CausalPlan: Empowering Efficient LLM Multi-Agent Coordination Through Causality-Driven Planning

Anonymous authors
Paper under double-blind review

Abstract

Large language model (LLM) agents often generate causally invalid plans in multi-agent coordination tasks due to reliance on spurious statistical correlations rather than grounded causal reasoning, leading to poor task performance. We propose CausalPlan, a framework that enforces causal consistency in LLM planning by embedding learned structural knowledge directly into the decoding process. CausalPlan first extracts a Structural Causal Action (SCA) model, which learns a policy-level causal graph from agent trajectories to capture how prior actions and current environment states influence future decisions. The learned SCA guides planning by reweighting candidate action tokens during generation and providing grounded alternatives when causal violations are detected. By embedding causal knowledge, CausalPlan constrains planning to causal-consistent behaviors under the learned causal model without requiring fine-tuning. We evaluated CausalPlan on the Overcooked-AI benchmark across five multi-agent tasks and four LLMs: Gemma-7B, Llama-8B, Qwen-14B and Llama-70B. Experimental results show that CausalPlan consistently reduces causally invalid actions and improves task completion in both AI-AI and human-AI collaboration settings, outperforming strong LLMs and reinforcement learning baselines. Our findings demonstrate that causality-driven planning is essential for deploying efficient, interpretable, and robust multi-agent systems.

1 Introduction

Large Language Models (LLMs) have evolved from static text processors into dynamic decision-makers, showing immense promise in multi-agent frameworks (Zhang et al., 2023a; Qian et al., 2024; Zhang et al., 2024a). A critical frontier in this domain is *zero-shot multi-agent coordination*—the ability to collaborate with unseen partners, including humans, without prior joint training (Legg & Hutter, 2007; Hu et al., 2020). Compared to traditional multi-agent reinforcement learning (RL) methods, LLM-based agents, trained on vast and diverse datasets that contain rich common-sense knowledge and linguistic abstractions, have demonstrated impressive performance and emerged as a promising solution to this challenge (Zhang et al., 2024a).

Despite these strengths, a persistent limitation remains: LLM agents often lack robust causal reasoning capabilities (Joshi et al., 2024; Chi et al., 2024), which leads to causally invalid actions during planning. We formally categorize these invalid actions into three failure modes: physical constraint violations (actions infeasible in the current state), temporal dependency errors (actions executed out of causal sequence), and coordination misalignments (conflicting or redundant actions relative to a partner). As shown in Fig. 1(a), even a large-scale model like Llama-70B produces a substantial number of invalid actions, all of which fall into one of these three categories. Existing attempts to address this shortcoming are largely confined to single-agent settings and either rely on the LLM itself to infer causal structure via prompting or inject externally specified causal graphs into the planning prompt (Yu & Lu, 2025; Chen et al., 2025; Chai et al., 2025). As a result, causal reasoning remains entangled with language generation and is constrained by the LLM's causal inference capabilities, which vary widely between model sizes, architectures, and prompting strategies, while offering limited support for inter-agent dependencies and coordination dynamics. To address these limitations, we propose CausalPlan, a framework inspired by Pearl's theory of causality and Structural

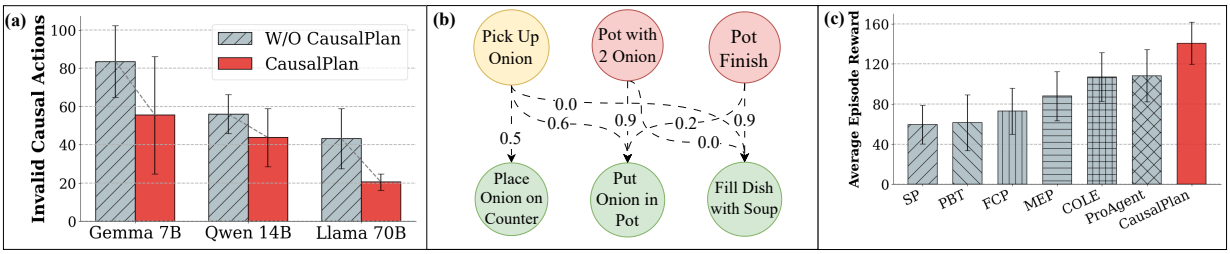

Figure 1: **(a)** Evaluation on the Overcooked Cramped Room layout showing the number of causally invalid actions (out of 400 timesteps), averaged over five seeds. CausalPlan significantly reduces the number of invalid moves. **(b)** Example of causal relationships discovered by CausalPlan on Cramped Room layout. Yellow and red nodes indicate parent actions and states, respectively, while green nodes denote child actions. The edge weights represent the learned probability of a causal influence existing between a source node (state or action) and a target action. "Pick Up Onion" strongly influences "Put Onion in Pot" (0.6) and "Place Onion on Counter" (0.5), but not "Fill Dish with Soup" (0). The state "Pot with 2 Onions" strongly drives "Put Onion in Pot" (0.9), while "Pot Finished" strongly influences "Fill Dish with Soup". **(c)** Mean and variance reward for CausalPlan and baselines when paired with five different unseen policies trained on human trajectories (over 400 timesteps). CausalPlan consistently outperforms all other baselines. Detailed results in Appx. Fig. 8.

Causal Model (SCM) (Pearl, 2009) that goes beyond prompt-level guidance by embedding structural causal knowledge directly into the LLM decoding process.

CausalPlan operates in two phases (see Fig. 2 for an illustration): *Causal Action Structure Learning* and *Agent Planning with Causal Knowledge*. In the first phase, we learn the Structural Causal Action (SCA) model, an extension of SCM, which captures policy-level causal dependencies in the agent's decision-making process. Rather than modeling environment dynamics, the SCA explicitly models how an agent's past actions and the current joint state of all agents causally influence future actions. Under this formulation, the SCA captures all three failure modes by modeling distinct dependency types: physical constraints are enforced by the dependency on the agent's current state (e.g., current location and held items), coordination misalignments are addressed through the dependency on the joint state of all agents to ensure non-redundant behavior, and temporal dependencies are encoded through the causal relationship with past actions in the sequence. Once learned, the SCA produces a Causal Action Matrix $\mathcal{M}$, which encodes causal relationships as causal scores (see Fig. 1(b)). In the second phase, CausalPlan aligns LLM action generation with causal scores in $\mathcal{M}$ through a causal-aware reweighting and resampling mechanism. To ensure compatibility with black-box LLM architectures, we treat the LLM as a proposal distribution, generating a set of candidate actions and computing their model-assigned joint probabilities via token log-likelihoods. The *Causal-Aware Planning* module then reweighted these actions according to their causal score under $\mathcal{M}$. Actions that respect the natural order of cause and effect receive higher weight, while causally inconsistent actions are down-weighted. These reweighted scores are then normalized over the candidate set to form a categorical distribution, from which the final action is sampled. This mechanism specifically targets temporal dependency errors and coordination misalignments by penalizing actions that deviate from the learned causal sequence or conflict with partner dynamics. in scenarios where the LLM's proposal set consists entirely of invalid actions—often due to physical constraint violations—the *Causal Backup Plan* takes over. It selects the action with the highest causal validity regardless of the LLM's original probability, preventing deadlocks and infeasible plans. By embedding causal constraints directly into decoding, CausalPlan ensures that generated actions remain causal-consistent under the learned causal model.

We evaluate CausalPlan on the Overcooked-AI benchmark using four open-source LLMs. While CausalPlan consistently outperforms baselines across both AI-AI (see Fig. 3 and Tab. 1) and human-AI settings, we place particular importance on the human-AI results (see Fig. 1(c)), as they demonstrate the model's superior ability to coordinate with diverse and unpredictable human-proxy behaviors. Empirical results also show that CausalPlan consistently reduces invalid actions. Our contributions are threefold: **(i)** We identify causal

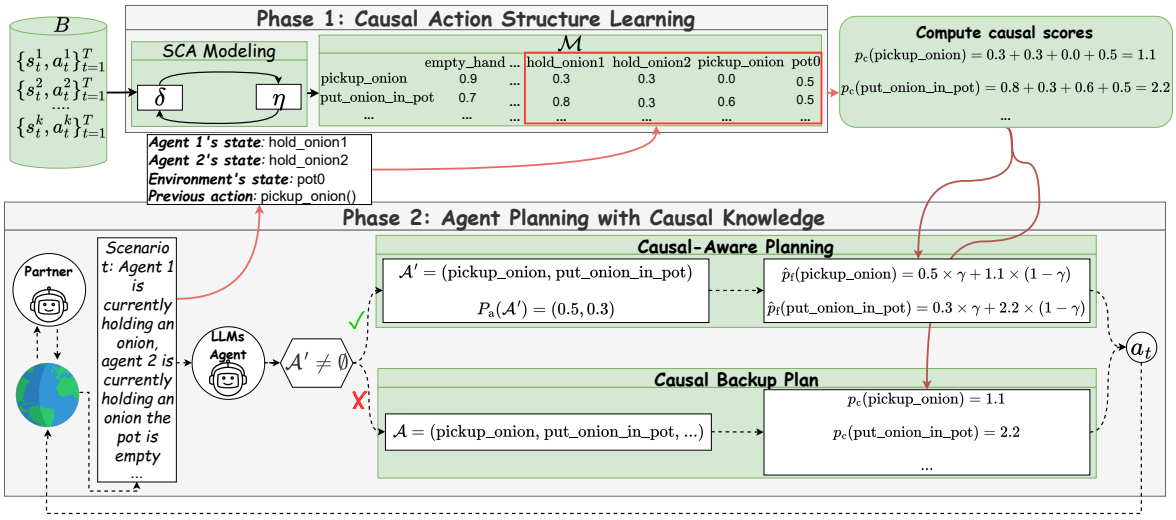

Figure 2: **Overview of the CausalPlan Framework**. The process begins with a dataset B collected by a behavior policy π_β. In Phase 1, we train the SCA Model by optimizing generating (δ) and structural (η) parameters, yielding the *Causal Action Matrix* $\mathcal{M}$, which encodes causal influence from states and past actions to future actions. In Phase 2, an LLM receives scenario t and proposes candidate actions $\mathcal{A}'$. If $\mathcal{A}' \neq \emptyset$, *Causal-Aware Planning* adjusts LLM probabilities; if $\mathcal{A}' = \emptyset$, *Causal Backup Plan* selects the most probable past action via $\mathcal{M}$. Black solid arrows denote causal training; dashed arrows denote LLM inference, and red arrows denote causal knowledge consultation. The red box represents the causal score extraction for each potential next action, where the score is computed as the sum of causal contributions from the current state and previous action.

invalidity as a core failure mode of LLM agents in zero-shot multi-agent coordination and propose causally aligned planning as a remedy; **(ii)** We introduce CausalPlan, the first framework to integrate learned policy-level causal structure directly into LLM decoding for multi-agent planning; **(iii)** We demonstrate through extensive experiments that causality-driven decoding substantially improves coordination performance across model scales and scenarios.

2 Preliminaries

Markov Decision Process. A two-player Markov Decision Process (MDP) is defined as $(\mathcal{S}, \{\mathcal{A}^i\}, P, \gamma, R)$, where $\mathcal{S}$ is the state space, $\mathcal{A}^i$ is the action set for agent $i \in \{1, 2\}$, P defines the transition dynamics, $\gamma \in [0, 1)$ is the discount factor, and $R : \mathcal{S} \times \mathcal{A} \mapsto \mathbb{R}$ is the reward function where $\mathcal{A} = \mathcal{A}_1 \times \mathcal{A}_2$ is the joint action space and $\mathcal{S}$ is the state space. Let S and A denote the dimensions of $\mathcal{S}$ and $\mathcal{A}$, respectively. At each timestep t, each agent $i \in \{1, 2\}$ observes the current state s_t and selects an action according to its policy $\pi^i(a_t^i \mid s_t)$, forming the joint action $a_t = (a_t^1, a_t^2)$. A trajectory is given by $\tau = (s_1, a_1, s_2, a_2, \dots)$, and the objective is to maximize the cumulative expected reward $\mathbb{E}\left[\sum_t R(s_t, a_t)\right]$. In our two-agent setting, one of the agents is the controlled agent (an LLM-based agent), while the other serves as its partner.

Causality and Structural Causal Model. Causality studies the relationships between variables and events (Pearl, 2009). The SCM framework represents causal relationships in a system, where for a set of variables $V = \{V_1, \dots, V_M\}$, each variable V_i is defined as $V_i := f_i(\mathrm{Pa}_\mathcal{G}(V_i), \varepsilon_i)$, with $\{f_1, f_2, \dots, f_M\}$ being generating functions, $\mathrm{Pa}_\mathcal{G}(V_i)$ the parents of V_i in the causal graph $\mathcal{G}$, and $\{\varepsilon_1, \dots, \varepsilon_M\}$ noise terms (Pearl, 2009). The directed acyclic graph (DAG) causal $\mathcal{G} = \{V, E\}$ contains edges $e_{ji} \in E$, where $e_{ji} = 1$ indicates that V_j causes V_i, and $e_{ji} = 0$ otherwise.

Structural Discovery from Interventions (SDI). Causal discovery (Pearl, 2009; Peters et al., 2017) is a fundamental process to infer causal relationship from data. While physical interventional data is typically

required to recover causal dependencies, the SDI framework (Ke et al., 2019) facilitates causal discovery from trajectories by simulating interventions through a structural masking mechanism. SDI operationalizes this by joint-learning the functional parameters θ of each f_i and a structural parameter matrix $\eta \in \mathbb{R}^{M\times M}$ of causal graph $\mathcal{G}$. The framework assume causal sufficiency, faithfulness, and acyclicity to ensure that the learned structure $\mathcal{G}$ reflects true causal dependencies rather than spurious correlations. Here, the existence of an edge e_{ji} is modeled with a probability defined by logistic sigmoid function, $\sigma(\eta_{ji})$, which maps real-valued parameters to the $(0, 1)$ range. By randomly masking inputs to the functional mechanism f_i, the framework simulates a *do*-intervention, which breaks the causal dependency of a variable on its parents to test for necessity. By "masking out" a potential parent, the framework effectively intervenes on the graph structure to test the causal necessity of that variable for inferring the child variable. If the absence of a masked variable significantly decreases the likelihood of the observed data, the gradient-based update increases the structural belief η_{ji}. This framework has been extensively applied to RL research to recover causal models (Peng et al., 2022; Zhang et al., 2023b).

3 Method

In this Sect. 3, we present the proposed CausalPlan framework. Sect. 3.1 introduces the causal modeling formulation, including the SCA model, which is derived via SDI technique, and the Causal Action Matrix $\mathcal{M}$ (see Sect. A.1 for algorithmic details). Sect. 3.2 then describes how this causal knowledge is incorporated into LLM planning through a causal-aware action selection mechanism (see Sect. A.2 for algorithmic details).

3.1 Causal Action Structure Learning

The first phase focuses on constructing an SCA model to estimate a sparse structural dependency graph $\mathcal{G}$, which serves as the foundation for plan refinement. We define the preceding action a_{t-1} and the current state s_t as parent nodes, with the subsequent action a_t as the target child node. Unlike traditional causal world models that focus on reward dynamics or state transitions (Zhang et al., 2023b), our approach explicitly treats the next action as the effect. By modeling a_t as the child node, the recovered graph $\mathcal{G}$ is intended to capture three critical structural constraints: *physical affordances* ($s_t \to a_t$), which capture environmental preconditions like having an empty hand to "pick up" an item; *temporal logic* ($a_{t-1} \to a_t$), which enforces sequential dependencies such as picking up an onion before placing it in a pot; and *coordination alignment*, which synchronizes agent behaviors through dependencies on the shared environment state. This structure ensures that both agents' plans remain causally grounded and mutually compatible.

Data Collection. To facilitate the process of SCA modeling using SDI technique, we collect a dataset $B = \left\{\{(s_t^k, a_t^k)\}_{t=1}^{T}\right\}_{k=1}^{N}$ containing actions that have been executed successfully in the environment. By focusing only on successful trajectories, the SCA model captures the necessary conditions for action feasibility. These trajectories are generated using a behavior policy π_β, which may take the form of an exploratory random policy (e.g., an LLM sampling actions) or a specialized pretrained expert model. In a standard causal sense, the behavior policy provides the necessary variance to test causal hypotheses. While a random policy provides broad coverage of the state-action space, an expert policy consistently selects actions only when their causal prerequisites are met.

SCA Modeling. We define the SCA model as a set of structural equations where each action $a_{t,i}$ is:

$$a_{t,i} := f_i\left(\mathrm{Pa}_{\mathcal{G}}(a_{t,i}), \varepsilon_i\right), \tag{1}$$

where $\mathrm{Pa}_{\mathcal{G}}(a_{t,i}) \subseteq \{s_t, a_{t-1}\}$ represents the causal parents: the specific subset of state features and preceding actions that directly influence the feasibility of $a_{t,i}$. While the generating function f_i is parameterized by δ_i, the structural dependencies are governed by latent edge logits $\eta_{ji} \in \mathbb{R}$, whose sigmoid-transformed values define continuous edge probabilities $p_{ji} = \sigma(\eta_{ji}) \in (0, 1)$ where σ is the sigmoid function. The parent set is effectively realized through a differentiable masking mechanism:

$$\tilde{x}_{t,i} = p_{\cdot,i} \odot [s_t, a_{t-1}], \tag{2}$$

where $\odot$ denotes element-wise multiplication and $\tilde{x}_{t,i}$ is the differentiably masked input used during training. This masking functions as a structural intervention during the likelihood calculation. By selectively hiding or revealing features to the generating function f_i, the model is forced to determine which inputs are strictly necessary to reconstruct the successful action. This ensures that the function f_i only receives information from features deemed causally relevant. We recover the graph structure by optimizing the joint loss: $L(\delta, \eta) = L_{\text{causal}}(\delta, \eta) + L_{\text{reg}}(\eta)$, where L_{causal} quantifies the reconstruction accuracy under these constraints:

$$L_{\text{causal}} = \mathbb{E}_{(a_{t-1}, s_t, a_t) \sim B} \Big[- \sum_{i=1}^{A} \log P\big(a_{t,i} \mid s_t, a_{t-1};\, \delta, \eta\big) \Big]. \tag{3}$$

We deliberately model action components independently to ensure that our structural interventions affect the likelihood of each action component in isolation. By training on successful trajectories, L_{causal} rewards the inclusion of an edge ($\eta_{ji} \to 1$) only if that specific feature is necessary to maintain a high likelihood for the executed action. If an action's success can be explained without a certain feature, the masking mechanism encourages its removal to minimize complexity. To prevent the discovery of spurious correlations, a negative-log-prior penalty is included:

$$L_{\text{reg}} = \lambda \sum_{i,j} p_{ji} \left(- \log P(e_{ji} = 1) \right), \tag{4}$$

where $\lambda > 0$ controls the sparsity of the graph. Assigning a high edge probability p_{ji} incurs a proportional sparsity cost $-\log P(e_{ji} = 1)$, ensuring that only edges with strong empirical evidence from the interventional data are retained. This decoupling ensures that the structural parameters η do not merely imitate the behavior policy's preferences, but instead converge on the true physical affordances and sequential logic.

Causal Action Matrix construction. We then construct a causal action matrix $\mathcal{M} \in \mathbb{R}^{A \times (S+A)}$ that encodes the causal score of selecting each action given the current state and past actions. Each row of the matrix corresponds to a possible next action, and each column corresponds to a state or past action feature. Each entry (i, j) stores the learned edge probability p_{ji}. A query $\mathcal{M}(s_t, a_{t-1}, a)$ returns the causal score $p_{\text{c}}(a) = \sum_{j \in J} p_{ji}$, where $J = \text{Active}(s_t, a_{t-1}) \subseteq \{1, \ldots, S+A\}$ denote the set of column indices corresponding to features that are "active" in the current state s_t and the previous action a_{t-1}, and i is the row index corresponding to action a (details refer to Appx. A.2.2). To reduce trivial pairwise cycles, we compare mutually opposing edge probabilities and retain only the stronger direction. This heuristic enforces local antisymmetry, though it does not guarantee global acyclicity in the general case.

3.2 Agent Planning with Causal Knowledge

At each decision step t, rather than directly generating an action a_t from the historical trajectory $h_t = (s_1, a_1, \ldots, a_{t-1}, s_t)$, the agent must evaluate candidate actions against the structural constraints encoded in the causal matrix $\mathcal{M}$. To leverage the reasoning capabilities of the LLM while adhering to the learned structural constraints, we adopt a dual-prompt strategy, similar to Chain-of-Thought (CoT) (Wei et al., 2022). We first provide the current observation s_t to the LLM agent and prompt it to analyze the scene. Both the observation s_t and the analysis are then used as inputs for a second prompt, where the agent is asked to generate a set of candidate actions $\mathcal{A}' = \left\{ a'_1, a'_2, \ldots, a'_{|\mathcal{A}'|} \right\} \subseteq \mathcal{A}$. Each candidate is associated with a sampling probability $p_{\text{a}}(a'_m)$ derived from the LLM. This probability is calculated as the joint probability of the token sequence:

$$p_{\text{a}}(a'_m) = \exp\left(\sum_{i=1}^{N} \log P(w_i \mid w_{<i}) \right), \tag{5}$$

where each $\log P(w_i \mid w_{<i})$ is the log-probability of the i-th token conditioned on the prompt and preceding tokens. This ensures that $p_{\text{a}}(a'_m)$ reflects the model's internal heuristic confidence in the generated action

string. We then pass these candidates through a verifier to ensure they comply with physical constraints. If $\mathcal{A}' \neq \emptyset$, the Causal-Aware Planning module (Sect. 3.2.1) selects the most suitable action based on $\mathcal{M}$. If no valid candidates are generated ($\mathcal{A}' = \emptyset$), the system invokes the Causal Backup Plan (Sect. 3.2.2).

3.2.1 Causal-Aware Planning

To integrate learned environmental logic into the planning process, we extract a causal score for each candidate action $a' \in \mathcal{A}'$ from our structural model: $p_{\mathrm{c}}(a') = \mathcal{M}(s_t, a_{t-1}, a'), \forall a' \in \mathcal{A}'$. This forms the set $P_{\mathrm{c}}(\mathcal{A}') = \left\{p_{\mathrm{c}}(a'_1), p_{\mathrm{c}}(a'_2), \ldots, p_{\mathrm{c}}(a'_{|\mathcal{A}'|})\right\}$ (details in Appx. A.2.2), representing the "causal likelihood" of each candidate given the current state and previous action. We then compute a joint probability $\hat{p}_{\mathrm{f}}(a'_m)$ as a weighted sum of the LLM's heuristic preference and the model's causal score:

$$\hat{p}_{\mathrm{f}}(a'_m) = \gamma \cdot p_{\mathrm{a}}(a'_m) + (1-\gamma) \cdot p_{\mathrm{c}}(a'_m), \tag{6}$$

where $\gamma \in [0, 1]$ is a hyperparameter balancing LLM intuition with structural rigor. After applying a softmax function to normalize the distribution into:

$$\begin{aligned} P_{\mathrm{f}}(\mathcal{A}') &= \{p_{\mathrm{f}}(a'_1), p_{\mathrm{f}}(a'_2), \ldots, p_{\mathrm{f}}(a'_{|\mathcal{A}'|})\}, \\ &\text{subject to } \sum_{m=1}^{|\mathcal{A}'|} p_{\mathrm{f}}(a'_m) = 1. \end{aligned} \tag{7}$$

We identify and merge any semantically redundant actions by summing their probabilities (details in Appx. A.2.4). The final action a_t is then sampled from the reduced set $\mathcal{A}'^*$ and the categorical distribution P^*_{f}:

$$a_t \sim \text{Categorical}\left(p^*_{\mathrm{f}}(a'_1), p^*_{\mathrm{f}}(a'_2), \ldots, p^*_{\mathrm{f}}(a'_{|\mathcal{A}'^*|})\right). \tag{8}$$

3.2.2 Causal Backup Plan

When the LLM fails to generate valid candidates ($\mathcal{A}' = \emptyset$), traditional agents often fall into "hallucination loops" where they persistently suggest invalid actions despite re-planning (Zhang et al., 2024a). Inspired by human behavior under uncertainty—where one reverts to highly familiar, reliable patterns—we propose a recovery mechanism grounded in the SCA model. Instead of an immediate re-plan, the agent retrieves the causal scores for all possible actions in the environment: $p_{\mathrm{c}}(a) = \mathcal{M}(s_t, a_{t-1}, a), \forall a \in \mathcal{A}$, (details in Appx. A.2.2). This yields a probability distribution $P_{\mathrm{c}}(\mathcal{A})$ over the entire action space. We then greedily select the action with the highest causal reliability:

$$a_t = \arg\max_{a \in \mathcal{A}} P_{\mathrm{c}}(a), \tag{9}$$

By bypassing the LLM's current reasoning state and relying on the learned structural matrix, the agent can move out of stagnant states. Only if this high-reliability action fails to trigger a state change do we prompt the LLM to re-plan.

4 Experiments

4.1 Experimental setup

We employ the Overcooked-AI environment suite (Carroll et al., 2019) as our primary testing platform. This suite consists of five distinct layouts: Cramped Room (CR), Asymmetric Advantages (AA), Coordination

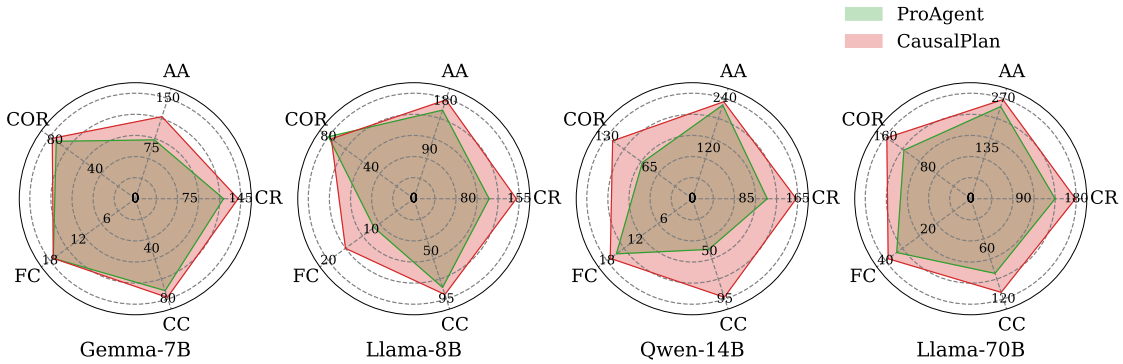

Figure 3: Performance of different LLMs with and without CausalPlan across various layouts. In these experiments, we use the LLM agent as Player 1, allowing it to collaborate with all other baselines for 400 timesteps and report the average episode reward.

Ring (COR), Forced Coordination (FC), and Counter Circuit (CC) (see Appx. B.2 for visual specifications). Each layout evaluates unique dimensions of multi-agent coordination, ranging from basic spatial navigation to complex sequential task dependencies. We evaluate our framework across several model scales, including `Gemma-7B`, `Llama-3-8B`, `Qwen-14B`, and `Llama-3.3-70B`. These models are integrated into the ProAgent framework (Zhang et al., 2024a). We apply CausalPlan to refine the actions generated by these agents. Additionally, we utilize `Command-R` (35B) to generate the structured observation analysis within our dual-prompt pipeline (refer to Appx. A.2.1).

To rigorously evaluate CausalPlan, we structure our analysis around several key questions: **Q1: Does CausalPlan enhance the coordination of LLM agents?** In Sect. 4.2, we compare base ProAgent models against their CausalPlan-enhanced counterparts, collaborating with diverse RL partners. **Q2: How does CausalPlan perform against traditional RL?** We benchmark our Llama-70B backbone directly against standard RL agents in bidirectional roles (Sect. 4.2). **Q3: Human-AI Collaboration. Can CausalPlan effectively collaborate with human-like partners?** Sect. 4.3 evaluates coordination with Behavior Cloning agents (Li et al., 2023b) that mimic human sub-optimality. **Q4: Does CausalPlan transcend simple imitation of the behavior policy?** In Sect. 4.4, we compare our framework against non-causal models—trained on the dataset collected from the same π_β—to determine if structural logic allows the agent to outperform suboptimal training patterns. **Q5: What is the impact of each CausalPlan component?** Sect. 4.6 quantifies the performance gains of individual modules. **Q6: How does causal knowledge prevent planning failures?** Sect. 4.7 investigates how the learned causal matrix prevents specific causal errors. **Q7: Can a causal graph learned in one setting be reused in another?** Section 4.5 examines the transferability of learned causal graphs across different environments. In the Appendix, we provide additional experiments such as parameter tuning γ (Appx. B.6.1), using different policies π_β (Appx. B.6.2), time complexity analysis (Appx. B.9), comparison to Causal-aware LLMs (Chen et al., 2025) (Appx. B.10), the causal action matrix $\mathcal{M}$ (Appx. B.8) and a verification of the causal graph found (Appx. C).

Baselines. Our evaluation includes two primary categories of baselines. First, we compare against **ProAgent** (Zhang et al., 2023a), which utilizes **ReAct** (Yao et al., 2023) and **Reflexion** (Shinn et al., 2023) prompting, to measure the specific gain from our causal modeling. Second, we evaluate against established RL methods for zero-shot human-AI coordination, including **SP** (Tesauro, 1994), **PBT** (Jaderberg et al., 2017), **FCP** (Strouse et al., 2021), **MEP** (Zhao et al., 2023), and **COLE** (Li et al., 2023b) (see Appx. B.3 for details). Furthermore, we assess the generalizability of CausalPlan in the Crafter environment (Hafner, 2021), a challenging long-horizon planning benchmark. In this single-agent setting, CausalPlan outperforms state-of-the-art causal prompting methods **Causal-aware LLMs** (Chen et al., 2025) by 14% (detailed results in Appx. B.10).

Table 1: Average performance (mean ± std), collaborating with AI partners, of baseline AI agents and CausalPlan (Ours) across layouts using Llama-70B. Results are averaged returns over both player positions (400 timesteps each). Best and second-best results are in **bold** and underlined, respectively. Detailed performance of playing as Player 0 or Player 1 is provided in Appx. Tab. 7.

Layout	Baseline AI Agents					CausalPlan (Ours)
	SP	PBT	FCP	MEP	COLE	
AA	184.0 ± 17.5	168.0 ± 15.4	176.6 ± 15.0	167.3 ± 5.8	185.3 ± 15.1	**258.7 ± 16.4**
CC	56.7 ± 9.2	52.0 ± 14.0	63.4 ± 10.5	50.0 ± 16.1	90.6 ± 10.1	**112.6 ± 7.6**
CR	162.0 ± 10.0	168.0 ± 5.0	**194.0 ± 10.1**	178.0 ± 16.1	153.4 ± 12.5	172.7 ± 4.2
COR	120.7 ± 11.0	139.4 ± 10.1	130.7 ± 6.2	**160.7 ± 7.2**	153.4 ± 4.6	156.6 ± 3.2
FC	18.0 ± 4.6	40.6 ± 10.3	42.0 ± 7.2	30.4 ± 5.4	44.6 ± 7.0	**53.9 ± 14.9**

4.2 AI partner evaluation

Enhancing open-source LLM performance using CausalPlan We first evaluate whether CausalPlan effectively enhances the collaborative capabilities of open-source LLMs across varying scales. As illustrated in Fig. 3 and detailed in Appx. Tab. 6, CausalPlan leads to consistent performance gains across all evaluated backbones. Notably, we observe significant improvements in Qwen-14B (29.04%) and Llama-70B (22.42%), suggesting that the integration of causal constraints is particularly effective for models with higher reasoning capacities. From a layout perspective, the most substantial boosts were in CR (20.83%) and COR (19.13%). These results demonstrate that CausalPlan effectively mitigates planning failures even in larger models, confirming its potential to enhance agent performance at scale by grounding LLM heuristics in causal logic.

Comparison with state-of-the-art RL baselines. To assess the competitive standing of our approach, we benchmark our top-performing agent (with Llama-70B backbone) against the set of SOTA baseline RL agents. As presented in Tab. 1, CausalPlan consistently ranks among the top performers, securing the highest score in three out of five layouts and the second-highest in one other layout. The most significant performance gaps between our method and the next best baseline are observed in the AA layout, showing a 39% advantage. This layout requires agents to specialize and coordinate based on physical constraints—a task for which causal modeling is ideally suited. Overall, these results highlight that integrating causal reasoning allows open-source LLMs to not only close the gap with specialized RL agents but, in many complex scenarios, significantly surpass them.

4.3 Human partner evaluation

To evaluate the effectiveness of CausalPlan in human-centric scenarios, we conducted a series of experiments using human proxy partners developed through BC to simulate the varied playstyles and potential sub-optimality of human agents. As illustrated in Fig. 1(c), our agent, utilizing Llama-70B, outperforms all established baselines. On average across all layouts, CausalPlan achieves a 30% improvement over the base ProAgent framework and surpasses COLE, the strongest RL baseline, by approximately 31%. This performance leap suggests that grounding the LLM's high-level planning in learned causal constraints allows the agent to navigate unpredictable coordination patterns of human-like partners with fewer logical errors. To verify the reliability of these observed improvements, we performed a rigorous statistical analysis using paired t-tests and corresponding p-values on the results of AI-partner evaluations. The results, detailed in Appendix B.5, demonstrate that CausalPlan consistently yields higher t-values than ProAgent when compared against the top-tier RL methods. A direct comparison reveals that the performance gains are statistically significant at the $p < 0.05$ level in 30% of the tested cases, specifically in the CR-P0, AA-P1, and COR-P1 configurations. An additional 30% of cases exhibit marginal significance with p-values ranging between 0.05 and 0.2. Crucially, the inclusion of the CausalPlan module does not result in performance degradation in any configuration, reinforcing the conclusion that the integration of causal knowledge provides a robust advantage for multi-agent coordination.

Table 2: Performance comparison between the non-causal supervised baselines ($\text{MEP}_{\text{guided}}$ and $\text{MEP}_{\text{backup}}$), and CausalPlan.

Layout	$\textbf{MEP}_{\text{guided}}$	$\textbf{MEP}_{\text{backup}}$	**CausalPlan**
AA-P1	207.3 ± 19.4	257.3 ± 9.2	$\mathbf{266.7 \pm 16.7}$
CC-P1	80.3 ± 9.2	90.7 ± 12.9	$\mathbf{112.0 \pm 6.9}$

4.4 Comparison to non-causal supervised conditional models

To assess whether CausalPlan contributes structural guidance beyond behavioral imitation, we evaluated two non-causal baselines trained on the same MEP data to estimate $P(a_t \mid s_t, a_{t-1})$. These baselines serve as supervised conditional models that reproduce the empirical behavior policy without incorporating causal structure, sparsity, or explicit dependency constraints. As shown in Tab. 2, integrating this conditional model with the Llama-70B backbone—which alone achieves 248.0 on AA-P1 and 89.3 on CC-P1—did not improve coordination performance. Specifically, averaging action probabilities ($\text{MEP}_{\text{guided}}$) led to substantial degradation, with scores dropping by 16.4% and 10.1%, respectively. This suggests that directly coupling the backbone with a non-causal imitation model can anchor decision-making to the limitations of the underlying demonstrations. In settings where the behavior policy π_β is weaker than the backbone model, this imitation bias can become detrimental, as it steers the backbone toward suboptimal actions rather than enabling stronger reasoning. By contrast, using the same model solely as a recovery mechanism ($\text{MEP}_{\text{backup}}$) yielded moderate gains, but remained significantly less effective than CausalPlan. The performance gap highlights a key distinction: whereas $\text{MEP}_{\text{backup}}$ primarily filters infeasible actions, CausalPlan introduces an interpretable sparse dependency structure that captures temporal and contextual relationships among action components. In complex coordination tasks, respecting these dependencies is as important as maintaining physical feasibility.

4.5 Causal graph transferability

To evaluate the transferability of causal graphs in the Overcooked domain, we tested whether a causal graph learned from the `aa` layout could be reused in other layouts when paired with the collaborative agent COLE. The objective was to assess whether CausalPlan captures reusable structural dependencies rather than layout-specific patterns. As shown in Table 3, the transferred causal graph remains competitive across all evaluated layouts. On average, the transferred graph achieves approximately 94% of the native graph's performance, indicating that much of the learned structural knowledge is preserved even when deployed outside its original training environment. While native graphs generally provide the strongest results, the transferred graph demonstrates only moderate degradation in most cases, and even slightly outperforms the native graph in the CC layout. Compared to the no-graph baseline, both native and transferred graphs provide clear performance gains in most settings. In particular, the transferred graph substantially improves results in CC and FC, where it exceeds the no-graph baseline by large margins. Although the gap is narrower in COR, the transferred graph remains competitive, suggesting that the learned causal structure contributes meaningful planning guidance beyond what can be achieved without an explicit graph. These findings suggest that the causal representations learned by CausalPlan are not merely overfitted to a single configuration, but instead encode reusable planning structure that can generalize across related layouts. However, this transferability is expected to hold primarily in environments that share the same underlying factorization or structural dependencies. In other words, successful reuse depends on the target environment preserving the causal relationships captured in the source graph. Although transfer does not fully match the effectiveness of native graphs, the strong retention of performance and consistent advantage over no-graph baselines highlight the practical utility of graph reuse in reducing the need for retraining in structurally aligned environments.

4.6 Impact of CausalPlan components

We investigate the individual contributions of the CausalPlan components to isolate the drivers of performance. The results are summarized in Tab. 4. We first compare the standard single-prompt setup (combining observation analysis and planning) against our dual-prompt architecture. As shown in Tab. 4, performance

Table 3: Transferability of causal graphs in Overcooked using a graph learned from the aa layout with the collaborative agent COLE.

Layout	No Graph	Native	Transfer (aa)	Retention (%)
CC	93.3 ± 80.8	140.0 ± 34.6	146.7 ± 9.4	104.8
COR	106.7 ± 61.1	120.0 ± 69.2	100.0 ± 43.2	83.3
FC	13.3 ± 11.5	30.0 ± 20.0	26.7 ± 18.9	89.0

Table 4: Ablation studies of CausalPlan components using Llama-8B on CR layout. Detailed table is in Appx. Tab. 10

Methods	**Average Results**
1-Prompt (ProAgent)	110.7 ± 12.8
2-Prompt	121.3 ± 2.3
CausalPlan (no CBP)	141.3 ± 12.9
CausalPlan (Full)	**150.7 ± 2.3**

between the two configurations is nearly identical. This marginal difference confirms that the primary performance gains stem from the integration of causal knowledge rather than the prompting structure itself. We then evaluate the impact of the Causal Backup Plan module. CausalPlan without the backup mechanism still outperforms the two-prompt baseline by 27%, but falls 7% short of the full framework. This gap underscores the importance of the backup policy in ensuring robustness; it successfully recovers the agent from scenarios where the LLM fails to generate a valid action.

4.7 Benefits of causal integration

We analyzed the behavior of Llama-8B in the CR layout, where CausalPlan achieves a substantial +36.1% improvement. Our findings reveal two primary mechanisms through which causal integration enhances planning. **(1) Physically/Temporally invalid actions.** As shown in Tab. 5, the baseline agent frequently attempts to pick up onion while its hands are occupied, violating the environment's physical constraints. CausalPlan reduces these invalid actions by 18%, while simultaneously increasing valid interactions by 17%. This shift demonstrates that the framework does not merely prune impossible actions; it actively promotes temporally coherent behavior. **(2) Poor coordination.** Coordination failures are similarly mitigated (see Appx. Tab. 12). In scenarios where the partner already carries an onion and the pot is nearly full, the baseline often selects redundant and conflicting pickup actions. With CausalPlan, such occurrences drop to zero. This elimination suggests that the agent effectively internalizes teammate states.

5 Related Work

Reasoning and planning with LLM agents. The rise of LLMs has enabled applications in both single- and multi-agent settings. The works in a single-agent setting focus on improving reasoning through chain-of-thought prompting (Wei et al., 2022; Kojima et al., 2022), self-consistency (Wang et al., 2022), and problem decomposition (Zhou et al., 2022). LLMs have also been applied to robotic planning (Ahn et al., 2022) and reflection-based learning (Shinn et al., 2023). While earlier works like Zhu et al. (2024) and Qiao et al. (2024) try to improve single-agent planning, recent research has shifted toward multi-agent cooperation. Frameworks like Park et al. (2023) and Li et al. (2023a) utilize manual perception and role-playing to facilitate coordination and communication. ReAd (Zhang et al., 2025) employs advantage-weighted selection to refine agent plans within the action-reward manifold; CaPo (Liu et al., 2025) relies on iterative linguistic discussion for coordination, and ELHPlan (Ling et al., 2025) optimizes efficiency through intention-bound action sequences. CausalPlan distinguishes itself by extracting and enforcing the structural causal dependencies between environmental states and multi-agent actions.

Table 5: Number of picking up onion actions under valid and invalid preconditions. CausalPlan effectively suppresses actions that violate physical constraints while increasing valid calls.

State	Without Graph	With Graph
Occupied Hand (**Invalid**)	14	10
Empty Hand (**Valid**)	20	33

Zero-shot multi-agent coordination. Zero-shot multi-agent coordination aims to train agents that can collaborate with unseen partners, human or AI. A classic method is Self-Play (SP) (Tesauro, 1994; Carroll et al., 2019), where agents train by interacting with themselves. Population-Based Training (PBT) (Jaderberg et al., 2017) promotes learning by diversifying the population of training agents. Recent methods combine SP and PBT to increase diversity, such as FCP (Strouse et al., 2021) and MEP (Zhao et al., 2023). COLE (Li et al., 2023b) shifts focus to strategic policy selection during training, but are often computationally intensive and lack interpretability. Zhang et al. (2024a) demonstrated that LLMs can excel in these tasks by leveraging vast linguistic prior knowledge. Nevertheless, LLMs lack causal reasoning ability (Joshi et al., 2024; Chi et al., 2024). We address this limitation by proposing a causal-aligned planning approach that grounds LLM action selection in environmental constraints.

Causality in decision making. Causality has received increasing attention for improving AI decision-making. In single-agent domains, counterfactual methods are used for data augmentation (Pitis et al., 2020; 2022). Corcoll & Vicente (2020) leverage causality to construct variable hierarchies. Zhang et al. (2023b) redistribute rewards based on causal impact. Seitzer et al. (2021) incorporate causal signals into reward shaping. Peng et al. (2022) learns causal graphs to define hierarchical RL subgoals. More recently, efforts have focused on integrating causality into LLM planning by directly providing the causal graph as part of the LLM prompt (Chen et al., 2025; Yu & Lu, 2025). In multi-agent settings, social influence has been used as causality to promote cooperation (Jaques et al., 2019), while subsequent work employs action influence and redistribution of rewards to encourage coordinated behaviors (Du et al., 2024; Zhang et al., 2024b). CausalMACE (Chai et al., 2025) targets multi-LLM-agent collaboration, prompting an LLM to infer a causal graph from task descriptions and rules, and was designed specifically for Minecraft gameplay. In contrast, CausalPlan does not require the LLM to infer the graph; instead, it learns the causal structure from data and integrates it directly into the decoding process, making it more robust across different models and environments.

6 Conclusion and future works

We introduced CausalPlan, a framework that grounds LLM decoding in causal knowledge to enhance multi-agent cooperation. Our experiments demonstrate that while the SCA model is trained on a single behavior policy, it generalizes effectively to diverse partner agents and various spatial configurations. By integrating learned causal constraints, CausalPlan improves the robustness of LLMs. This work marks a significant step toward making LLM-based agents safer and more interpretable in complex environments. Future research could investigate combining this structural approach with causal prompting or developing layout-agnostic SCA models to further improve the generalizability and efficiency of multi-agent coordination.

7 Broader impact statement

This work represents an important foundational step toward integrating causal reasoning into multi-agent planning with large language models (LLMs). Our causality-driven framework aims to improve the safety, efficiency, and interpretability of collaborative AI systems by enabling agents to better understand the consequences of their states and actions. Although primarily exploratory and not yet intended for real-world deployment, the results demonstrate promising potential for advancing multi-agent coordination.

At this stage, we do not expect any direct negative societal impacts, as the framework requires further development and validation before practical use. Nevertheless, as autonomous multi-agent systems mature,

concerns related to fairness, reliability, misuse, and broader ethical implications will become increasingly important. Addressing these challenges through responsible design, transparency, and rigorous evaluation will be critical to ensure the safe and trustworthy deployment of such systems in the future.

Acknowledgments

Use unnumbered third level headings for the acknowledgments. All acknowledgments, including those to funding agencies, go at the end of the paper. Only add this information once your submission is accepted and deanonymized.

References

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*, 2022.

Micah Carroll, Rohin Shah, Mark K Ho, Thomas L Griffiths, Sanjit A Seshia, Pieter Abbeel, and Anca Dragan. On the utility of learning about humans for human-AI coordination. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 5174–5185, 2019.

Qi Chai, Zhang Zheng, Junlong Ren, Deheng Ye, Zichuan Lin, and Hao Wang. Causalmace: Causality empowered multi-agents in minecraft cooperative tasks. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 14410–14426, 2025.

Wei Chen, Jiahao Zhang, Haipeng Zhu, Boyan Xu, Zhifeng Hao, Keli Zhang, Junjian Ye, and Ruichu Cai. Causal-aware large language models: Enhancing decision-making through learning, adapting and acting. In James Kwok (ed.), *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence, IJCAI*, pp. 4292–4300. International Joint Conferences on Artificial Intelligence Organization, 8 2025. doi: 10.24963/ijcai.2025/478. URL `https://doi.org/10.24963/ijcai.2025/478`. Main Track.

Haoang Chi, He Li, Wenjing Yang, Feng Liu, Long Lan, Xiaoguang Ren, Tongliang Liu, and Bo Han. Unveiling causal reasoning in large language models: Reality or mirage? In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 37, pp. 96640–96670, 2024.

Cohere. The command R model (details and application). `https://docs.cohere.com/v2/docs/command-r`, 2024. Accessed: 2025-05-12.

Oriol Corcoll and Raul Vicente. Disentangling causal effects for hierarchical reinforcement learning. *arXiv preprint arXiv:2010.01351*, 2020.

Xiao Du, Yutong Ye, Pengyu Zhang, Yaning Yang, Mingsong Chen, and Ting Wang. Situation-dependent causal influence-based cooperative multi-agent reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 17362–17370, 2024.

Danijar Hafner. Benchmarking the spectrum of agent capabilities. *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021.

Danijar Hafner, Timothy Lillicrap, Mohammad Norouzi, and Jimmy Ba. Mastering atari with discrete world models. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2020.

Patrik O Hoyer, Dominik Janzing, Joris Mooij, Jonas Peters, and Bernhard Schölkopf. Nonlinear causal discovery with additive noise models. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 689–696, 2008.

Hengyuan Hu, Adam Lerer, Alex Peysakhovich, and Jakob Foerster. "Other-play" for zero-shot coordination. In *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 4399–4410, 2020.

Max Jaderberg, Valentin Dalibard, Simon Osindero, Wojciech M Czarnecki, Jeff Donahue, Ali Razavi, Oriol Vinyals, Tim Green, Iain Dunning, Karen Simonyan, Chrisantha Fernando, and Koray Kavukcuoglu. Population based training of neural networks. *arXiv preprint arXiv:1711.09846*, 2017.

Natasha Jaques, Angeliki Lazaridou, Edward Hughes, Caglar Gulcehre, Pedro Ortega, DJ Strouse, Joel Z Leibo, and Nando De Freitas. Social influence as intrinsic motivation for multi-agent deep reinforcement learning. In *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 3040–3049, 2019.

Nitish Joshi, Abulhair Saparov, Yixin Wang, and He He. Llms are prone to fallacies in causal inference. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 10553–10569, 2024.

Nan Rosemary Ke, Olexa Bilaniuk, Anirudh Goyal, Stefan Bauer, Hugo Larochelle, Bernhard Schölkopf, Michael C Mozer, Chris Pal, and Yoshua Bengio. Learning neural causal models from unknown interventions. *arXiv preprint arXiv:1910.01075*, 2019.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 22199–22213, 2022.

Shane Legg and Marcus Hutter. Universal intelligence: A definition of machine intelligence. *Minds and Machines*, 17:391–444, 2007.

Guohao Li, Hasan Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem. Camel: Communicative agents for "mind" exploration of large language model society. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, pp. 51991–52008. Curran Associates, Inc., 2023a.

Yang Li, Shao Zhang, Jichen Sun, Yali Du, Ying Wen, Xinbing Wang, and Wei Pan. Cooperative open-ended learning framework for zero-shot coordination. In *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 20470–20484, 2023b.

Shaobin Ling, Yun Wang, Chenyou Fan, Tin Lun Lam, and Junjie Hu. Elhplan: Efficient long-horizon task planning for multi-agent collaboration. *arXiv preprint arXiv:2509.24230*, 2025.

Jie Liu, Pan Zhou, Yingjun Du, Ah-Hwee Tan, Cees GM Snoek, Jan-Jakob Sonke, and Efstratios Gavves. Capo: Cooperative plan optimization for efficient embodied multi-agent cooperation. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2025.

Minh Hoang Nguyen, Hung Le, and Svetha Venkatesh. Variable-agnostic causal exploration for reinforcement learning. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 216–232. Springer, 2024.

Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S Bernstein. Generative agents: interactive simulacra of human behavior. In *Proceedings of the Annual ACM Symposium on User Interface Software and Technology*, pp. 1–22, 2023.

Judea Pearl. *Causality*. Cambridge University Press, 2009.

Shaohui Peng, Xing Hu, Rui Zhang, Ke Tang, Jiaming Guo, Qi Yi, Ruizhi Chen, Xishan Zhang, Zidong Du, Ling Li, Qi Guo, and Yunji Chen. Causality-driven hierarchical structure discovery for reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 20064–20076, 2022.

Jonas Peters, Dominik Janzing, and Bernhard Schlkopf. *Elements of Causal Inference: Foundations and Learning Algorithms*. The MIT Press, 2017.

Silviu Pitis, Elliot Creager, and Animesh Garg. Counterfactual data augmentation using locally factored dynamics. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 3976–3990, 2020.

Silviu Pitis, Elliot Creager, Ajay Mandlekar, and Animesh Garg. Mocoda: model-based counterfactual data augmentation. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 18143–18156, 2022.

Chen Qian, Zihao Xie, Yifei Wang, Wei Liu, Yufan Dang, Zhuoyun Du, Weize Chen, Cheng Yang, Zhiyuan Liu, and Maosong Sun. Scaling large-language-model-based multi-agent collaboration. *arXiv preprint arXiv:2406.07155*, 2024.

Shuofei Qiao, Runnan Fang, Ningyu Zhang, Yuqi Zhu, Xiang Chen, Shumin Deng, Yong Jiang, Pengjun Xie, Fei Huang, and Huajun Chen. Agent planning with world knowledge model. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 37, pp. 114843–114871, 2024.

Maximilian Seitzer, Bernhard Schölkopf, and Georg Martius. Causal influence detection for improving efficiency in reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 22905–22918, 2021.

Xinwei Shen, Furui Liu, Hanze Dong, Qing Lian, Zhitang Chen, and Tong Zhang. Disentangled generative causal representation learning. 2020.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 8634–8652, 2023.

Peter Spirtes, Clark Glymour, and Richard Scheines. *Causation, Prediction, and Search*. The MIT Press, 2000.

DJ Strouse, Kevin R McKee, Matt Botvinick, Edward Hughes, and Richard Everett. Collaborating with humans without human data. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 14502–14515, 2021.

Gerald Tesauro. TD-Gammon, a self-teaching backgammon program, achieves master-level play. *Neural Computation*, 6(2):215–219, 1994.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2022.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H Chi, Quoc V Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 24824–24837, 2022.

Christopher KI Williams and Carl Edward Rasmussen. *Gaussian Processes for Machine Learning*, volume 2. MIT Press Cambridge, MA, 2006.

Mengyue Yang, Furui Liu, Zhitang Chen, Xinwei Shen, Jianye Hao, and Jun Wang. Causalvae: Disentangled representation learning via neural structural causal models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9593–9602, 2021.

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

Shu Yu and Chaochao Lu. Adam: An embodied causal agent in open-world environments.". In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2025.

Ceyao Zhang, Kaijie Yang, Siyi Hu, Zihao Wang, Guanghe Li, Yihang Sun, Cheng Zhang, Zhaowei Zhang, Anji Liu, Song-Chun Zhu, et al. Proagent: building proactive cooperative agents with large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 17591–17599, 2024a.

Hongxin Zhang, Weihua Du, Jiaming Shan, Qinhong Zhou, Yilun Du, Joshua B Tenenbaum, Tianmin Shu, and Chuang Gan. Building cooperative embodied agents modularly with large language models. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2023a.

Yang Zhang, Shixin Yang, Chenjia Bai, Fei Wu, Xiu Li, Zhen Wang, and Xuelong Li. Towards efficient llm grounding for embodied multi-agent collaboration. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 1663–1699, 2025.

Yudi Zhang, Yali Du, Biwei Huang, Ziyan Wang, Jun Wang, Meng Fang, and Mykola Pechenizkiy. Interpretable reward redistribution in reinforcement learning: a causal approach. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 20208–20229, 2023b.

Yudi Zhang, Yali Du, Biwei Huang, Meng Fang, and Mykola Pechenizkiy. A causality-inspired spatial-temporal return decomposition approach for multi-agent reinforcement learning. In *NeurIPS 2024 Causal Representation Learning Workshop*, 2024b.

Rui Zhao, Jinming Song, Yufeng Yuan, Haifeng Hu, Yang Gao, Yi Wu, Zhongqian Sun, and Wei Yang. Maximum entropy population-based training for zero-shot human-ai coordination. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 6145–6153, 2023.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc V Le, et al. Least-to-most prompting enables complex reasoning in large language models. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2022.

Yuqi Zhu, Shuofei Qiao, Yixin Ou, Shumin Deng, Ningyu Zhang, Shiwei Lyu, Yue Shen, Lei Liang, Jinjie Gu, and Huajun Chen. Knowagent: knowledge-augmented planning for LLM-based agents. *arXiv preprint arXiv:2403.03101*, 2024.