Causal Pespective. *arXiv:2502.18239*.

Fu, J.; Gao, R.; Yu, Y.; Wu, J.; Li, J.; and Liu, D. 2024. Contrastive graph learning long and short-term interests for POI recommendation. *Expert Systems with Applications*, 238.

Gupta, T.; Gong, W.; Ma, C.; Pawlowski, N.; Hilmkil, A.; Scetbon, M.; Rigter, M.; Famoti, A.; Llorens, A. J.; Gao, J.; Bauer, S.; Kragic, D.; Schölkopf, B.; and Zhang, C. 2024. The Essential Role of Causality in Foundation World Models for Embodied AI. *arXiv:2402.06665*.

Hafner, D. 2022. Benchmarking the Spectrum of Agent Capabilities. *arXiv:2109.06780*.

Hessel, M.; Modayil, J.; Van Hasselt, H.; Schaul, T.; Ostrovski, G.; Dabney, W.; Horgan, D.; Piot, B.; Azar, M.; and Silver, D. 2018. Rainbow: Combining improvements in deep reinforcement learning. In *AAAI*.

Kaddour, J.; Lynch, A.; Liu, Q.; Kusner, M. J.; and Silva, R. 2022. Causal machine learning: A survey and open problems. *arXiv:2206.15475*.

Karpas, E.; Abend, O.; Belinkov, Y.; Lenz, B.; Lieber, O.; Ratner, N.; Shoham, Y.; Bata, H.; Levine, Y.; Leyton-Brown, K.; Muhlgay, D.; Rozen, N.; Schwartz, E.; Shachaf, G.; Shalev-Shwartz, S.; Shashua, A.; and Tenenholtz, M. 2022. MRKL Systems: A modular, neuro-symbolic architecture that combines large language models, external knowledge sources and discrete reasoning. *arXiv:2205.00445*.

Kauvar, I.; Doyle, C.; Zhou, L.; and Haber, N. 2023. Curious Replay for Model-Based Adaptation. *In ICML*.

Liu, S.; Chen, C.; Qu, X.; Tang, K.; and Ong, Y.-S. 2024. Large language models as evolutionary optimizers. In *IEEE Congress on Evolutionary Computation*.

Luo, J.; Dong, P.; Zhai, Y.; Ma, Y.; and Levine, S. 2024. RLIF: Interactive Imitation Learning as Reinforcement Learning. In *ICLR*.

Ma, Y.; Frauen, D.; Javurek, E.; and Feuerriegel, S. 2025. Foundation Models for Causal Inference via Prior-Data Fitted Networks. *arXiv:2506.10914*.

Moon, S.; Yeom, J.; Park, B.; and Song, H. O. 2024. Discovering hierarchical achievements in reinforcement learning via contrastive learning. *In NeurIPS*.

OpenAI. 2024a. GPT-4o System Card. *arXiv:2410.21276*.

OpenAI. 2024b. OpenAI o1 System Card. *arXiv:2412.16720*.

Pearl, J. 2009. *Causality*. Cambridge university press.

Qian, H.; Bai, C.; Zhang, J.; Wu, F.; Song, W.; and Li, X. 2025. Discriminator-Guided Embodied Planning for LLM Agent. In *ICLR*.

Saikh, T.; Ghosal, T.; Mittal, A.; Ekbal, A.; and Bhattacharyya, P. 2022. Scienceqa: A novel resource for question answering on scholarly articles. *International Journal on Digital Libraries*, 23.

Sheth, A.; Roy, K.; and Gaur, M. 2023. Neurosymbolic AI – Why, What, and How. *arXiv:2305.00813*.

Stanić, A.; Tang, Y.; Ha, D.; and Schmidhuber, J. 2022. Learning to Generalize with Object-centric Agents in the Open World Survival Game Crafter.

Tan, H.; Zhang, Z.; Ma, C.; Chen, X.; Dai, Q.; and Dong, Z. 2025. MemBench: Towards More Comprehensive Evaluation on the Memory of LLM-based Agents. *arXiv:2506.21605*.

Team, Q. 2024. Qwen2.5: A Party of Foundation Models.

Touvron, H.; Lavril, T.; Izacard, G.; Martinet, X.; Lachaux, M.-A.; Lacroix, T.; Rozière, B.; Goyal, N.; Hambro, E.; Azhar, F.; Rodriguez, A.; Joulin, A.; Grave, E.; and Lample, G. 2023. LLaMA: Open and Efficient Foundation Language Models. *arXiv:2302.13971*.

Wu, Y.; Min, S. Y.; Prabhumoye, S.; Bisk, Y.; Salakhutdinov, R. R.; Azaria, A.; Mitchell, T. M.; and Li, Y. 2023a. Spring: Studying papers and reasoning to play games. *In NeurIPS*.

Wu, Z.; Wang, Z.; Xu, X.; Lu, J.; and Yan, H. 2023b. Embodied Task Planning with Large Language Models. *arXiv:2305.03716*.

Zhang, D.; Zhoubian, S.; Hu, Z.; Yue, Y.; Dong, Y.; and Tang, J. 2024a. Rest-mcts*: Llm self-training via process reward guided tree search. *In NeurIPS*.

Zhang, W.; and Lu, Z. 2024. Adarefiner: Refining decisions of language models with adaptive feedback. *In Findings of NAACL*.

Zhang, Y.; Zhang, Y.; Gan, Y.; Yao, L.; and Wang, C. 2024b. Causal graph discovery with retrieval-augmented generation based large language models. *arXiv:2402.15301*.