Towards Foresighted AI Cooperators with LLM-driven Decision-Time Planning

Yuheng Jing
C^{2}DL, Institute of Automation, Chinese Academy of Sciences
School of Artificial Intelligence, University of Chinese Academy of Sciences
Beijing, China
jingyuheng2022@ia.ac.cn

Kai Li†
C^{2}DL, Institute of Automation, Chinese Academy of Sciences
School of Artificial Intelligence, University of Chinese Academy of Sciences
Beijing, China
kai.li@ia.ac.cn

Bingyun Liu
C^{2}DL, Institute of Automation, Chinese Academy of Sciences
School of Artificial Intelligence, University of Chinese Academy of Sciences
Beijing, China
liubingyun2021@ia.ac.cn

Ziwen Zhang
C^{2}DL, Institute of Automation, Chinese Academy of Sciences
School of Artificial Intelligence, University of Chinese Academy of Sciences
Beijing, China
zhangziwen2024@ia.ac.cn

Zhe Wu
Department of Computer Science and Technology, Tsinghua University
Beijing, China
wu-z24@mails.tsinghua.edu.cn

Yifan Zhang
C^{2}DL, Institute of Automation, Chinese Academy of Sciences
University of Chinese Academy of Science, Nanjing
Beijing, China
yfzhang@nlpr.ia.ac.cn

Junliang Xing
Department of Computer Science and Technology, Tsinghua University
Beijing, China
jlxing@tsinghua.edu.cn

Jian Cheng†
C^{2}DL, Institute of Automation, Chinese Academy of Sciences
School of Future Technology, University of Chinese Academy of Sciences AiRiA
Beijing, China
jian.cheng@ia.ac.cn

ABSTRACT

In multi-agent systems, building agents capable of seamlessly collaborating with unknown partners is a long-standing research goal. Existing approaches primarily generate a diverse population of partners and then train an agent against this population to master various cooperation conventions. However, these approaches are often hindered in two aspects: (1) They heavily rely on task-specific training; (2) Their trained agents lack adaptability at test time. In this paper, we investigate how to leverage *Large Language Models* (LLMs) to build agents capable of foresighted coordination, addressing the challenges faced by existing work. To facilitate structured reasoning mechanisms, we introduce **DTPAgent**, a novel LLM-driven *Decision-Time Planning* (DTP) framework. Within this framework, LLMs, without relying on task-specific training and solely through in-context learning, estimate the partner policy and the transition-reward function to model the full dynamics of the environment. Based on these LLM-driven modelings, **DTPAgent** simulates a range of possible trajectories to dynamically search for the most advantageous policy at each timestep. We demonstrate on the popular benchmark, Overcooked, that **DTPAgent**, built with small-scale LLMs, effectively outperforms various types of baselines when faced with unseen partners. Our **DTPAgent** also exhibits a scalable property that existing agents lack: the ability to translate test-time computation into improved performance.

KEYWORDS

Multi-Agent Coordination; In-Context Learning; Decision-Time Planning

ACM Reference Format:
Yuheng Jing, Kai Li†, Bingyun Liu, Ziwen Zhang, Zhe Wu, Yifan Zhang, Junliang Xing, and Jian Cheng†. 2026. Towards Foresighted AI Cooperators with LLM-driven Decision-Time Planning. In *Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), Paphos, Cyprus, May 25 – 29, 2026*, IFAAMAS, 12 pages. https://doi.org/10.65109/XALP4331

Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), C. Amato, L. Dennis, V. Mascardi, J. Thangarajah (eds.), May 25 – 29, 2026, Paphos, Cyprus. © 2026 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). https://doi.org/10.65109/XALP4331

1 INTRODUCTION

Developing cooperative agents is a long-standing research direction in multi-agent systems, with numerous real-world applications in domains such as robotics [28, 114], autonomous driving [69,

119], and human-AI dialogue [35, 63]. Although many multi-agent *Reinforcement Learning* (RL) algorithms are capable of deriving performant cooperative policies through self-play [66, 80, 118], the derived policies typically lack good generalization ability, as they are unable to cooperate with policies unseen during training [16, 22, 36, 50]. Therefore, significant endeavours have been made to develop agents that seamlessly cooperate with novel partners on complex tasks [18, 21, 44, 55, 59, 60, 67, 83, 98, 99, 117, 127].

To enhance the generalization ability of agents, mainstream approaches focus on generating a diverse population of policies through *Population-Based Training* (PBT) [41, 42] and using various policies within this population as partners to train an agent capable of mastering multiple cooperative conventions [21, 67, 83, 93, 112, 117, 127]. Their motivation is that the more partners an agent encounters during training, the more potential partners it can collaborate with during testing.

However, existing approaches are typically **hindered in two aspects**: (1) *They heavily rely on task-specific training*, which often involves complex learning stages and meticulous design choices. When the deployment environment undergoes any change, the entire training process must be re-executed, making it impossible to guarantee efficiency and generalizability. (2) *Their trained agents lack adaptability at test time*, whose generalization abilities are primarily limited by the quality and diversity of the partners encountered during training. Additionally, fine-tuning these agents (through gradient updates) typically completely corrupts the original policy, especially when only a very limited number of samples can be collected at test time [27, 36, 44, 72].

Recently, *Large Language Models* (LLMs), pretrained on massive-scale data and thereby possessing extensive common knowledge and human priors [1, 64], have demonstrated outstanding abilities in numerous domains such as long conversations [47, 74], reasoning [13, 32], and software engineering [26, 46, 122, 124]. LLM-based agent architectures such as SayCan [11], ReAct [116], DEPS [105], RAP [34], Reflexion [85], JARVIS-1 [106], and ICE [79] have exhibited unexpected reflective behaviors and policy improvements through *In-Context Learning* (ICL) [3, 58, 81, 115, 125]. However, these works primarily focus on single-agent decision-making tasks.

In this paper, we investigate how to leverage LLMs to develop agents capable of achieving foresighted cooperation with unknown partners, thereby addressing the challenges faced by existing approaches. There have been some preliminary attempts to develop such cooperative agents [65, 120]. *E.g.*, ProAgent [120], building upon self-reflection, incorporates reasoning about partner beliefs to enhance its cooperative abilities. Existing LLM agents tend to force LLMs to think in a *human-like manner* (*e.g.*, using CoT [107]) to generate better decisions. However, we argue that human-like reasoning mechanisms are not necessarily the best way to utilize LLMs' ICL abilities, as they can fail to fully utilize test-time computation.

How can we foster more structured LLM reasoning mechanisms that fully utilize test-time computation? In RL, there is a class of model-based methodology called *Decision-Time Planning* (DTP) [6, 7, 17, 40, 73, 89]. This method first learns an environment model, then simulates various trajectories on this model to evaluate the value of all legal actions, and finally selects the action with the highest value estimate. DTP generates 'foresighted' policies that are usually better than greedy decisions, as it leverages test-time computation to the fullest extent possible.

Enlightened by the above analysis, we introduce a novel LLM-driven DTP framework, **`DTPAgent`**, to facilitate more structured reasoning mechanisms. Unlike traditional DTP, **`DTPAgent`** does not rely on gradient descent but instead leverages LLMs' ICL ability to learn the environment model. In this framework, we instantiate one LLM as a **Partner Policy Model (PPM)** for predicting the *partner actions given states* and another LLM as a **Transition-Reward Model (TRM)** for estimating the environment's *transition and reward function*. Together, these two LLMs model the *full dynamics* of the environment, making DTP feasible. Upon the LLM-driven environment model, **`DTPAgent`** simulates a range of possible trajectories to search for the most advantageous policy at each timestep, enabling adaptive cooperation with unknown partners.

Our **`DTPAgent`** contributes a simple and general algorithmic framework. To generate foresighted cooperative policies, it only needs to continuously update the LLMs' contexts with full dynamic transitions obtained during interactions. More importantly, the design choices of **`DTPAgent`** naturally overcome the two key challenges of mainstream multi-agent coordination approaches: (1) **`DTPAgent`** learns to collaborate through the ICL ability of LLMs *without relying on complex and inefficient task-specific training*. (2) **`DTPAgent`** searches policies online by modeling the environment model and planning in real-time to *maximize its adaptability to unseen partners*. A fundamental advantage of our DTP framework is its ability to **directly convert test-time computation into improved performance**. As shown in Fig. 4, increasing the computational budget consistently leads to better coordination—a scalable property that existing reactive agents lack.

We conduct a series of comparative and ablation experiments on Overcooked [16], a popular and challenging multi-agent coordination benchmark. We build **`DTPAgent`** using representative open-source LLMs with fewer than 10B parameters. Despite using small-scale LLMs, the coordination results with unknown partners, including unseen human proxy models and all other approaches, show that **`DTPAgent`** generally outperforms various types of baselines, including mainstream PBT-driven ones and new-fashion LLM-driven ones. Compared to ablation variants, LLMs demonstrate more accurate estimations of the environment model, which is the foundation to the emergence of **`DTPAgent`**'s generalization ability.

2 RELATED WORK

Multi-Agent Coordination. There is a line of work in multi-agent systems that focuses on improving the generalization ability of agents to novel partners [50]. In cooperative games, this literature is commonly referred to as *ad hoc teamplay* [92] or *zero-shot coordination* [36], both of which emphasize the capacity to collaborate effectively without prior coordination or shared training history. Early works enable seamless cooperation by training precise partner models from collected data, particularly in the realm of human-AI collaboration [16, 52, 53, 101]. However, these approaches are fundamentally limited by the expensive data collection process and systematic biases inherent in human behavior [15, 48, 77, 82].

To circumvent the reliance on precise partner models, mainstream approaches employ PBT to create a diverse population of

partners and subsequently leverage this population to train a generalizable agent [44, 59, 60, 93, 112]. Specifically, some works explicitly optimize statistical metrics for diversity [67, 127], while others utilize expert knowledge-dependent approaches such as *domain randomization* [94, 98, 117] and *quality diversity* [75, 78, 108] to generate behaviorally diverse populations. Follow-up works attempt to minimize the cross-play reward between different agents in the population to encourage the emergence of distinct, high-level strategies [18, 21, 83]. Beyond PBT-driven approaches, complementary methods employ Bayesian inference to infer the partner type based on historical experience [55, 99, 109].

Despite the rich methodologies established by existing work, they all rely on complex and inefficient task-specific training. In contrast, our **DTPAgent** is purely driven by the ICL ability of LLMs. Through interactions, **DTPAgent** updates its policy in context without any fine-tuning (*i.e.*, gradient-based learning) during testing.

Verbal RL. Building LLM agents capable of completing challenging decision-making tasks through *In-Context Learning* (ICL) [3, 5, 8, 29, 58, 100, 111, 125] is an emerging research topic commonly called **verbal RL** [20, 85, 103, 116]. Many works adopt general-purpose closed-loop reasoning approaches via self-evaluation and reflection [49, 51, 104, 105], allowing agents to iteratively improve based on environmental feedback. Some also include domain knowledge of embodied agents in language feedback [33, 39, 70].

Another line of work uses hierarchical planning methods to decompose complex tasks into simpler, executable subtasks. Among them, some focus on instructing LLMs to continuously generate goals [23, 102, 110, 123], enabling LLM agents to explore open-ended environments. Others leverage LLMs to decompose a given goal into appropriate subgoals [19, 38, 76, 129], facilitating better solutions for long-horizon and complex tasks.

Recently, building upon ideas of Shinn et al. [85], Yao et al. [116], some studies have attempted to construct cooperative LLM agents by incorporating the modeling of partner beliefs or intentions [65, 120]. In contrast, our work investigates how to use DTP to foster more structured reasoning mechanisms, allowing LLMs to fully utilize test-time computation for scaling cooperation abilities.

Decision-Time Planning (DTP). DTP is an advanced online policy search method in model-based RL [25, 96, 97, 126]. The most representative is the AlphaGo series of works [84, 86–88], which achieved superhuman performance in real-world games such as Go, Chess, and Atari through self-play, RL training, and a DTP algorithm—*Monte Carlo Tree Search* [17, 31]. DTP also makes it possible for AI to defeat top human players in challenging imperfect-information games such as Texas Hold'em Poker [12, 71, 89].

Recently, using DTP to enhance reasoning abilities of LLM agents has attracted great research interest [34, 37, 61, 121, 128], which primarily considers LLMs as a world model and adopts tree search for planning. Compared to other methods, DTP emphasizes scaling test-time computations to foster more 'foresighted' LLM reasonings. However, these works are mostly limited to single-agent settings.

Inspired by previous DTP works, our work explores how to develop an LLM-driven DTP framework suited for multi-agent coordination. We adopt the idea of DTP to develop more structured LLM reasoning mechanisms that fully utilize test-time computation, enabling the online generation of adaptive cooperative policies.

3 PRELIMINARIES

In this work, we focus on the setting of two-player, fully cooperative multi-agent systems. We formalize the environment using a **T**wo-player **C**ooperative **M**arkov **G**ame (**TCMG**) [60, 112, 117] $\langle \mathcal{S}, \mathcal{A}, \mathcal{T}, R, T, \gamma \rangle$. Here, $\mathcal{S}$ represents the *set of all possible states*. $\mathcal{A}$ denotes the *set of all possible actions*, which is the same for each player. $\mathcal{T} : \mathcal{S} \times \mathcal{A} \times \mathcal{A} \to \mathcal{S}$ is the *transition function* over next states given states and actions from both players. $R : \mathcal{S} \times \mathcal{A} \times \mathcal{A} \to \mathcal{R}$ represents the *reward function* commonly shared between the two players. For simplicity, we jointly denote the transition function $\mathcal{T}$ and the reward function R as the *transition-reward function* $\mathcal{P} : \mathcal{S} \times \mathcal{A} \times \mathcal{A} \to \mathcal{S} \times \mathcal{R}$. T denotes the number of timesteps in each episode of the game. γ denotes the discount factor.

In this TCMG, one player is the **Agent** (*controlled by us*), while the other player is the **Partner** (*uncontrolled*). We use the superscript 'A' to represent terms related to the agent player and 'P' to represent those related to the partner player. In this way, we can denote the policies of the agent and the partner as π^{A} and π^{P}, respectively. Let the expected *return* (*i.e.*, *cumulative discounted reward*) obtained from the cooperation between the agent and the partner be denoted as $\mathcal{J}(\pi^{\mathrm{A}}, \pi^{\mathrm{P}}) = \mathbf{E}_{a^{\mathrm{A}} \sim \pi^{\mathrm{A}}, a^{\mathrm{P}} \sim \pi^{\mathrm{P}}} \left[\sum_{t=0}^{T-1} \gamma^t R(s_t, a_t^{\mathrm{A}}, a_t^{\mathrm{P}}) \right]$. Assume that $P_{\text{test}} : \Pi^{\mathrm{P}} \to [0, 1]$ represents the distribution of unknown partner policies during testing, where Π^{P} denotes the set of all possible partner policies. Our objective in this TCMG is to find an agent policy π^{A} that maximizes the expected return collaborating with unknown partners during testing, *i.e.*, $\arg\max_{\pi^{\mathrm{A}}} \mathbf{E}_{\pi^{\mathrm{P}} \sim P_{\text{test}}} \left[\mathcal{J}(\pi^{\mathrm{A}}, \pi^{\mathrm{P}}) \right]$.

4 METHODOLOGY

Existing multi-agent coordination approaches have **two key challenges**: (1) *They heavily rely on task-specific training;* (2) *Their trained agents lack adaptability at test time.* To address these challenges, we introduce a simple and general LLM-driven DTP framework, **DTPAgent**. Methodologically, **DTPAgent** leverages the ICL ability of pretrained LLMs to model the full dynamics of the environment, thus constructing a novel DTP framework. Upon this framework, **DTPAgent** collaborates with unknown partners by simulating possible trajectories to estimate action values and then generating foresighted cooperative policies. The overview of our LLM-driven DTP framework is illustrated in Fig. 1.

4.1 Building LLM-driven Decision-Time Planning Framework

To make DTP feasible for multi-agent coordination, two components are essential: (1) A **rollout policy** for the agent to simulate the agent actions a^{A} during DTP process. (2) An **environment model** to predict the next state s' and the reward r given the current state s and the agent action a^{A} during DTP process.

For the **rollout policy**, it is theoretically necessary to explore all possible trajectories, *i.e.*, trying every legal action at each decision node of the agent, to find the optimal action sequence for the agent. Insufficient exploration of the trajectory space can lead to the generation of arbitrarily suboptimal π^{A}. However, given the limited computational budget in practice, we consider using a *random policy* as the rollout policy instead of exhaustive traversal. This enables the agent to explore as large trajectory subspace as possible within the available resources.

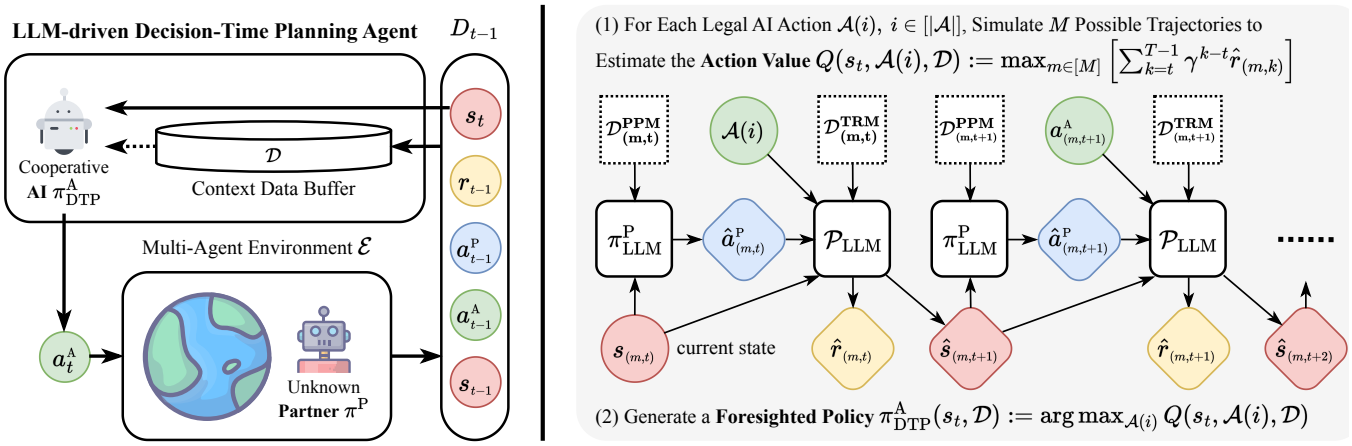

Figure 1: Left: Through continuous interaction with the environment, DTPAgent accumulates *context* to construct an increasingly accurate *modeling of the environment*, allowing it to adapt more effectively to unknown partners. Right: Policy search of DTPAgent at each timestep includes: (1) *Simulate Trajectories to Estimate Action Values*: Simulate trajectories using the LLM-driven *Partner Policy Model (PPM)* $\pi^{\mathrm{P}}_{\mathrm{LLM}}$ and *Transition-Reward Model (TRM)* $\mathcal{P}_{\mathrm{LLM}}$ to compute the value of each legal action. (2) *Generate a Foresighted Cooperative Policy*: Select the action with the highest estimated value to form an adaptive policy.

For the **environment model**, we propose explicitly modeling the partner policy to reduce the uncertainty caused by the partners. If the partner is treated as part of the environment, the environment can become *non-stationary*, as the partner could always adopt *stochastic*, *changing*, or *learning* policies. Consequently, we decouple the *full dynamics* of environment into two parts: a *partner policy* and a *transition-reward function*. These two dynamics are respectively modeled using the following two LLM instances.

Partner Policy Model (PPM). We instantiate one LLM as the PPM, denoted as $\pi^{\mathrm{P}}_{\mathrm{LLM}}(\hat{a}^{\mathrm{P}}|s, \mathcal{D}^{\mathrm{PPM}})$, to estimate the *partner policy* π^{P} based on the ICL ability of LLM.[1] Follow up, we will slightly abuse some mathematical language to intuitively describe the approximation mechanism by which PPM models the partner policy through ICL. Given an unknown partner policy π^{P}, let $\mathcal{D}^{\mathrm{PPM}}$ denote the *PPM Context Data*. Specifically, $\mathcal{D}^{\mathrm{PPM}}$ contains several ground-truth (s, a^{P}) tuples generated by π^{P}. These tuples serve as in-context examples, enabling $\pi^{\mathrm{P}}_{\mathrm{LLM}}$ to better approximate the true partner policy. Then, PPM can be approximately viewed as optimizing the formula:

$$\max_{\mathcal{D}^{\mathrm{PPM}}} \mathbf{E}_{a^{\mathrm{P}}\sim\pi^{\mathrm{P}}} \left[\log \pi^{\mathrm{P}}_{\mathrm{LLM}}(a^{\mathrm{P}}|s, \mathcal{D}^{\mathrm{PPM}})\right]. \tag{1}$$

To estimate the true partner policy, traditional methods approximate π^{P} using gradient descent with a great number of ground-truth (s, a^{P}) samples. However, these methods requires re-training when π^{P} changes, and exhibits high instability with limited training samples. In contrast, ICL has been widely demonstrated as a form of implicit gradient descent to learn algorithms in-context efficiently [3, 5, 8, 29, 58, 81, 100, 111, 125]. In light of this, we leverage the ICL ability of the LLM to implicitly achieve Eq. (1). In this way, even when the unknown partner policy π^{P} changes, the LLM can quickly approximate the new π^{P} with few samples, and is more stable as it does not rely on gradient descent.

[1]We use superscript $\wedge$ to denote the terms generated by LLMs.

To measure the quality of the partner policy estimated by PPM, we can directly collect the partner actions $\hat{a}^{\mathrm{P}}$ predicted by PPM and compare them with the ground-truth a^{P} to calculate the *Accuracy*.

Transition-Reward Model (TRM). We instantiate another LLM as the TRM, denoted as $\mathcal{P}_{\mathrm{LLM}}(\hat{s}', \hat{r}|s, a^{\mathrm{A}}, a^{\mathrm{P}}, \mathcal{D}^{\mathrm{TRM}})$, to model the *transition-reward function* $\mathcal{P}$ based on the ICL ability of LLM. Once again, we will slightly abuse mathematical notation to intuitively depict how TRM leverages ICL to estimate the transition-reward function. Given the true transition-reward function $\mathcal{P}$, let $\mathcal{D}^{\mathrm{TRM}}$ represent the *TRM Context Data*. Specifically, $\mathcal{D}^{\mathrm{TRM}}$ contains several ground-truth $(s, a^{\mathrm{A}}, a^{\mathrm{P}}, r, s')$ tuples collected during interactions. These tuples serve as in-context examples, enabling $\mathcal{P}_{\mathrm{LLM}}$ to more accurately predict the true transition-reward function. Assuming the general case where the state set $\mathcal{S}$ is continuous, TRM can be approximately viewed as optimizing the formula:

$$\min_{\mathcal{D}^{\mathrm{TRM}}} \mathbf{E}_{(\hat{s}',\hat{r})\sim\mathcal{P}_{\mathrm{LLM}}(\cdots,\mathcal{D}^{\mathrm{TRM}}),(s',r)\sim\mathcal{P}} \left[\frac{1}{2}(s'-\hat{s}')^2 + \frac{1}{2}(r-\hat{r})^2\right] \tag{2}$$

Similarly, rather than using gradient descent to learn the transition-reward function $\mathcal{P}$, we leverage the ICL ability of LLMs to implicitly approximate Eq. (2). In practice, the state set $\mathcal{S}$ of the environment we use can be discrete. In this case, to measure the quality of TRM's approximation of the transition-reward function, we propose a novel *Weighted Edit Distance* (WED) metric.

Specifically, we convert each variable into a string X decomposed into Z information segments, denoted as $X = X^{(1)} \oplus \cdots \oplus X^{(Z)}$. The WED metric aggregates the distances across these segments:

$$\mathcal{M}_{\mathrm{WED}}(X_i, X_j) := \sum_{z=1}^{Z} \omega_z \cdot \mathcal{M}_z(X_i^{(z)}, X_j^{(z)}), \tag{3}$$

where ω_z is the weight of the z-th segment and $\mathcal{M}_z$ is the corresponding metric. We use *Manhattan distance* for numerical segments and *Levenshtein distance* [57] for text segments. Tailored for natural language representations of states and rewards, WED

quantifies the similarity between predicted $(\hat{s}', \hat{r})$ and ground-truth (s', r) by calculating the minimum weighted operations required to transform one sequence into another. This metric effectively captures the accuracy of TRM's predictions in discrete state spaces.

Based on the **rollout policy** and the **environment model** described above, we construct a complete DTP framework. This framework is entirely driven by LLMs and does not rely on any task-specific training, eliminating the need to deal with complex learning stages and meticulous design choices. Essentially, LLMs learn to model the full dynamics of the environment purely in context and continuously refine this modeling through self-evaluation and reflection based on environmental feedback.

4.2 Collaborating with Unknown Partners using DTPAgent

During testing, **DTPAgent** collaborates with unknown partners at each timestep through the following two steps: (1) Simulate a range of possible trajectories to estimate action values; (2) Generate a foresighted cooperative policy based on the calculated value estimations. This process iteratively repeats until the testing ends. We provide the pseudocode for **DTPAgent** in Algo. 1.

Simulate Trajectories to Estimate Action Values. The core step of **DTPAgent** is to use the LLM-driven environment model at each timestep t to simulate various trajectories, enabling the value estimation of all legal agent actions in discrete set $\mathcal{A}$. Given the current state s_t and the *Context Data Buffer* $\mathcal{D}$, we adopt the following procedure to compute the action value $Q(s_t, a^{\mathrm{A}}, \mathcal{D})$ for all $a^{\mathrm{A}} \in \mathcal{A}$.

Let M denote the *rollout times* for a given legal action a^{A}. Given the current *rollout index* $m \in [M]$ and *simulation timestep* $h \in \{t, \ldots, T-1\}$, we first retrieve the PPM Context Data $\mathcal{D}^{\mathrm{PPM}}_{(m,h)}$ from $\mathcal{D}$. Then, we use the PPM $\pi^{\mathrm{P}}_{\mathrm{LLM}}$ to predict partner action $\hat{a}^{\mathrm{P}}_{(m,h)}$. Next, we retrieve the TRM Context Data $\mathcal{D}^{\mathrm{TRM}}_{(m,h)}$ from $\mathcal{D}$ and use the TRM $\mathcal{P}_{\mathrm{LLM}}$ to predict the next state $\hat{s}_{(m,h+1)}$ and reward $\hat{r}_{(m,h)}$. Finally, we sample next agent action $a^{\mathrm{A}}_{(m,h+1)}$ with the random policy. We repeat the above procedure until M complete trajectories are simulated, and approximate the value estimation for the action a^{A} given the state s_t with the *maximum simulated return*, *i.e.*,

$$Q(s_t, a^{\mathrm{A}}, \mathcal{D}) := \max_{m \in [M]} \left[\sum_{k=t}^{T-1} \gamma^{k-t} \hat{r}_{(m,k)} \right]. \tag{4}$$

Generate a Foresighted Cooperative Policy. Based on the action value estimations described above, **DTPAgent** generates an agent policy at each timestep t through a maximization operator:

$$\pi^{\mathrm{A}}_{\mathrm{DTP}}(s_t, \mathcal{D}) := \arg\max_{a^{\mathrm{A}} \in \mathcal{A}} Q(s_t, a^{\mathrm{A}}, \mathcal{D}). \tag{5}$$

As **DTPAgent** interacts with the *environment* $\mathcal{E}$, the *ground-truth full dynamic transition* $D_t := (s_t, a^{\mathrm{A}}_t, a^{\mathrm{P}}_t, r_t, s_{t+1})$ is generated. Naturally, we continuously add the newly generated D to the buffer $\mathcal{D}$, where $\mathcal{D}$ is used to update the estimation of Q in the next iteration.

Intuitively, as $\mathcal{D}$ continuously updates, the LLM-driven environment model becomes more accurate through ICL, leading to more reliable Q estimations, which in turn generates increasingly performant cooperative policy $\pi^{\mathrm{A}}_{\mathrm{DTP}}$. This iterative process essentially facilitates more structured LLM reasoning mechanisms, as it constructs an adaptive model of the environment, performs in-depth planning, and fully utilizes test-time computation to enhance generalization ability. Moreover, as we increase M, the estimate of Q becomes increasingly accurate, leading to the generation of a better cooperative policy $\pi^{\mathrm{A}}_{\mathrm{DTP}}$. This introduces a feasible path for scalability: as available computational resources increase, we can improve performance by increasing the number of simulated trajectories, a characteristic absent in existing reactive agents.

Algorithm 1 LLM-driven Decision-Time Planning Agent

```
// Simulate Trajectories to Estimate Action Values
function Q(s_t, a^A, D)
    Let rollout index m ← 0.
    repeat
        Let simulation timestep h ← t, state s_(m,h) = s_t, agent action a^A_(m,h) = a^A.
        repeat
            Retrieve the PPM Context Data D^PPM_(m,h) ~ D.
            Predict partner action â^P_(m,h) by sampling from the PPM π^P_LLM(·|s_(m,h), D^PPM_(m,h)).
            Retrieve the TRM Context Data D^TRM_(m,h) ~ D.
            Predict next state ŝ_(m,h+1) and reward r̂_(m,h) with the TRM P_LLM(·,·|s_(m,h), a^A_(m,h), â^P_(m,h), D^TRM_(m,h)).
            Sample next agent action a^A_(m,h+1) ~ Uniform[A].
            Let simulation timestep h ← h + 1.
        until h ≥ # timesteps per episode T.
        Let rollout index m ← m + 1.
    until m ≥ # simulated trajectories per legal action M.
    return max simulated return max_{m∈[M]} [Σ_{k=t}^{T-1} γ^{k-t} r̂_(m,k)].
end function
// Collaborate with Unknown Partners
function RunDTPAgent(environment E)
    Initialize the Context Data Buffer D = {}.
    while testing is not done do
        Let timestep t ← 0 and get initial state s_0 by resetting the environment E.
        while t < T do
            // Generate a Foresighted Cooperative Policy
            Generate agent action a^A_t with searched agent policy π^A_DTP(s_t, D) := arg max_{a^A∈A} Q(s_t, a^A, D).
            Get partner action a^P_t, next state s_{t+1}, and reward r_t by executing agent action a^A_t in the environment E.
            Add the ground-truth full dynamic transition D_t := (s_t, a^A_t, a^P_t, r_t, s_{t+1}) into the Context Data Buffer D.
            Let timestep t ← t + 1.
        end while
    end while
end function
```

5 EXPERIMENTS

In this section, Sec. 5.1 introduces the experimental setup in detail. Sec. 5.2 poses a series of questions and provides empirical results to answer them, aiming to analyze effectiveness of **DTPAgent**.

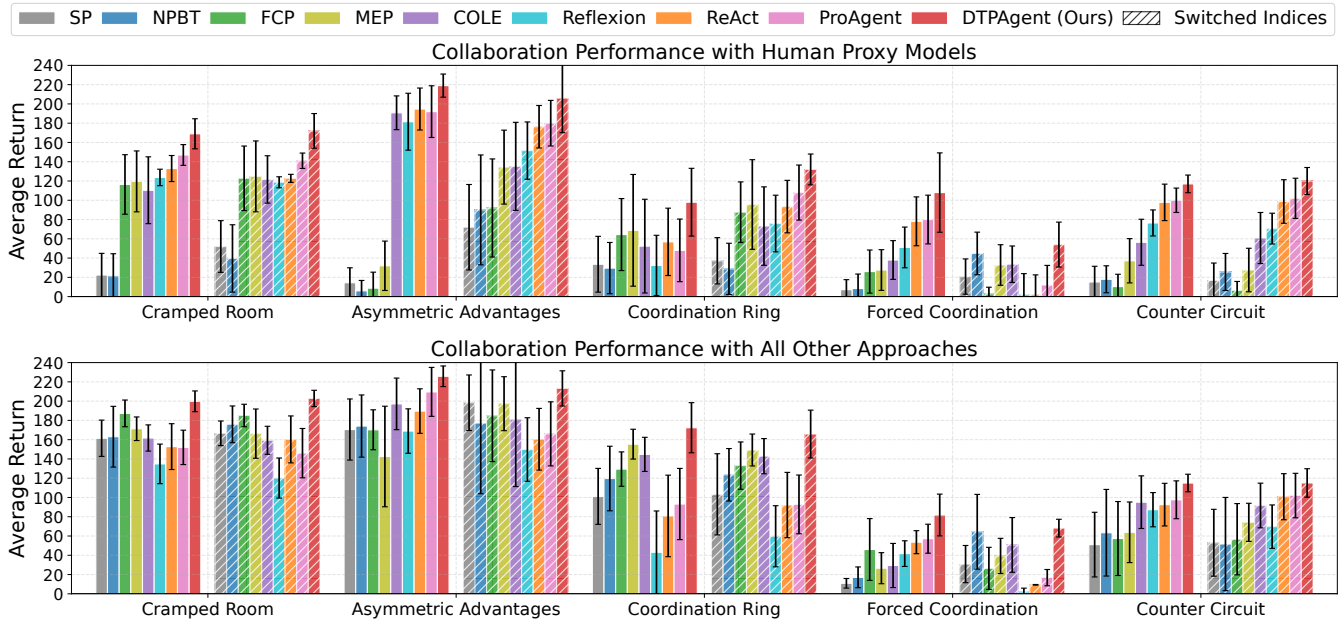

Figure 2: Top: Performance of agents collaborating with human proxy models as unknown partners. Bottom: Performance with all other approaches as unknown partners. The results show that DTPAgent generally outperforms various types of baselines, including PBT-driven ones and LLM-driven ones. We report the mean and standard deviation over 50 episodes.

5.1 Experimental Setup

Environment. We evaluate **DTPAgent** on Overcooked [16]. Overcooked is a representative TCMG where two players must skillfully coordinate, avoid collisions, and complete a series of subtasks under time pressure to serve dishes successfully. This requires the agent to infer the partner's intentions and decide which subtask would be most beneficial to assist the partner. We use five commonly adopted layouts: `Cramped Room`, `Asymmetric Advantages`, `Coordination Ring`, `Forced Coordination`, and `Counter Circuit`.

Baselines. We consider both mainstream PBT-driven and new-fashion LLM-driven approaches as comparative baselines.

PBT-driven baselines include: (1) **Self-Play (SP)** [9]: The agent is trained by interacting only with itself. (2) **Naive PBT (NPBT)** [41]: A population of agents is trained through random pair-wise interactions and a survival-of-the-fittest mechanism. (3) **FCP** [93]: A population is constructed using checkpoints from different stages of multiple runs of the SP algorithm, and the agent is trained to interact with this population. (4) **MEP** [127]: Builds on FCP by optimizing an objective to maximize population entropy, thereby increasing the diversity of the population. (5) **COLE** [59]: Utilizes an open-ended framework that continuously computes the population's cooperation incompatibility distribution and solves for the best response, gradually producing increasingly stronger agents. For the PBT-driven baselines, we use the open-source trained models provided by Li et al. [59] for evaluation.

LLM-driven baselines include: (1) **ReAct** [116]: Prompting the agent to first reason based on observations and then make a decision. (2) **Reflexion** [85]: The agent continuously reflects based on environmental feedback and improves decisions in subsequent trails. (3) **ProAgent** [120]: Compared to the previous two approaches, it introduces the modeling of partner beliefs to enhance the reasoning process. For the LLM-driven baselines, if not specified otherwise, we use the open-source pretrained LLM *Qwen2.5-7B-Instruct* [95, 113] for their implementations.

Testing Protocol. Following the best practices in multi-agent coordination, we set up two types of unknown partners to evaluate the generalization ability of the agent: (1) We use the *human proxy models* proposed in Carroll et al. [16] to approximate unseen human partners, which is widely used for simulating human-like play styles while avoiding expensive human evaluation [2, 56, 59, 83, 120]. (2) We use *all other approaches* to represent unseen agentic partners. For example, when **DTPAgent** acts as one of the players, this means we have all other approaches, including SP, NPBT, ..., ProAgent, take turns as the other player. These agentic partners often have less bias and higher skill levels compared to humans [48, 82].

In all figures and tables, we report the *mean* and *standard deviation* of performance or metric results over 50 game episodes. To practically implement **DTPAgent**, rule-based coding is used to convert between the ontology of states, actions, and rewards and their corresponding natural language representations. Unless otherwise specified, we use the LLM *Qwen2.5-7B-Instruct* to build **DTPAgent**. The rollout times M for **DTPAgent** is set to a default value of 10.

5.2 Emperical Analysis

Question 1. Compared to PBT-driven approaches, can **DTPAgent** *collaborate more effectively with the unknown partners (including human proxy models and all other approaches)?*

The Top of Fig. 2 presents the average return for all agents collaborating with the human proxy models. The Bottom of Fig. 2 shows the results with all other approaches as partners. We report

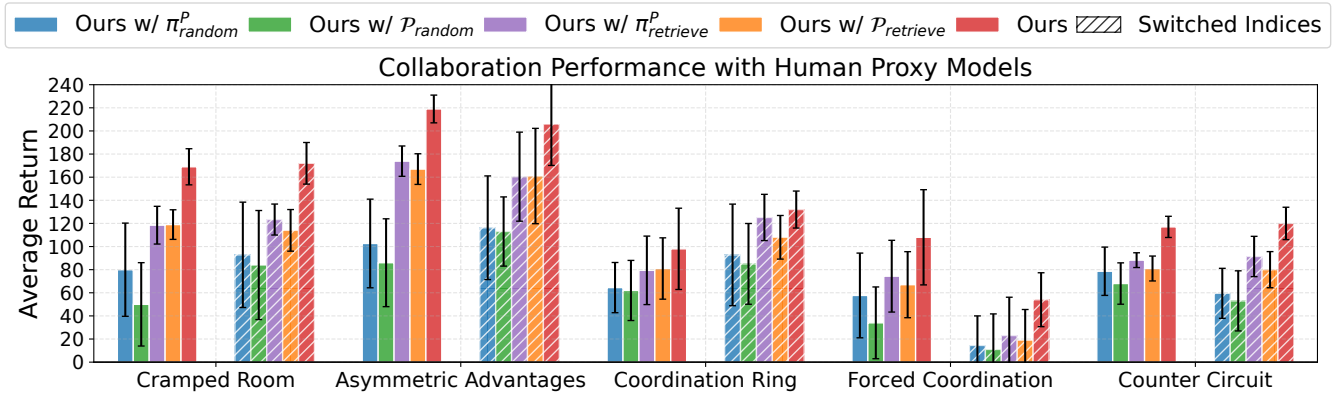

Figure 3: Performance of DTPAgent's ablation variants collaborating with human proxy models. The results show that modeling of partner policy and transition-reward function are both crucial for DTPAgent to collaborate effectively with unknown partners.

Table 1: Quantitative metrics of estimations on environment model by DTPAgent's ablation variants. We measure the quality of partner policy estimation using Accuracy (Acc.) and the quality of transition-reward function estimation using WED. The results show that both the PPM and TRM in DTPAgent can effectively model the dynamics they are responsible for.

Layout	Evaluation Term	Ablation Variant of DTPAgent				
		Ours w/ $\pi^{\mathrm{P}}_{\mathbf{random}}$	Ours w/ $\mathcal{P}_{\mathbf{random}}$	Ours w/ $\pi^{\mathrm{P}}_{\mathbf{retrieve}}$	Ours w/ $\mathcal{P}_{\mathbf{retrieve}}$	Ours
Cramped Room	PPM Acc. (%) ↑	16.9 ± 6.5	45.5 ± 9.6	20.5 ± 5.2	48.3 ± 7.2	**53.1 ± 8.6**
	TRM WED ↓	0.45 ± 0.62	2.56 ± 1.55	0.39 ± 0.67	1.36 ± 1.54	**0.26 ± 0.91**
Asymmetric Advantages	PPM Acc. (%) ↑	16.4 ± 4.3	50.7 ± 9.2	17.9 ± 6.1	50.4 ± 8.8	**62.4 ± 10.6**
	TRM WED ↓	0.35 ± 0.62	2.52 ± 1.35	0.44 ± 0.60	0.98 ± 1.32	**0.26 ± 0.99**
Coordination Ring	PPM Acc. (%) ↑	17.0 ± 6.5	**50.7 ± 8.8**	21.7 ± 6.0	41.9 ± 7.9	49.8 ± 9.7
	TRM WED ↓	0.41 ± 0.61	2.82 ± 1.58	0.33 ± 0.55	1.53 ± 1.53	**0.28 ± 0.50**
Forced Coordination	PPM Acc. (%) ↑	17.8 ± 7.9	43.1 ± 8.3	23.8 ± 7.9	44.6 ± 8.4	**51.5 ± 8.1**
	TRM WED ↓	0.38 ± 0.77	3.77 ± 2.09	**0.34 ± 0.73**	2.65 ± 2.21	0.43 ± 1.45
Counter Circuit	PPM Acc. (%) ↑	16.3 ± 7.7	46.6 ± 10.1	13.5 ± 6.9	47.1 ± 8.3	**59.3 ± 6.9**
	TRM WED ↓	0.37 ± 0.67	2.92 ± 1.60	0.37 ± 0.68	1.60 ± 1.50	**0.19 ± 0.50**

the results separately for the agent acting as *Player 0* and *Player 1*. The 'Switched Indices' with shaded highlight represents the case where the partner is *Player 0* and the agent is *Player 1*, while the non-shaded bars represents the opposite. We observe that **DTPAgent** generally collaborates more effectively with unknown partners to complete tasks compared to existing PBT-driven approaches. In some challenging cases, such as `Forced Coordination` with agents acting as *Player 0*, **DTPAgent** significantly outperforms other baselines. These results demonstrate that our LLM-driven DTP framework inherently produces foresighted cooperative policies, despite not being trained on the Overcooked task.

Question 2. Compared to LLM-driven approaches, can **DTPAgent** *exhibit better collaborations with the unknown partners?*

Continuing to observe Fig. 2, we will now focus on the comparison results between **DTPAgent** and other LLM-driven approaches. We can find that **DTPAgent** generally outperforms LLM-driven baselines in collaboration performance with unseen partners, demonstrating more foresighted cooperations. This finding can be attributed to the fact that, compared to LLM-driven approaches such as ReAct, Reflexion, ProAgent, and others that reasons by imitating the human thinking processes, **DTPAgent** fosters more structured LLM reasoning mechanisms, as it structurally models the full environment dynamics, searches policies in-depth, and effectively translates test-time computation into generalization ability.

*Question 3. Are the LLM-driven environment model (*i.e.*, PPM and TRM) the key to the emergence of* **DTPAgent***'s abilities?*

To quantitatively ablate the impact of the LLM-driven PPM and TRM, we introduce several variants of **DTPAgent**: (1) **Ours w/** $\pi^{\mathrm{P}}_{\mathbf{random}}$: Replaces $\pi^{\mathrm{P}}_{\mathrm{LLM}}$ with a random policy $\pi^{\mathrm{P}}_{\mathrm{random}}$ for partner action prediction. (2) **Ours w/** $\mathcal{P}_{\mathbf{random}}$: Replaces $\mathcal{P}_{\mathrm{LLM}}$ with $\mathcal{P}_{\mathrm{random}}$, who randomly samples the possible next state s' and reward r. (3) **Ours w/** $\pi^{\mathrm{P}}_{\mathbf{retrieve}}$: Replaces $\pi^{\mathrm{P}}_{\mathrm{LLM}}$ with a retrieval-based policy $\pi^{\mathrm{P}}_{\mathrm{retrieve}}$, who adopts the same retrieval strategy of **DTPAgent**'s $\mathcal{D}^{\mathrm{PPM}}$ to find the most similar transition data D in the buffer $\mathcal{D}$ and uses a^{P} from this D as prediction. (4) **Ours w/** $\mathcal{P}_{\mathbf{retrieve}}$: Replaces $\mathcal{P}_{\mathrm{LLM}}$ with a retrieval-based function $\mathcal{P}_{\mathrm{retrieve}}$, who adopts the same retrieval strategy of **DTPAgent**'s $\mathcal{D}^{\mathrm{TRM}}$ to find the most similar D in $\mathcal{D}$ and uses s' and r from this D as prediction.

We present the average return and statistical metrics for these ablation variants when collaborating with the human proxy models in Fig. 3 and Table 1, respectively. In Table 1, we measure the prediction accuracy of the PPM and TRM using *Accuracy* and *WED*, respectively. When using $\pi^{\mathrm{P}}_{\mathrm{random}}$ to replace $\pi^{\mathrm{P}}_{\mathrm{LLM}}$ or using $\mathcal{P}_{\mathrm{random}}$ to replace $\mathcal{P}_{\mathrm{LLM}}$, both the prediction accuracy and performance of **DTPAgent** drop significantly. Moreover, compared to using heuristic approaches like 'Ours w/ $\pi^{\mathrm{P}}_{\mathrm{retrieve}}$' or 'Ours w/ $\mathcal{P}_{\mathrm{retrieve}}$', **DTPAgent** ('Ours') shows considerable improvement in both prediction accuracy and performance. This indicates that LLMs can effectively

Table 2: Performance of DTPAgent constructed with different LLMs when collaborating with human proxy models. For each layout, the first and second row report the average return when the agent acts as *Player 0* and *Player 1*, respectively. We find that DTPAgent can be consistently effective with various LLMs, constituting a general framework for multi-agent coordination.

Layout	LLM Used to Construct DTPAgent				
	internlm2_5-7b-chat	*glm-4-9b-chat*	*Llama-3.1-8B-Instruct*	*Ministral-8B-Instruct-2410*	*Qwen2.5-7B-Instruct*
Cramped Room	163.6 ± 13.3 160.0 ± 20.9	168.8 ± 13.4 162.0 ± 15.0	160.8 ± 22.0 170.0 ± 22.3	150.4 ± 17.9 170.0 ± 20.4	168.8 ± 15.6 172.0 ± 18.0
Asymmetric Advantages	200.4 ± 9.8 178.0 ± 47.7	206.4 ± 10.8 162.0 ± 26.0	214.4 ± 16.0 182.0 ± 36.3	215.2 ± 22.3 176.0 ± 40.8	219.2 ± 12.0 206.0 ± 35.8
Coordination Ring	93.2 ± 32.0 122.0 ± 34.0	98.4 ± 32.3 122.0 ± 18.9	105.2 ± 37.1 118.0 ± 38.4	103.2 ± 40.0 134.0 ± 22.0	98.0 ± 35.2 132.0 ± 16.0
Forced Coordination	108.0 ± 16.0 62.0 ± 17.4	102.0 ± 26.0 44.0 ± 15.6	84.0 ± 46.3 52.0 ± 23.7	100.0 ± 26.8 52.0 ± 25.4	108.0 ± 41.2 54.0 ± 23.3
Counter Circuit	114.8 ± 18.0 108.8 ± 16.0	114.8 ± 16.0 115.6 ± 8.9	114.0 ± 9.8 120.8 ± 9.2	122.0 ± 20.0 110.4 ± 12.8	116.8 ± 9.2 120.0 ± 14.0

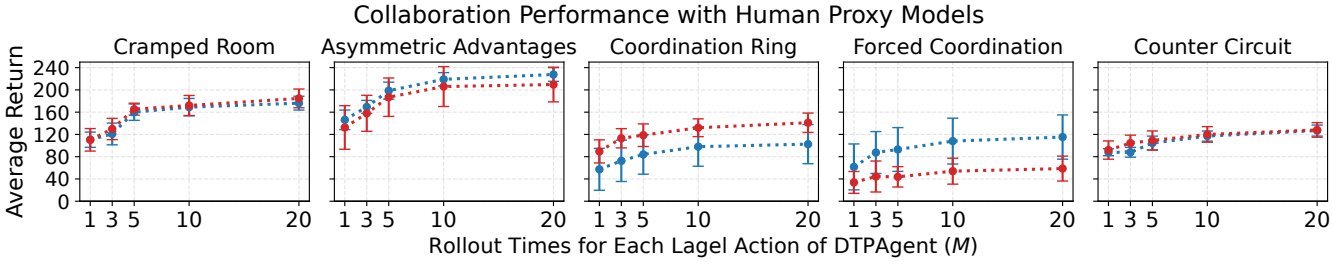

Figure 4: Performance curves of collaborating with human proxy models as rollout times M vary. For each layout, the blue and red curves represent the average return when the agent acts as *Player 0* and *Player 1*, respectively. The results show that as the rollout times M increase, DTPAgent's ability to adapt to unknown partners generally improves.

learn to model the full dynamics of environment purely in context, thereby endowing **DTPAgent** with strong adaptability.

Question 4. Can **DTPAgent** *be compatible with various kinds of LLMs to constitute a general framework for multi-agent coordination?*

We explore the generality of **DTPAgent** by constructing it with different open-source LLMs. In addition to *Qwen2.5-7B-Instruct*, we include: *internlm2_5-7b-chat* [14], *glm-4-9b-chat* [30], *Llama-3.1-8B-Instruct* [24], and *Ministral-8B-Instruct-2410* [45]. Table 2 presents the average return of **DTPAgent** constructed with different LLMs when collaborating with the human proxy models. Although there are some performance gaps across different LLMs, they all generally achieve good collaboration results with unseen human models. These results show that our **DTPAgent** contributes a general LLM-driven algorithmic framework for multi-agent coordination.

Question 5. Can **DTPAgent** *scale its ability on adapting to unknown partners as the available test-time computation increases?*

We are also concerned with the **scalability** of **DTPAgent**, *i.e.*, whether its cooperating ability improves as the test-time computation increases. Specifically, the *rollout times* M for each legal action constitute the major computational budget for **DTPAgent**. Larger M means more LLM inference. Fig. 4 presents the performance curve showing how the average return of **DTPAgent** changes as M increases when collaborating with the human proxy models. Within a limited set of hyperparameters, as M increases, **DTPAgent**'s adaptation ability shows an overall upward trend. This suggests that **DTPAgent** enables a scalable LLM reasoning mechanism, as it scales as test-time computation increases. This scalability is what existing reactive agents lack, which provides a feasible path for future improvements as more computational resources become available.

6 LIMITATIONS AND FUTURE WORK

Our experiments reveal that although LLMs provide better estimates of the environment model compared to random or heuristic predictions, there is still considerable room for improvement in their prediction accuracy. A potential future direction is to improve context data using methods like multimodal LLMs or data augmentation to further enhance the ability to predict the environment model through ICL. Additionally, the computational cost of scaling test-time LLM inference is relatively high. Introducing new methods to reduce the computational overhead of the LLM-driven DTP framework is a research direction worth further exploration. Possible techniques include model caching [10, 54] and compression [43, 62], hierarchical planning [4, 90], and pruning less promising trajectories [68, 91] to reduce computational cost while maintaining performance. Extending **DTPAgent** to more complex multi-agent systems, such as those with more than two agents or mixed motive settings, is also a promising future research direction.

ACKNOWLEDGMENTS

This work is supported in part by the National Key R&D Program of China (No. 2025ZD0122000), the Natural Science Foundation of China (Nos. 62222606 and 61902402), the Key Research and Development Program of Jiangsu Province (No. BE2023016), and the CCF-Baidu Open Fund.

REFERENCES

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

[2] Saaket Agashe, Yue Fan, Anthony Reyna, and Xin Eric Wang. 2023. Llm-coordination: evaluating and analyzing multi-agent coordination abilities in large language models. *arXiv preprint arXiv:2310.03903* (2023).

[3] Kwangjun Ahn, Xiang Cheng, Hadi Daneshmand, and Suvrit Sra. 2023. Transformers learn to implement preconditioned gradient descent for in-context learning. In *Advances in Neural Information Processing Systems*. 35151–35174.

[4] Anurag Ajay, Seungwook Han, Yilun Du, Shuang Li, Abhi Gupta, Tommi Jaakkola, Josh Tenenbaum, Leslie Kaelbling, Akash Srivastava, and Pulkit Agrawal. 2023. Compositional foundation models for hierarchical planning. *Advances in Neural Information Processing Systems* 36 (2023), 22304–22325.

[5] Ekin Akyürek, Dale Schuurmans, Jacob Andreas, Tengyu Ma, and Denny Zhou. 2023. What learning algorithm is in-context learning? investigations with linear models. In *International Conference on Learning Representations*.

[6] Safa Alver and Doina Precup. 2024. A look at value-based decision-time vs. background planning methods across different settings. In *Seventeenth European Workshop on Reinforcement Learning*.

[7] Ioannis Antonoglou, Julian Schrittwieser, Sherjil Ozair, Thomas K Hubert, and David Silver. 2021. Planning in stochastic environments with a learned model. In *International Conference on Learning Representations*.

[8] Yu Bai, Fan Chen, Huan Wang, Caiming Xiong, and Song Mei. 2023. Transformers as Statisticians: Provable In-Context Learning with In-Context Algorithm Selection. *arXiv preprint arXiv:2306.04637* (2023).

[9] Yu Bai, Chi Jin, and Tiancheng Yu. 2020. Near-optimal reinforcement learning with self-play. *Advances in neural information processing systems* 33 (2020), 2159–2170.

[10] Fu Bang. 2023. Gptcache: An open-source semantic cache for llm applications enabling faster answers and cost savings. In *Proceedings of the 3rd Workshop for Natural Language Processing Open Source Software (NLP-OSS 2023)*. 212–218.

[11] Anthony Brohan, Yevgen Chebotar, Chelsea Finn, Karol Hausman, Alexander Herzog, Daniel Ho, Julian Ibarz, Alex Irpan, Eric Jang, Ryan Julian, et al. 2023. Do as i can, not as i say: Grounding language in robotic affordances. In *Conference on robot learning*. PMLR, 287–318.

[12] Noam Brown, Anton Bakhtin, Adam Lerer, and Qucheng Gong. 2020. Combining deep reinforcement learning and search for imperfect-information games. *Advances in Neural Information Processing Systems* 33 (2020), 17057–17069.

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

[14] Zheng Cai, Maosong Cao, Haojiong Chen, Kai Chen, Keyu Chen, Xin Chen, Xun Chen, Zehui Chen, Zhi Chen, Pei Chu, Xiaoyi Dong, Haodong Duan, Qi Fan, Zhaoye Fei, Yang Gao, Jiaye Ge, Chenya Gu, Yuzhe Gu, Tao Gui, Aijia Guo, Qipeng Guo, Conghui He, Yingfan Hu, Ting Huang, Tao Jiang, Penglong Jiao, Zhenjiang Jin, Zhikai Lei, Jiaxing Li, Jingwen Li, Linyang Li, Shuaibin Li, Wei Li, Yining Li, Hongwei Liu, Jiangning Liu, Jiawei Hong, Kaiwen Liu, Kuikun Liu, Xiaoran Liu, Chengqi Lv, Haijun Lv, Kai Lv, Li Ma, Runyuan Ma, Zerun Ma, Wenchang Ning, Linke Ouyang, Jiantao Qiu, Yuan Qu, Fukai Shang, Yunfan Shao, Demin Song, Zifan Song, Zhihao Sui, Peng Sun, Yu Sun, Huanze Tang, Bin Wang, Guoteng Wang, Jiaqi Wang, Jiayu Wang, Rui Wang, Yudong Wang, Ziyi Wang, Xingjian Wei, Qizhen Weng, Fan Wu, Yingtong Xiong, Chao Xu, Ruiliang Xu, Hang Yan, Yirong Yan, Xiaogui Yang, Haochen Ye, Huaiyuan Ying, Jia Yu, Jing Yu, Yuhang Zang, Chuyu Zhang, Li Zhang, Pan Zhang, Peng Zhang, Ruijie Zhang, Shuo Zhang, Songyang Zhang, Wenjian Zhang, Wenwei Zhang, Xingcheng Zhang, Xinyue Zhang, Hui Zhao, Qian Zhao, Xiaomeng Zhao, Fengzhe Zhou, Zaida Zhou, Jingming Zhuo, Yicheng Zou, Xipeng Qiu, Yu Qiao, and Dahua Lin. 2024. InternLM2 Technical Report. arXiv:2403.17297 [cs.CL]

[15] Colin F Camerer. 2011. *Behavioral game theory: Experiments in strategic interaction*. Princeton university press.

[16] Micah Carroll, Rohin Shah, Mark K Ho, Tom Griffiths, Sanjit Seshia, Pieter Abbeel, and Anca Dragan. 2019. On the utility of learning about humans for human-ai coordination. *Advances in neural information processing systems* 32 (2019).

[17] Alberto Castellini, Federico Bianchi, Edoardo Zorzi, Thiago D Simao, Alessandro Farinelli, and Matthijs TJ Spaan. 2023. Scalable safe policy improvement via Monte Carlo tree search. In *International Conference on Machine Learning*. PMLR, 3732–3756.

[18] Rujikorn Charakorn, Poramate Manoonpong, and Nat Dilokthanakul. 2023. Generating diverse cooperative agents by learning incompatible policies. In *The Eleventh International Conference on Learning Representations*.

[19] Jiaqi Chen, Yuxian Jiang, Jiachen Lu, and Li Zhang. 2024. S-Agents: Self-organizing Agents in Open-ended Environments. In *ICLR 2024 Workshop on Large Language Model (LLM) Agents*.

[20] Ching-An Cheng, Allen Nie, and Adith Swaminathan. 2024. Trace is the Next AutoDiff: Generative Optimization with Rich Feedback, Execution Traces, and LLMs. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

[21] Brandon Cui, Andrei Lupu, Samuel Sokota, Hengyuan Hu, David J Wu, and Jakob Nicolaus Foerster. 2023. Adversarial diversity in hanabi. In *The Eleventh International Conference on Learning Representations*.

[22] Allan Dafoe, Edward Hughes, Yoram Bachrach, Tantum Collins, Kevin R McKee, Joel Z Leibo, Kate Larson, and Thore Graepel. 2020. Open problems in cooperative ai. *arXiv preprint arXiv:2012.08630* (2020).

[23] Yuqing Du, Olivia Watkins, Zihan Wang, Cédric Colas, Trevor Darrell, Pieter Abbeel, Abhishek Gupta, and Jacob Andreas. 2023. Guiding pretraining in reinforcement learning with large language models. In *International Conference on Machine Learning*. PMLR, 8657–8677.

[24] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783* (2024).

[25] Yonathan Efroni, Gal Dalal, Bruno Scherrer, and Shie Mannor. 2018. Beyond the one-step greedy approach in reinforcement learning. In *International Conference on Machine Learning*. PMLR, 1387–1396.

[26] Angela Fan, Beliz Gokkaya, Mark Harman, Mitya Lyubarskiy, Shubho Sengupta, Shin Yoo, and Jie M Zhang. 2023. Large language models for software engineering: Survey and open problems. In *2023 IEEE/ACM International Conference on Software Engineering: Future of Software Engineering (ICSE-FoSE)*. IEEE, 31–53.

[27] Arnaud Fickinger, Hengyuan Hu, Brandon Amos, Stuart Russell, and Noam Brown. 2021. Scalable online planning via reinforcement learning fine-tuning. *Advances in Neural Information Processing Systems* 34 (2021), 16951–16963.

[28] Carole S Franklin, Elena G Dominguez, Jeff D Fryman, and Mark L Lewandowski. 2020. Collaborative robotics: New era of human–robot cooperation in the workplace. *Journal of Safety Research* 74 (2020), 153–160.

[29] Shivam Garg, Dimitris Tsipras, Percy S Liang, and Gregory Valiant. 2022. What can transformers learn in-context? a case study of simple function classes. In *Advances in Neural Information Processing Systems*, Vol. 35. 30583–30598.

[30] Team GLM, Aohan Zeng, Bin Xu, Bowen Wang, Chenhui Zhang, Da Yin, Diego Rojas, Guanyu Feng, Hanlin Zhao, Hanyu Lai, Hao Yu, Hongning Wang, Jiadai Sun, Jiajie Zhang, Jiale Cheng, Jiayi Gui, Jie Tang, Jing Zhang, Juanzi Li, Lei Zhao, Lindong Wu, Lucen Zhong, Mingdao Liu, Minlie Huang, Peng Zhang, Qinkai Zheng, Rui Lu, Shuaiqi Duan, Shudan Zhang, Shulin Cao, Shuxun Yang, Weng Lam Tam, Wenyi Zhao, Xiao Liu, Xiao Xia, Xiaohan Zhang, Xiaotao Gu, Xin Lv, Xinghan Liu, Xinyi Liu, Xinyue Yang, Xixuan Song, Xunkai Zhang, Yifan An, Yifan Xu, Yilin Niu, Yuantao Yang, Yueyan Li, Yushi Bai, Yuxiao Dong, Zehan Qi, Zhaoyu Wang, Zhen Yang, Zhengxiao Du, Zhenyu Hou, and Zihan Wang. 2024. ChatGLM: A Family of Large Language Models from GLM-130B to GLM-4 All Tools. arXiv:2406.12793 [cs.CL]

[31] Jean-Bastien Grill, Florent Altché, Yunhao Tang, Thomas Hubert, Michal Valko, Ioannis Antonoglou, and Rémi Munos. 2020. Monte-Carlo tree search as regularized policy optimization. In *International Conference on Machine Learning*. PMLR, 3769–3778.

[32] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948* (2025).

[33] Yanjiang Guo, Yen-Jen Wang, Lihan Zha, and Jianyu Chen. 2024. Doremi: Grounding language model by detecting and recovering from plan-execution misalignment. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 12124–12131.

[34] Shibo Hao, Yi Gu, Haodi Ma, Joshua Jiahua Hong, Zhen Wang, Daisy Zhe Wang, and Zhiting Hu. 2023. Reasoning with Language Model is Planning with World Model. In *The 2023 Conference on Empirical Methods in Natural Language Processing*.

[35] Ehsan Hosseini-Asl, Bryan McCann, Chien-Sheng Wu, Semih Yavuz, and Richard Socher. 2020. A simple language model for task-oriented dialogue. *Advances in Neural Information Processing Systems* 33 (2020), 20179–20191.

[36] Hengyuan Hu, Adam Lerer, Alex Peysakhovich, and Jakob Foerster. 2020. "other-play" for zero-shot coordination. In *International Conference on Machine Learning*. PMLR, 4399–4410.

[37] Mengkang Hu, Yao Mu, Xinmiao Chelsey Yu, Mingyu Ding, Shiguang Wu, Wenqi Shao, Qiguang Chen, Bin Wang, Yu Qiao, and Ping Luo. 2023. Tree-Planner: Efficient Close-loop Task Planning with Large Language Models. In *The Twelfth International Conference on Learning Representations*.

[38] Wenlong Huang, Pieter Abbeel, Deepak Pathak, and Igor Mordatch. 2022. Language models as zero-shot planners: Extracting actionable knowledge for embodied agents. In *International conference on machine learning*. PMLR, 9118–9147.

[39] Wenlong Huang, Fei Xia, Ted Xiao, Harris Chan, Jacky Liang, Pete Florence, Andy Zeng, Jonathan Tompson, Igor Mordatch, Yevgen Chebotar, et al. 2023. Inner Monologue: Embodied Reasoning through Planning with Language Models. In *Conference on Robot Learning*. PMLR, 1769–1782.

[40] Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Mohammadamin Barekatain, Simon Schmitt, and David Silver. 2021. Learning and planning in complex action spaces. In *International Conference on Machine Learning*. PMLR, 4476–4486.

[41] Max Jaderberg, Wojciech M Czarnecki, Iain Dunning, Luke Marris, Guy Lever, Antonio Garcia Castaneda, Charles Beattie, Neil C Rabinowitz, Ari S Morcos, Avraham Ruderman, et al. 2019. Human-level performance in 3D multiplayer games with population-based reinforcement learning. *Science* 364, 6443 (2019), 859–865.

[42] Max Jaderberg, Valentin Dalibard, Simon Osindero, Wojciech M Czarnecki, Jeff Donahue, Ali Razavi, Oriol Vinyals, Tim Green, Iain Dunning, Karen Simonyan, et al. 2017. Population based training of neural networks. *arXiv preprint arXiv:1711.09846* (2017).

[43] Ajay Jaiswal, Zhe Gan, Xianzhi Du, Bowen Zhang, Zhangyang Wang, and Yinfei Yang. 2023. Compressing llms: The truth is rarely pure and never simple. *arXiv preprint arXiv:2310.01382* (2023).

[44] Kunal Jha, Wilka Carvalho, Yancheng Liang, Simon S Du, Max Kleiman-Weiner, and Natasha Jaques. 2025. Cross-environment Cooperation Enables Zero-shot Multi-agent Coordination. *arXiv preprint arXiv:2504.12714* (2025).

[45] Albert Jiang, Alexandre Abou Chahine, Alexandre Sablayrolles, Alexis Tacnet, Alodie Boissonnet, Alok Kothari, Amélie Héliou, Andy Lo, Anna Peronnin, Antoine Meunier, Antoine Roux, Antonin Faure, Aritra Paul, Arthur Darcet, Arthur Mensch, Audrey Herblin-Stoop, Augustin Garreau, Austin Birky, Avinash Sooriyarachchi, Baptiste Rozière, Barry Conklin, Bastien Bouillon, Blanche Savary de Beauregard, Carole Rambaud, Caroline Feldman, Charles de Freminville, Charline Mauro, Chih-Kuan Yeh, Chris Bamford, Clement Auguy, Corentin Heintz, Cyriaque Dubois, Devendra Singh Chaplot, Diego Las Casas, Diogo Costa, Eléonore Arcelin, Emma Bou Hanna, Etienne Metzger, Fanny Olivier Autran, Francois Lesage, Garance Gourdel, Gaspard Blanchet, Gaspard Donada Vidal, Gianna Maria Lengyel, Guillaume Bour, Guillaume Lample, Gustave Denis, Harizo Rajaona, Himanshu Jaju, Ian Mack, Ian Mathew, Jean-Malo Delignon, Jeremy Facchetti, Jessica Chudnovsky, Joachim Studnia, Justus Murke, Kartik Khandelwal, Kenneth Chiu, Kevin Riera, Leonard Blier, Leonard Suslian, Leonardo Deschaseaux, Louis Martin, Louis Ternon, Lucile Saulnier, Lélio Renard Lavaud, Sophia Yang, Margaret Jennings, Marie Pellat, Marie Torelli, Marjorie Janiewicz, Mathis Felardos, Maxime Darrin, Michael Hoff, Mickaël Seznec, Misha Jessel Kenyon, Nayef Derwiche, Nicolas Carmont Zaragoza, Nicolas Faurie, Nicolas Moreau, Nicolas Schuhl, Nikhil Raghuraman, Niklas Muhs, Olivier de Garrigues, Patricia Rozé, Patricia Wang, Patrick von Platen, Paul Jacob, Pauline Buche, Pavankumar Reddy Muddireddy, Perry Savas, Pierre Stock, Pravesh Agrawal, Renaud de Peretti, Romain Sauvestre, Romain Sinthe, Roman Soletskyi, Sagar Vaze, Sandeep Subramanian, Saurabh Garg, Soham Ghosh, Sylvain Regnier, Szymon Antoniak, Teven Le Scao, Theophile Gervet, Thibault Schueller, Thibaut Lavril, Thomas Wang, Timothée Lacroix, Valeriia Nemychnikova, Wendy Shang, William El Sayed, and William Marshall. 2024. Ministral-8B-Instruct-2410. https://huggingface.co/mistralai/Ministral-8B-Instruct-2410

[46] Carlos E Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik Narasimhan. 2023. Swe-bench: Can language models resolve real-world github issues? *arXiv preprint arXiv:2310.06770* (2023).

[47] Hongye Jin, Xiaotian Han, Jingfeng Yang, Zhimeng Jiang, Zirui Liu, Chia-Yuan Chang, Huiyuan Chen, and Xia Hu. 2024. Llm maybe longlm: Self-extend llm context window without tuning. *arXiv preprint arXiv:2401.01325* (2024).

[48] Bryan D Jones. 1999. Bounded rationality. *Annual review of political science* 2, 1 (1999), 297–321.

[49] Geunwoo Kim, Pierre Baldi, and Stephen McAleer. 2023. Language models can solve computer tasks. *Advances in Neural Information Processing Systems* 36 (2023), 39648–39677.

[50] Paul Knott, Micah Carroll, Sam Devlin, Kamil Ciosek, Katja Hofmann, Anca Dragan, and Rohin Shah. 2021. Evaluating the Robustness of Collaborative Agents. In *Proceedings of the 20th International Conference on Autonomous Agents and MultiAgent Systems*. 1560–1562.

[51] Michal Kosinski. 2023. Theory of mind may have spontaneously emerged in large language models. *arXiv preprint arXiv:2302.02083* 4 (2023), 169.

[52] Hadas Kress-Gazit, Kerstin Eder, Guy Hoffman, Henny Admoni, Brenna Argall, Rüdiger Ehlers, Christoffer Heckman, Nils Jansen, Ross Knepper, Jan Křetínskỳ, et al. 2021. Formalizing and guaranteeing human-robot interaction. *Commun. ACM* 64, 9 (2021), 78–84.

[53] Minae Kwon, Erdem Biyik, Aditi Talati, Karan Bhasin, Dylan P Losey, and Dorsa Sadigh. 2020. When humans aren't optimal: Robots that collaborate with risk-aware humans. In *Proceedings of the 2020 ACM/IEEE international conference on human-robot interaction*. 43–52.

[54] Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th symposium on operating systems principles*. 611–626.

[55] Cassidy Laidlaw and Anca Dragan. 2022. The Boltzmann Policy Distribution: Accounting for Systematic Suboptimality in Human Models. In *International Conference on Learning Representations*.

[56] Niklas Lauffer, Ameesh Shah, Micah Carroll, Michael D Dennis, and Stuart Russell. 2023. Who needs to know? minimal knowledge for optimal coordination. In *International Conference on Machine Learning*. PMLR, 18599–18613.

[57] VI Levenshtein. 1966. Binary codes capable of correcting deletions, insertions, and reversals. *Proceedings of the Soviet physics doklady* (1966).

[58] Yingcong Li, M Emrullah Ildiz, Dimitris Papailiopoulos, and Samet Oymak. 2023. Transformers as algorithms: Generalization and implicit model selection in in-context learning. In *International Conference on Machine Learning*. 19565–19594.

[59] Yang Li, Shao Zhang, Jichen Sun, Yali Du, Ying Wen, Xinbing Wang, and Wei Pan. 2023. Cooperative open-ended learning framework for zero-shot coordination. In *International Conference on Machine Learning*. PMLR, 20470–20484.

[60] Yancheng Liang, Daphne Chen, Abhishek Gupta, Simon Shaolei Du, and Natasha Jaques. 2024. Learning to Cooperate with Humans using Generative Agents. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

[61] Jessy Lin, Yuqing Du, Olivia Watkins, Danijar Hafner, Pieter Abbeel, Dan Klein, and Anca Dragan. 2023. Learning to Model the World With Language. In *Forty-first International Conference on Machine Learning*.

[62] Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, Wei-Ming Chen, Wei-Chen Wang, Guangxuan Xiao, Xingyu Dang, Chuang Gan, and Song Han. 2024. Awq: Activation-aware weight quantization for on-device llm compression and acceleration. *Proceedings of machine learning and systems* 6 (2024), 87–100.

[63] Jessy Lin, Nicholas Tomlin, Jacob Andreas, and Jason Eisner. 2024. Decision-oriented dialogue for human-ai collaboration. *Transactions of the Association for Computational Linguistics* 12 (2024), 892–911.

[64] Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. 2024. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437* (2024).

[65] Jijia Liu, Chao Yu, Jiaxuan Gao, Yuqing Xie, Qingmin Liao, Yi Wu, and Yu Wang. 2024. LLM-Powered Hierarchical Language Agent for Real-time Human-AI Coordination. In *Proceedings of the 23rd International Conference on Autonomous Agents and Multiagent Systems*. 1219–1228.

[66] Ryan Lowe, Yi I Wu, Aviv Tamar, Jean Harb, OpenAI Pieter Abbeel, and Igor Mordatch. 2017. Multi-agent actor-critic for mixed cooperative-competitive environments. *Advances in neural information processing systems* 30 (2017).

[67] Andrei Lupu, Brandon Cui, Hengyuan Hu, and Jakob Foerster. 2021. Trajectory diversity for zero-shot coordination. In *International conference on machine learning*. PMLR, 7204–7213.

[68] Xinyin Ma, Gongfan Fang, and Xinchao Wang. 2023. Llm-pruner: On the structural pruning of large language models. *Advances in neural information processing systems* 36 (2023), 21702–21720.

[69] Sumbal Malik, Manzoor Ahmed Khan, and Hesham El-Sayed. 2021. Collaborative autonomous driving—A survey of solution approaches and future challenges. *Sensors* 21, 11 (2021), 3783.

[70] Zhao Mandi, Shreeya Jain, and Shuran Song. 2024. Roco: Dialectic multi-robot collaboration with large language models. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 286–299.

[71] Matej Moravčík, Martin Schmid, Neil Burch, Viliam Lisỳ, Dustin Morrill, Nolan Bard, Trevor Davis, Kevin Waugh, Michael Johanson, and Michael Bowling. 2017. Deepstack: Expert-level artificial intelligence in heads-up no-limit poker. *Science* 356, 6337 (2017), 508–513.

[72] Hadi Nekoei, Xutong Zhao, Janarthanan Rajendran, Miao Liu, and Sarath Chandar. 2023. Towards few-shot coordination: Revisiting ad-hoc teamplay challenge in the game of hanabi. In *Conference on Lifelong Learning Agents*. PMLR, 861–877.

[73] Yazhe Niu, Yuan Pu, Zhenjie Yang, Xueyan Li, Tong Zhou, Jiyuan Ren, Shuai Hu, Hongsheng Li, and Yu Liu. 2024. Lightzero: A unified benchmark for monte carlo tree search in general sequential decision scenarios. *Advances in Neural Information Processing Systems* 36 (2024).

[74] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems* 35 (2022), 27730–27744.

[75] Jack Parker-Holder, Aldo Pacchiano, Krzysztof M Choromanski, and Stephen J Roberts. 2020. Effective diversity in population based reinforcement learning. *Advances in Neural Information Processing Systems* 33 (2020), 18050–18062.

[76] Archiki Prasad, Alexander Koller, Mareike Hartmann, Peter Clark, Ashish Sabharwal, Mohit Bansal, and Tushar Khot. 2024. ADaPT: As-Needed Decomposition and Planning with Language Models. In *Findings of the Association for Computational Linguistics: NAACL 2024*. 4226–4252.

[77] John W Pratt. 1978. Risk aversion in the small and in the large. In *Uncertainty in economics*. Elsevier, 59–79.

[78] Justin K Pugh, Lisa B Soros, and Kenneth O Stanley. 2016. Quality diversity: A new frontier for evolutionary computation. *Frontiers in Robotics and AI* 3 (2016), 202845.

[79] Cheng Qian, Shihao Liang, Yujia Qin, Yining Ye, Xin Cong, Yankai Lin, Yesai Wu, Zhiyuan Liu, and Maosong Sun. 2024. Investigate-consolidate-exploit: A general strategy for inter-task agent self-evolution. *arXiv preprint arXiv:2401.13996* (2024).

[80] Tabish Rashid, Mikayel Samvelyan, Christian Schroeder De Witt, Gregory Farquhar, Jakob Foerster, and Shimon Whiteson. 2020. Monotonic value function factorisation for deep multi-agent reinforcement learning. *Journal of Machine Learning Research* 21, 178 (2020), 1–51.

[81] Allan Raventos, Mansheej Paul, Feng Chen, and Surya Ganguli. 2023. Pretraining task diversity and the emergence of non-Bayesian in-context learning for regression. In *Advances in Neural Information Processing Systems*. 1–13.

[82] Stuart Russell. 2019. *Human compatible: AI and the problem of control*. Penguin Uk.

[83] Bidipta Sarkar, Andy Shih, and Dorsa Sadigh. 2024. Diverse conventions for human-AI collaboration. *Advances in Neural Information Processing Systems* 36 (2024).

[84] Julian Schrittwieser, Ioannis Antonoglou, Thomas Hubert, Karen Simonyan, Laurent Sifre, Simon Schmitt, Arthur Guez, Edward Lockhart, Demis Hassabis, Thore Graepel, et al. 2020. Mastering atari, go, chess and shogi by planning with a learned model. *Nature* 588, 7839 (2020), 604–609.

[85] Noah Shinn, Beck Labash, and Ashwin Gopinath. 2023. Reflexion: an autonomous agent with dynamic memory and self-reflection. *arXiv preprint arXiv:2303.11366* 2 (2023), 9.

[86] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. 2016. Mastering the game of Go with deep neural networks and tree search. *Nature* 529, 7587 (2016), 484–489.

[87] David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, et al. 2018. A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play. *Science* 362, 6419 (2018), 1140–1144.

[88] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. 2017. Mastering the game of Go without human knowledge. *Nature* 550, 7676 (2017), 354–359.

[89] Samuel Sokota, Gabriele Farina, David J Wu, Hengyuan Hu, Kevin A Wang, J Zico Kolter, and Noam Brown. 2024. The Update-Equivalence Framework for Decision-Time Planning. In *The Twelfth International Conference on Learning Representations*.

[90] Chan Hee Song, Jiaman Wu, Clayton Washington, Brian M Sadler, Wei-Lun Chao, and Yu Su. 2023. Llm-planner: Few-shot grounded planning for embodied agents with large language models. In *Proceedings of the IEEE/CVF international conference on computer vision*. 2998–3009.

[91] Sharath Turuvekere Sreenivas, Saurav Muralidharan, Raviraj Joshi, Marcin Chochowski, Ameya Sunil Mahabaleshwarkar, Gerald Shen, Jiaqi Zeng, Zijia Chen, Yoshi Suhara, Shizhe Diao, et al. 2024. Llm pruning and distillation in practice: The minitron approach. *arXiv preprint arXiv:2408.11796* (2024).

[92] Peter Stone, Gal Kaminka, Sarit Kraus, and Jeffrey Rosenschein. 2010. Ad hoc autonomous agent teams: Collaboration without pre-coordination. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 24. 1504–1509.

[93] DJ Strouse, Kevin McKee, Matt Botvinick, Edward Hughes, and Richard Everett. 2021. Collaborating with humans without human data. *Advances in Neural Information Processing Systems* 34 (2021), 14502–14515.

[94] Zhenggang Tang, Chao Yu, Boyuan Chen, Huazhe Xu, Xiaolong Wang, Fei Fang, Simon Shaolei Du, Yu Wang, and Yi Wu. 2021. Discovering Diverse Multi-Agent Strategic Behavior via Reward Randomization. In *International Conference on Learning Representations*.

[95] Qwen Team. 2024. Qwen2.5: A Party of Foundation Models. https://qwenlm.github.io/blog/qwen2.5/

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

[97] Gerald Tesauro and Gregory Galperin. 1996. On-line policy improvement using Monte-Carlo search. *Advances in neural information processing systems* 9 (1996).

[98] Johannes Treutlein, Michael Dennis, Caspar Oesterheld, and Jakob Foerster. 2021. A new formalism, method and open issues for zero-shot coordination. In *International Conference on Machine Learning*. PMLR, 10413–10423.

[99] Victor Villin, Thomas Kleine Buening, and Christos Dimitrakakis. 2025. A Minimax Approach to Ad Hoc Teamwork. *arXiv preprint arXiv:2502.02377* (2025).

[100] Johannes Von Oswald, Eyvind Niklasson, Ettore Randazzo, João Sacramento, Alexander Mordvintsev, Andrey Zhmoginov, and Max Vladymyrov. 2023. Transformers learn in-context by gradient descent. In *International Conference on Machine Learning*. PMLR, 35151–35174.

[101] Chen Wang, Claudia Pérez-D'Arpino, Danfei Xu, Li Fei-Fei, Karen Liu, and Silvio Savarese. 2022. Co-gail: Learning diverse strategies for human-robot collaboration. In *Conference on Robot Learning*. PMLR, 1279–1290.

[102] Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. 2023. Voyager: An Open-Ended Embodied Agent with Large Language Models. *Transactions on Machine Learning Research* (2023).

[103] Jun Wang, Meng Fang, Ziyu Wan, Muning Wen, Jiachen Zhu, Anjie Liu, Ziqin Gong, Yan Song, Lei Chen, Lionel M Ni, et al. 2024. Openr: An open source framework for advanced reasoning with large language models. *arXiv preprint arXiv:2410.09671* (2024).

[104] Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022. Self-Consistency Improves Chain of Thought Reasoning in Language Models. In *The Eleventh International Conference on Learning Representations*.

[105] Zihao Wang, Shaofei Cai, Guanzhou Chen, Anji Liu, Xiaojian Ma, Yitao Liang, and Team CraftJarvis. 2023. Describe, explain, plan and select: interactive planning with large language models enables open-world multi-task agents. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*. 34153–34189.

[106] Zihao Wang, Shaofei Cai, Anji Liu, Yonggang Jin, Jinbing Hou, Bowei Zhang, Haowei Lin, Zhaofeng He, Zilong Zheng, Yaodong Yang, et al. 2024. Jarvis-1: Open-world multi-task agents with memory-augmented multimodal language models. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2024).

[107] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems* 35 (2022), 24824–24837.

[108] Shuang Wu, Jian Yao, Haobo Fu, Ye Tian, Chao Qian, Yaodong Yang, Qiang Fu, and Yang Wei. 2023. Quality-similar diversity via population based reinforcement learning. In *The Eleventh International Conference on Learning Representations*.

[109] Sarah A Wu, Rose E Wang, James A Evans, Joshua B Tenenbaum, David C Parkes, and Max Kleiman-Weiner. 2021. Too many cooks: Bayesian inference for coordinating multi-agent collaboration. *Topics in Cognitive Science* 13, 2 (2021), 414–432.

[110] Yue Wu, So Yeon Min, Shrimai Prabhumoye, Yonatan Bisk, Russ R Salakhutdinov, Amos Azaria, Tom M Mitchell, and Yuanzhi Li. 2024. Spring: Studying papers and reasoning to play games. *Advances in Neural Information Processing Systems* 36 (2024).

[111] Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. 2021. An Explanation of In-context Learning as Implicit Bayesian Inference. In *International Conference on Learning Representations*.

[112] Xue Yan, Jiaxian Guo, Xingzhou Lou, Jun Wang, Haifeng Zhang, and Yali Du. 2023. An efficient end-to-end training approach for zero-shot human-AI coordination. *Advances in Neural Information Processing Systems* 36 (2023), 2636–2658.

[113] An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zhihao Fan. 2024. Qwen2 Technical Report. *arXiv preprint arXiv:2407.10671* (2024).

[114] Canjun Yang, Yuanchao Zhu, and Yanhu Chen. 2021. A review of human–machine cooperation in the robotics domain. *IEEE Transactions on Human-Machine Systems* 52, 1 (2021), 12–25.

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

[116] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. 2022. ReAct: Synergizing Reasoning and Acting in Language Models. In *The Eleventh International Conference on Learning Representations*.

[117] Chao Yu, Jiaxuan Gao, Weilin Liu, Botian Xu, Hao Tang, Jiaqi Yang, Yu Wang, and Yi Wu. 2023. Learning Zero-Shot Cooperation with Humans, Assuming Humans Are Biased. In *The Eleventh International Conference on Learning Representations*.

[118] Chao Yu, Akash Velu, Eugene Vinitsky, Jiaxuan Gao, Yu Wang, Alexandre Bayen, and Yi Wu. 2022. The surprising effectiveness of ppo in cooperative multi-agent games. *Advances in neural information processing systems* 35 (2022), 24611–24624.

[119] Haibao Yu, Wenxian Yang, Jiaru Zhong, Zhenwei Yang, Siqi Fan, Ping Luo, and Zaiqing Nie. 2025. End-to-end autonomous driving through V2X cooperation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 39. 9598–9606.

[120] Ceyao Zhang, Kaijie Yang, Siyi Hu, Zihao Wang, Guanghe Li, Yihang Sun, Cheng Zhang, Zhaowei Zhang, Anji Liu, Song-Chun Zhu, et al. 2024. Proagent: building proactive cooperative agents with large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38. 17591–17599.

[121] Dan Zhang, Sining Zhoubian, Ziniu Hu, Yisong Yue, Yuxiao Dong, and Jie Tang. 2024. Rest-mcts*: Llm self-training via process reward guided tree search. *Advances in Neural Information Processing Systems* 37 (2024), 64735–64772.

[122] Jiajun Zhang, Zeyu Cui, Jiaxi Yang, Lei Zhang, Yuheng Jing, Zeyao Ma, Tianyi Bai, Zilei Wang, Qiang Liu, Liang Wang, et al. 2026. From Completion to Editing:

Unlocking Context-Aware Code Infilling via Search-and-Replace Instruction Tuning. *arXiv preprint arXiv:2601.13384* (2026).

[123] Jenny Zhang, Joel Lehman, Kenneth Stanley, and Jeff Clune. 2023. OMNI: Open-endedness via Models of human Notions of Interestingness. In *The Twelfth International Conference on Learning Representations*.

[124] Jiajun Zhang, Jianke Zhang, Zeyu Cui, Jiaxi Yang, Lei Zhang, Binyuan Hui, Qiang Liu, Zilei Wang, Liang Wang, and Junyang Lin. 2025. PlotCraft: Pushing the Limits of LLMs for Complex and Interactive Data Visualization. *arXiv preprint arXiv:2511.00010* (2025).

[125] Ruiqi Zhang, Spencer Frei, and Peter L Bartlett. 2023. Trained Transformers Learn Linear Models In-Context. *arXiv preprint arXiv:2306.09927* (2023).

[126] Mingde Zhao, Zhen Liu, Sitao Luan, Shuyuan Zhang, Doina Precup, and Yoshua Bengio. 2021. A consciousness-inspired planning agent for model-based reinforcement learning. *Advances in neural information processing systems* 34 (2021), 1569–1581.

[127] Rui Zhao, Jinming Song, Yufeng Yuan, Haifeng Hu, Yang Gao, Yi Wu, Zhongqian Sun, and Wei Yang. 2023. Maximum entropy population-based training for zero-shot human-ai coordination. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 37. 6145–6153.

[128] Zirui Zhao, Wee Sun Lee, and David Hsu. 2023. Large language models as commonsense knowledge for large-scale task planning. *Advances in Neural Information Processing Systems* 36 (2023), 31967–31987.

[129] Xizhou Zhu, Yuntao Chen, Hao Tian, Chenxin Tao, Weijie Su, Chenyu Yang, Gao Huang, Bin Li, Lewei Lu, Xiaogang Wang, et al. 2023. Ghost in the minecraft: Generally capable agents for open-world environments via large language models with text-based knowledge and memory. *arXiv preprint arXiv:2305.17144* (2023).