Scalable and Safe Multi-Agent Coordination with Reconstructed Level-k Monte Carlo Tree Search

Zhihao Lin
University of Glasgow
Glasgow, United Kingdom
2800400L@student.gla.ac.uk

Lin Wu
University of Glasgow
Glasgow, United Kingdom
3058596w@student.gla.ac.uk

Zhen Tian
University of Glasgow
Glasgow, United Kingdom
2620920Z@student.gla.ac.uk

Alessio Lomuscio
Imperial College London
London, United Kingdom
a.lomuscio@imperial.ac.uk

Jianglin Lan
University of Glasgow
Glasgow, United Kingdom
Jianglin.Lan@glasgow.ac.uk

ABSTRACT

Multi-agent coordination without central control requires balancing safety and computational efficiency. We present a novel framework that transforms Level-k cognitive reasoning from a descriptive model of bounded rationality into a constructive planning algorithm for agent coordination. Our key insight is to replace Level-k reasoning's assumption of random Level-0 behavior with safety-oriented baselines where all agents compute conservative trajectories. Safety emerges naturally from the recursive structure: each reasoning level inherits and strengthens the safety margins of lower levels, creating cascading conservatism that prevents collisions without explicit constraints. Beyond ensuring safety, this hierarchical conservatism also provides a natural foundation for efficient planning. By integrating this reconstructed hierarchy with Monte Carlo Tree Search (MCTS), we achieve significant computational advantages through two complementary mechanisms: a Dynamic Interaction Graph that constrains candidate interactions and reduces complexity from exponential to linear in agent count, and Safety-aware Pruning within MCTS that eliminates infeasible actions before evaluation. We evaluate our framework on symmetric multi-agent intersections, demonstrating collision-free coordination and real-time efficiency across scenarios of varying complexity, highlighting its scalability and robustness for safety-critical planning.

KEYWORDS

Multi-agent coordination; cognitive hierarchy; Monte Carlo tree search; game-theoretic planning; safety-critical systems.

ACM Reference Format:
Zhihao Lin, Lin Wu, Zhen Tian, Alessio Lomuscio, and Jianglin Lan. 2026. Scalable and Safe Multi-Agent Coordination with Reconstructed Level-k Monte Carlo Tree Search. In *Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), Paphos, Cyprus, May 25 – 29, 2026*, IFAAMAS, 9 pages. https://doi.org/10.65109/TAVV6081

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/TAVV6081

1 INTRODUCTION

Coordinating autonomous agents in symmetric scenarios is a key challenge in multi-agent systems [27]. Consider eight agents simultaneously approaching an intersection from equidistant positions. Perfect symmetry eliminates natural priorities, creating a coordination dilemma: without principled mechanisms, agents face deadlock or collision, while the exponential joint action space renders centralized optimization intractable [35]. Such symmetric configurations arise in practice, including urban intersections, parking facilities, and warehouse automation systems [29].

The complexity of multi-agent coordination grows dramatically with scale [9]. Consider an N-agent intersection, where each agent has a discrete action space $\mathcal{A}$ with $|\mathcal{A}|$ possible actions. For $N = 8$ and $|\mathcal{A}| = 15$, the joint action space contains $15^8 \approx 2.5 \times 10^9$ possibilities per time step [19]. Over a 10-step planning horizon, this yields $15^{80} \approx 10^{94}$ potential trajectories, far beyond practical computational limits for real-time systems [28]. This exponential explosion, known as the curse of dimensionality, renders exhaustive search infeasible and challenges even approximate methods [17].

Existing approaches to multi-agent coordination including game-theoretic methods [25, 34], sampling-based planners [2, 4], and learning-based techniques [31, 37], can be effective in certain settings, but they often struggle with symmetric interactions, scalability, or safety guarantees. In particular, classical Level-k reasoning models [27] assume Level-0 agents behave randomly or naively. While this assumption provides computational tractability, it renders Level-k reasoning unsuitable for safety-critical scenarios, as naive Level-0 behavior can easily lead to collisions.

We address these limitations by reconceptualizing Level-k reasoning, transforming it from a descriptive model of human bounded rationality into a constructive algorithm for multi-agent coordination. Instead of assuming naive Level-0 behavior, we define Level-0 as a conservative safety baseline that generates collision-free trajectories. Higher-level agents recursively best-respond to these safe lower-level trajectories, decomposing the exponential joint optimization problem into sequential single-agent Markov Decision Processes (MDPs) [1]. This reduces computational complexity from $O(|\mathcal{A}|^N)$ to $O(N \cdot |\mathcal{A}|)$ while preserving safety through cascading conservatism across reasoning levels.

Building on the computationally tractable Level-k hierarchy, our framework integrates Monte Carlo Tree Search (MCTS) [4] to achieve efficient and scalable coordination. Safety-aware pruning

reduces unnecessary exploration, while spatial-strategic filtering focuses on relevant agents for interactions. Together, these mechanisms enable real-time planning, robustly handling diverse scenarios. The framework dynamically adjusts reasoning depth based on interaction complexity, ensuring safe and coordinated behavior without the need for scenario-specific reconfiguration.

We evaluate our framework on scenarios that represent some of the most challenging decentralized coordination problems, involving increasing agent counts and complex interaction patterns. Our Level-k framework consistently outperforms baselines such as vanilla MCTS. Beyond standard performance metrics, it provides interpretable decisions through explicit reasoning traces, crucial for safety-critical deployment. By repurposing cognitive models of human limitations as computational tools for robots, we open new directions for bridging behavioral economics and autonomous systems. Our main contributions are as follows.

(1) We put forward a reconceptualized hierarchical cognitive framework that transforms joint optimization into tractable sequential single-agent MDPs, reducing complexity from $O(|\mathcal{A}|^N)$ to $O(N \cdot |\mathcal{A}|)$ while ensuring safety through cascading conservatism.
(2) We show that the proposed MCTS-Level-k approach achieves sub-100ms planning through safety-aware pruning and dual spatial-strategic filtering, making it suitable for interpretable, real-time multi-agent coordination.
(3) We evaluate the approach experimentally on symmetric multi-agent intersections demonstrating that our framework consistently outperforms vanilla MCTS and game-theoretic methods, achieving superior safety, efficiency, and scalability across scenarios with varying agent counts.

2 RELATED WORK

Game-theoretic approaches.

Game-theoretic frameworks provide principled solutions for multi-agent coordination through Nash equilibrium computation [25, 34, 39], ensuring that no agent can improve its outcome via unilateral deviation [6]. However, even when equilibria exist, they may not be discoverable or computable in practice [8]. Computing equilibria requires exploring the full joint action space, which quickly becomes intractable beyond 2–3 agents [10]. Symmetric scenarios exacerbate the problem, as multiple equilibria exist and additional selection mechanisms may be needed, often conflicting with safety objectives [14]. Hierarchical game solvers [7, 15] decompose problems into subgames but remain limited by the exponential growth of joint action spaces [20].

Sampling-based methods. Sampling-based planning, particularly Monte Carlo Tree Search (MCTS), reduces computational burden by selectively exploring promising branches [2, 4]. Upper confidence bound heuristics guide search toward high-value actions [38]. Yet, vanilla MCTS typically assumes static or randomly acting opponents [18], resulting in optimistic predictions that fail under multi-agent interactions [26]. Extensions incorporating opponent modeling either require centralized coordination [21, 27] or scale poorly, as maintaining beliefs over all agents' policies becomes infeasible with increasing agent numbers.

Learning-based methods. Multi-agent reinforcement learning (MARL) enables agents to learn coordination strategies from simulation [31, 37], discovering emergent behaviors without explicit communication. However, learned policies often lack generalization: architectures and state representations tuned for small agent sets fail in larger or dynamically varying configurations [11, 12]. Retraining for each scenario is required, limiting applicability in real-time or open-world deployments [5].

Level-k reasoning. Level-k reasoning from behavioral economics models bounded rationality [27], providing a middle ground between optimal but intractable game-theoretic solutions and naïve sampling or learning approaches. Agents recursively predict others' actions based on lower-level reasoning [22, 32], breaking symmetry and decomposing joint optimization into sequential single-agent problems [36]. Classical Level-k assumes Level-0 agents act randomly, which is unsafe for critical scenarios and insufficient for real-time performance [23]. This motivates extensions integrating conservative initialization, predictive models, and scalable planning, forming the basis of our MCTS-Level-k framework.

3 PROBLEM FORMULATION

We consider a symmetric multi-agent coordination problem where N agents simultaneously approach an unsignalized intersection from four directions. Each direction has two lanes, with agents placed equidistant from the center, creating a perfectly symmetric configuration. This setup poses three key challenges: First, the lack of natural ordering leads to simultaneous conflicts at the intersection center. Second, the joint action space scales as $O(|\mathcal{A}|^N)$, where $\mathcal{A}$ is the action space of a single agent, making centralized optimization intractable. Third, the symmetry increases deadlock risk, as no agent has inherent priority—agents may either all yield or move simultaneously, risking collision.

To address these challenges, we adopt a decentralized framework where each agent plans locally based on its observations and limited communication. Decision making follows a recursive Level-k reasoning process where Level-k agents anticipate the behaviors of agents with lower reasoning levels ($< k$). Reasoning levels are assigned dynamically based on interaction complexity. *Level-0 initializes each agent's conservative baseline trajectory using constant-velocity predictions for other agents, instead of the random or rule-based behaviors assumed in classical formulations [3]. This predictable baseline provides a safe foundation for higher-level reasoning, enabling cascading safety through the recursive Level-k structure.* For a Level-k agent i ($k \in \{1, 2\}$), the planning problem is formulated as:

$$\begin{aligned} \mathbf{a}_i^{(k)*}(t) := \arg\max_{\mathbf{a}_i(t)} \mathcal{R}_i^{(k)}(\mathbf{a}_i(t) \mid S(t), \mathbf{a}_{-i}^{(<k)}(t)) \\ \text{s.t.} \quad \mathbf{a}_i(t) \in \mathcal{A}^H,\ s_i(t+j) \in \mathcal{S}_{\text{safe}}^i,\ \forall j \in [0, H] \end{aligned}, \tag{1}$$

where $\mathbf{a}_i^{(k)*}(t)$ denotes the optimal solution under Level-k reasoning, $\mathbf{a}_i(t) = \{a_i(t), ..., a_i(t+H-1)\}$ is the action sequence over horizon H, $S(t)$ is the global state, $\mathcal{S}_{\text{safe}}^i$ denotes the safe state space for agent i, $\mathcal{A}^H$ is the H-step action space, and $\mathbf{a}_{-i}^{(<k)}(t)$ represents actions of lower-level agents. The reward function $\mathcal{R}_i^{(k)}(\cdot)$ encodes safety, efficiency, and comfort objectives. Detailed state representation and dynamics are formalized in Section 4.

4 AGENT MODELING AND INTERACTIONS

In this section, we present the mathematical formulation of our agent modeling and interaction framework, which serves as the foundation for the MCTS-Level-k framework.

4.1 State, Dynamics, and Collision Modeling

The state of agent i at time t is defined as:

$$s_i(t) = [x_i(t), y_i(t), v_i(t), \psi_i(t)]^\top \in \mathcal{S}_i \subset \mathbb{R}^4, \tag{2}$$

where $(x_i(t), y_i(t))$ is the position, $v_i(t) \in [0, v_{\max}]$ the velocity, and $\psi_i(t) \in [-\pi, \pi]$ the heading angle. The global state is $S(t) = \{s_1(t), \ldots, s_N(t)\} \in \mathcal{S}^N$. Each agent selects from a discrete action space $\mathcal{A}_i$ consisting of 15 control primitives, combining longitudinal acceleration $a_i(t)$ and angular velocity $\delta_i(t)$. This size balances expressiveness and computational tractability, providing sufficient maneuver options while keeping planning efficient.

We adopt the following point-mass kinematics model as it is widely used for agent decision making [16]:

$$x_i(t+1) = x_i(t) + v_i(t)\cos(\psi_i(t))\Delta t, \tag{3a}$$

$$y_i(t+1) = y_i(t) + v_i(t)\sin(\psi_i(t))\Delta t, \tag{3b}$$

$$v_i(t+1) = v_i(t) + a_i(t)\Delta t, \tag{3c}$$

$$\psi_i(t+1) = \psi_i(t) + \delta_i(t)\Delta t, \tag{3d}$$

where Δt is the discrete time step.

To ensure safe operation, each agent's feasible states satisfy three requirements: (i) collision avoidance, (ii) road boundary adherence, and (iii) longitudinal speed limits. In practice, each agent is represented as an oriented rectangle of length l_i and width w_i, extended to $l_i + \Delta l$ and $w_i + \Delta w$ to account for a safety margin. Given an agent state $s_i(t)$, the four vertices of the occupied rectangle are

$$\mathbf{v}_i^j = (x_i(t), y_i(t)) + \mathbf{R}(\psi_i(t)) \cdot \mathbf{p}^j, \quad j \in \{1, 2, 3, 4\}, \tag{4}$$

where $(x_i(t), y_i(t))$ is the center position, $\mathbf{R}(\psi_i(t))$ is the rotation matrix, and $\mathbf{p}^j$ are the corner offsets. Collisions between agents i and j are then checked via the Separating Axis Theorem (SAT) [33]:

$$\text{Collision}(i, j) = \text{SAT}(\mathbf{V}_i, \mathbf{V}_j), \quad \mathbf{V}_i = \{\mathbf{v}_i^1, \mathbf{v}_i^2, \mathbf{v}_i^3, \mathbf{v}_i^4\}. \tag{5}$$

The agent polygon must remain within the drivable area and respect speed bounds. Combining these with collision avoidance, the safe state space is defined as

$$\mathcal{S}_{\text{safe}}^i = \left\{ s_i \in \mathcal{S}_i \;\middle|\; \begin{array}{l} \text{hull}(\mathbf{V}_i(s_i)) \subset \mathcal{R}_{\text{valid}}, \\ v_i \in [0, v_{\max}], \\ \text{Collision}(i, j) = \text{false}, \forall j \neq i \end{array} \right\}, \tag{6}$$

where $\text{hull}(\mathbf{V}_i(s_i))$ denotes the convex hull (i.e., rectangular polygon) formed by the four vertices $\mathbf{V}_i = \{\mathbf{v}_i^1, \mathbf{v}_i^2, \mathbf{v}_i^3, \mathbf{v}_i^4\}$ at state s_i, and $\mathcal{R}_{\text{valid}}$ represents the drivable road area.

4.2 Dynamic Interaction Graph

To address the scalability challenge, we propose a dynamic interaction graph with improved computational efficiency by considering only strategically and spatially relevant interactions, using a filtering mechanism based on spatial proximity and reasoning levels.

Time-varying graph formulation. As agents move through the intersection, their spatial relationships evolve dynamically: interaction edges appear when agents approach potential conflict zones and disappear once conflicts are resolved. We represent this as a time-varying directed graph $G^t = (A, E^t, W^t)$, where $A = \{1, 2, \ldots, N\}$ is the set of agents, $E^t \subset A \times A$ is the time-varying edge set at time t, and $W^t = [w_{ij}^t]$ is the weighted adjacency matrix.

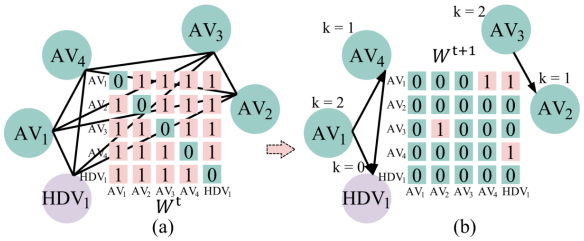


Figure 1: Dynamic interaction graph: (a) Complete graph showing all potential interactions. (b) Filtered graph at time t with active edges in E^t (shown by non-zero entries in W^t).

Spatial Filtering. An interaction edge $(j, i) \in E^t$ exists if and only if agent j influences the decision-making of agent i at time t, determined by analyzing potential conflicts within a prediction horizon:

$$(j, i) \in E^t \iff \exists t' > t : \text{Conflict}(i, j, t') = 1. \tag{7}$$

The conflict between agents i and j is detected by analyzing the intersection of their predicted future trajectories as follows:

$$\text{Conflict}(i, j, t') = \begin{cases} 1, & \text{if } (\mathcal{B}_i^{t'} \cap \mathcal{B}_j^{t'} \neq \emptyset) \\ & \quad \wedge (|t_i^{\text{arr}} - t_j^{\text{arr}}| < \tau), \\ 0, & \text{otherwise} \end{cases} \tag{8}$$

where $\mathcal{B}_i^{t'}$ represents the predicted boundary of agent i at time t', t_i^{arr} is the arrival time of agent i at the potential conflict point, and τ is a temporal threshold for collision risk.

Strategic Filtering. To reduce planning complexity, we introduce strategic filtering within the Level-k reasoning hierarchy. Agents are dynamically assigned to operational Levels 1–2 based on interaction complexity, including proximity to conflict zones, traffic density, and trajectory conflicts. Note that Level-0 is not an actual agent type but an initialization procedure that generates conservative baseline trajectories. For each active level k, we collect all agents assigned to that level into the set

$$\mathcal{L}_k = \{i \in A : k_i = k\}, \; k \in \{1, 2\}. \tag{9}$$

Here, A is the set of all agents, and k_i denotes the reasoning level of agent i. The set $\mathcal{L}_k$ provides the pool of agents at level k from which each agent i forms its level-specific interaction set $\mathcal{N}_i^{(k)}$, selecting which agents influence its reasoning based on the dual filtering:

$$\mathcal{N}_i^{(k)} = \begin{cases} \emptyset, & \text{if } k = 0 \\ \{j \in A : (k_j < k) \wedge ((j, i) \in E^t)\}, & \text{if } k \in \{1, 2\} \end{cases}. \tag{10}$$

The first case ($k = 0$) is included for completeness, though Level-0 agents do not perform strategic planning. The resulting interaction graph (Fig. 1) includes only Levels 1 and 2, where higher-level agents consider lower-level ones for anticipatory planning. This dual filtering significantly reduces computational complexity. Specifically, for each Level-k agent i, the complexity reduces from $O(|\mathcal{A}_i|^N)$ to

$O(|\mathcal{A}_i|^{|\mathcal{N}_i^{(k)}|})$, where $|\mathcal{N}_i^{(k)}| \ll N$ represents the number of interaction neighbors. For instance, in Fig. 1, Level-2 agent A_1 considers only Level-1 agents A_4 and A_5, reducing the action space while preserving solution quality.

4.3 Multi-Agent Interaction Model

To solve the optimization problem as defined in (1), we formulate the multi-agent coordination as a partially observable stochastic game. For each agent i with Level-k reasoning, the cumulative reward over a finite horizon H is:

$$\mathcal{R}_i^{(k)}(\mathbf{a}_i(t) \mid S(t), \mathbf{a}_{-i}^{(<k)}(t)) = \sum_{j=0}^{H-1} \gamma^j \cdot r_i(S(t+j), a_i(t+j), a_{-i}(t+j)), \quad \gamma^j \in (0,1] \tag{11}$$

where $a_i(t+j) \in \mathcal{A}_i$ denotes the action taken by agent i at time $t+j$, $a_{-i}(t+j) = \{a_m(t+j)\}_{m \neq i}$ denotes the joint actions of all other agents, and $r_i(\cdot)$ is the instantaneous reward function of agent i. The state evolution follows the system dynamics (3).

The reward function that encodes task-specific priorities including safety, efficiency, and comfort, is structured as:

$$r_i(S(t), a_i(t), a_{-i}(t)) = w_s \cdot r_s^i(S(t)) + w_e \cdot r_e^i(s_i(t)) + w_d \cdot r_d^i(s_i(t)) + w_c \cdot r_c^i(a_i(t)), \tag{12}$$

where $w_\cdot$ represent weights balancing safety, efficiency, trajectory tracking, and comfort objectives. The safety reward $r_s^i(S(t))$ encourages maintaining safe distances from other agents and road boundaries:

$$r_s^i(S(t)) = -\sum_{j \neq i} \exp(-d_{ij}^2/2d_{\text{safe}}^2) - \sum_{b \in \mathcal{B}_r} \exp(-d_{ib}^2/2d_{\text{safe}}^2), \tag{13}$$

where d_{ij} is the inter-agent distance computed via SAT (5), d_{ib} is the distance to the road boundary $b \in \mathcal{B}_r$, and d_{safe} is the safety threshold. The efficiency reward encourages maintaining the reference velocity:

$$r_e^i(s_i(t)) = -|v_{\text{ref}} - v_i(t)|, \tag{14}$$

where v_{ref} is the reference velocity for traffic flow. The trajectory tracking reward encourages following the reference path:

$$r_d^i(s_i(t)) = -\min_{p \in \mathcal{P}_i} \|[x_i(t), y_i(t)]^T - p\|_2, \tag{15}$$

where $\mathcal{P}_i$ denotes the set of reference path points for agent i, and $p \in \mathcal{P}_i$ represents an individual point. The comfort reward discourages abrupt control changes:

$$r_c^i(a_i(t)) = -\|a_i(t) - a_i(t-1)\|_2^2. \tag{16}$$

These components collectively define each agent's objectives. However, joint optimization remains intractable due to the exponential growth of the joint action space $\prod_{i=1}^N \mathcal{A}_i$ or simply $\mathcal{A}^N$.

4.4 Level-k Solution Framework

To address the computational intractability of joint optimization, we employ the Level-k reasoning framework, which transforms the multi-agent problem into a hierarchy of tractable subproblems through bounded rationality assumptions. Building upon the interaction structure established in Section 4.2, we now formalize how each agent determines its optimal trajectory.

4.4.1 Recursive Best Response. The key insight of our Level-k framework is that each agent with reasoning level k treats all lower-level agents as predictable, effectively converting the intractable multi-agent game into a single-agent optimization problem. Based on the reward functional in (11) and the interaction sets $\mathcal{N}_i^{(k)}$ defined in (10), each Level-k agent ($k \in \{1,2\}$) solves a recursive best-response problem:

$$\mathbf{a}_i^{(k)*}(t) = \arg\max_{\mathbf{a}_i(t)} \mathcal{R}_i^{(k)}(\mathbf{a}_i(t) \mid S(t), \mathbf{a}_{-i}^{(<k)}(t)), \quad \text{s.t.} \quad (3),\ (6),\ \mathbf{a}_i(t) \in \mathcal{A}^H,\ s_i(t+j) \in \mathcal{S}_{\text{safe}}^i, \forall j \in [0,H] \tag{17}$$

where the predicted behaviors $\mathbf{a}_{-i}^{(<k)}(t)$ are obtained recursively: Level-1 agents assume all others follow Level-0 trajectories (treating others as static obstacles), while Level-2 agents anticipate Level-1 behaviors. This recursive structure breaks symmetry through reasoning-level differentiation, creating an implicit ordering for conflict resolution. Since Level-0 generates conservative baselines rather than random behaviors, each subsequent level inherits and amplifies these safety buffers, resulting in system-wide robustness to prediction errors without requiring explicit coordination.

4.4.2 Induced Single-Agent MDP Formulation. Given fixed opponent strategies $\mathbf{a}_{-i}^{(<k)}$ from lower-level agents within the interaction set $\mathcal{N}_i^{(k)}$, agent i's planning problem reduces to an induced MDP:

$$\mathcal{M}_i^{(k)} = (\mathcal{S}_i, \mathcal{A}_i, P_i^{(k)}, r_i^{(k)}, H), \tag{18}$$

where H is the planning horizon, and $r_i^{(k)}$ is the induced reward function that incorporates predicted opponent behaviors. The induced dynamics are obtained by marginalizing over predicted opponent actions:

$$P_i^{(k)}(s_i'|s_i, a_i) = \sum_{a_{-i}} \hat{\pi}_{-i}^{(<k)}(a_{-i}|s_{-i}) P(s_i'|s_i, a_i, a_{-i}), \tag{19}$$

where s_i' is the next state of agent i, and $\hat{\pi}_{-i}^{(<k)}$ represents the predicted policies of lower-level agents. The corresponding induced reward function is:

$$r_i^{(k)}(s_i, a_i) = \sum_{a_{-i}} \hat{\pi}_{-i}^{(<k)}(a_{-i}|s_{-i}) r_i(S, a_i, a_{-i}), \tag{20}$$

where $S = \{s_i\} \cup \{s_j : j \in \mathcal{N}_i^{(k)}\}$ represents the joint state of agent i and its interaction neighbors defined in (10), and $r_i(S, a_i, a_{-i})$ is the instantaneous reward from (12). This hierarchical decomposition reduces computational complexity from $O(|\mathcal{A}_i|^N)$ to $O(|\mathcal{A}_i|^{|\mathcal{N}_i^{(k)}|})$ by transforming multi-agent planning into a single-agent MDP under fixed opponent strategies [30], enabling standard methods like MCTS to achieve real-time planning.

5 MCTS-LEVEL-K DECISION-MAKING

To enable tractable planning under recursive reasoning, we adopt Monte Carlo Tree Search (MCTS) as a sampling-based solver for the Level-k optimization problem defined in (17).

5.1 MCTS-Level-k Optimization

Building on the induced MDP formulation (Section 4.4), MCTS performs a forward search to find optimal action sequences, as illustrated in Fig. 2, by directly maximizing the cumulative reward

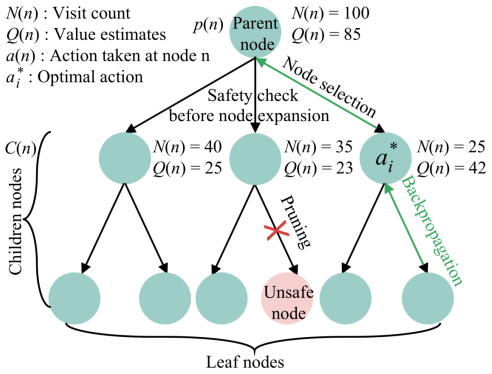


Figure 2: Illustration of the safety-aware MCTS framework.

defined in (11). For each Level-k agent i, we first define the state-action value function for a single action:

$$Q(s,a) = \mathbb{E}\Big[\sum_{j=0}^{H-1} r_i\big(S(t+j), a_i(t+j), a_{-i}(t+j)\big) \,\Big|\, s_0 = s,\ a_0 = a\Big], \quad (21)$$

where $r_i(\cdot)$ is the instantaneous reward from (12). To evaluate action sequences, we define the sequence value function:

$$V(\mathbf{a}_i \mid S(t), \mathbf{a}_{-i}^{(<k)}) = \sum_{j=0}^{H-1} r_i\big(S(t+j),\ a_i(t+j),\ a_{-i}^{(<k)}(t+j)\big), \quad (22)$$

where $\mathbf{a}_i = \{a_i(t), \ldots, a_i(t+H-1)\}$ denotes the action sequence over horizon H, and $\mathbf{a}_{-i}^{(<k)}$ represents the predicted strategies of Level-$(<k)$ opponents in $\mathcal{N}_i^{(k)}$. MCTS then solves the following optimization problem:

$$\mathbf{a}_i^{(k)*} = \arg\max_{\mathbf{a}_i \in \mathcal{A}^H} V(\mathbf{a}_i \mid S(t), \mathbf{a}_{-i}^{(<k)}). \quad (23)$$

In the receding horizon framework, only the first action of the optimal sequence $\mathbf{a}_i^{(k)*}$, i.e., $a_i^{(k)*}(t)$, is executed, and planning is repeated at each subsequent time step.

5.2 Tree Structure and Action Selection

The search tree $\mathcal{T}_i$ for agent i consists of nodes representing state-action pairs. Each node $n \in \mathcal{T}_i$ maintains:

$$n := \{s(n), a(n), d(n), Q(n), N(n), C(n)\}, \quad (24)$$

where $s(n)$ is the state, $a(n)$ is the action leading to this node, $d(n)$ is the depth, $Q(n)$ and $N(n)$ are the node-level value estimate and visit count, $C(n)$ is the set of child nodes, and for each edge (n,a) we additionally record action-specific statistics $Q(n,a)$ and $N(n,a)$.

During the selection phase, the algorithm traverses the tree from root to a leaf node using *the Upper Confidence Bound applied to Trees* (UCT) [13] criterion:

$$a^* = \arg\max_{a \in \mathcal{A}_{\text{safe}}(n)} \left\{ \frac{Q(n,a)}{N(n,a)} + c\sqrt{\frac{\ln N(n)}{N(n,a)}} \right\}, \quad (25)$$

where c is the exploration constant. Importantly, the maximization is restricted to the safe action set

$$\mathcal{A}_{\text{safe}}(n) = \{a \in \mathcal{A}_i : \Phi_{\text{safe}}(s(n), a) = \text{true}\}, \quad (26)$$

where Φ_{safe} is a Boolean predicate that returns `true` only if state feasibility constraints (6) are satisfied. This early pruning reduces the search space without compromising optimality, as unsafe trajectories cannot belong to any feasible solution.

5.3 Rollout with Level-k Predictions

From each leaf node reached during tree expansion, we perform rollouts to estimate future rewards. The rollout value from a leaf node with state s_{leaf} is:

$$V_{\text{rollout}}(s_{\text{leaf}}) = \sum_{h=0}^{H-d} \gamma^h r_i(S(t_{\text{leaf}}+h), a_i(t_{\text{leaf}}+h), \hat{a}_{-i}(t_{\text{leaf}}+h)), \quad (27)$$

where s_{leaf} is the state at the leaf node, t_{leaf} is the corresponding time step, d is the current tree depth, $\gamma \in (0,1]$ is the discount factor, $a_i(t_{\text{leaf}}+h) \sim \pi_{\text{default}}$ follows a default policy, while $\hat{a}_{-i}(t_{\text{leaf}}+h)$ are sampled from predicted Level-$(<k)$ strategies.

5.4 Value Backup and Action Selection

After each simulation, the rollout return $V_{\text{new}} = V_{\text{rollout}}(s)$ from the reached leaf node is propagated back through the traversed path:

$$Q(n) \leftarrow Q(n) + \frac{V_{\text{new}} - Q(n)}{N(n)+1}, \quad N(n) \leftarrow N(n)+1. \quad (28)$$

After K_{iter} iterations, the optimal action sequence is extracted by following the highest-value path from root to leaves:

$$\mathbf{a}_i^{(k)*} = \{a^*(n_0), a^*(n_1), \ldots, a^*(n_{H-1})\}, \quad (29)$$

where $a^*(n_h) = \arg\max_{a \in C(n_h)} Q(n_h, a)$ for each depth h. Consistent with the receding horizon principle and (23), we execute only the first action $a^*(n_0)$, and the entire planning process repeats at the next time step with updated state information, enabling adaptive responses to the dynamic interaction environment.

6 EVALUATION

Setup. To validate the effectiveness and scalability of the proposed framework, we design representative scenarios with increasing agent counts (from four to eight), which induce progressively higher interaction complexity. These scenarios include symmetric intersections with dense crossing conflicts as well as heterogeneous maneuvers that create asymmetric and tightly coupled interactions.
Baseline. We compare our method against several advanced optimization algorithms, including the Stackelberg game [10], Nash equilibrium [8], and MCTS [24] approaches.
Metrics. To provide a quantitative comparison of different methods, we report standard metrics including collision rate (%), arrival rate (%), computation time (ms), and maximum iterations. These capture safety, efficiency, and computational cost at a coarse level. In addition, to assess coordination quality and robustness, we also analyze trajectory deviation, temporal evolution of agent states, and minimum distance distribution between agents. These metrics offer detailed insights into multi-agent interaction behaviors and the effectiveness of Level-k reasoning in handling complex scenarios.

6.1 Case 1: Four-Agent Left-Turn Scenario

Scenario description. This symmetric intersection scenario (Fig. 3) involves four agents approaching from orthogonal directions, each

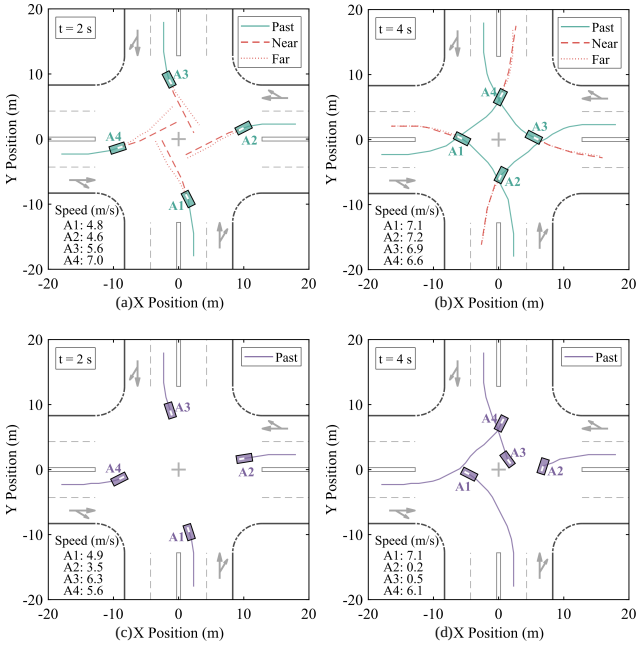


Figure 3: Comparison of agent coordination in Case 1. Top row (a-b): Our method shows smooth coordination with minimal trajectory deviation. Bottom row (c-d): Vanilla MCTS exhibits larger deviations and longer delays.

starting 18 m from the center and executing left turns. The setup creates dense crossing conflicts at the intersection center, requiring sophisticated coordination.

Qualitative comparison. Figure 3 compares the temporal evolution of our MCTS-Level-k framework (top) and vanilla MCTS (bottom). At $t = 2$ s, our method shows anticipatory coordination: A3 and A4 maintain moderate speeds (5.6, 7.0 m/s) for upcoming left turns, while A1 and A2 adjust to 4.8 and 4.6 m/s to establish an implicit passing sequence. This results in conflict-free, coordinated trajectories through Level-k reasoning. In contrast, vanilla MCTS is reactive and inconsistent (4.9, 3.5, 6.3, 5.6 m/s), suggesting imminent conflicts. By $t = 4$ s, our method enables A1 and A2 to clear the intersection smoothly (7.1, 7.2 m/s), with A3 and A4 entering precisely (6.9, 6.6 m/s), achieving synchronized motion via cognitive hierarchy. Vanilla MCTS fails critically: A2 and A3 nearly stop (0.2, 0.5 m/s), while A1 proceeds at 7.1 m/s, causing risky differentials and potential deadlock. This contrast illustrates our framework's strength: recursive Level-k reasoning enables implicit, communication-free coordination in symmetric left-turn scenarios.

Table 1: Performance comparison in Case 1

Method	Collision Rate (%)	Travel Time (s)	Computation Time (ms)	Max. Iters
Stackelberg	18.2 ± 6.6	9.7 ± 2.2	54.4 ± 14.1	1000
Nash	14.1 ± 6.3	9.1 ± 1.7	74.6 ± 12.2	1000
MCTS	10.7 ± 5.2	8.5 ± 2.6	60.4 ± 17.5	1000
Ours	**0**	**5.3 ± 1.3**	**21.2 ± 6.3**	**300**

Quantitative results. Table 1 reports the performance of all methods in the left-turn task. Our MCTS-Level-k framework achieves 0% collision rate, significantly outperforming Stackelberg (18.2 ± 6.6%), Nash (14.1 ± 6.3%), and vanilla MCTS (10.7 ± 5.2%). It completes the maneuver in 5.3 ± 1.3s—about 40% faster than baselines (8.5–9.7s)—by breaking symmetry and establishing implicit passing orders. Despite using only 300 iterations, it runs in 21.2 ± 6.3 ms, reducing compute time by 65–71%. These results highlight the safety, efficiency, and practicality of the framework.

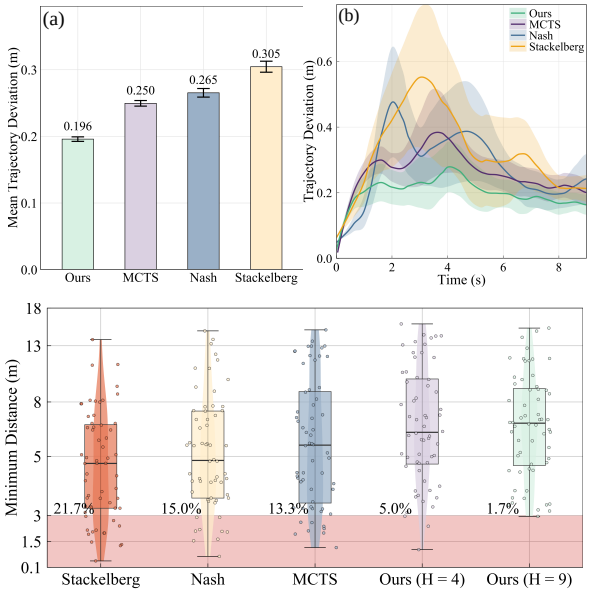


Figure 4: Statistical analysis in Case 1.

Statistical analysis. We conduct 10 trials to evaluate the performance and robustness of our framework in ensuring safe and efficient multi-agent coordination.

(i) Trajectory deviation (Fig. 4(a)) Our method yields the lowest mean deviation at 0.196 m, improving over vanilla MCTS (0.250 m, 22%), Nash (0.265 m, 26%), and Stackelberg (0.305 m, 36%). Stackelberg shows the worst performance due to its rigid leader-follower assumption breaking down under symmetric scenarios.

(ii) Temporal evolution (Fig. 4(b)) During the critical conflict phase ($t = 2$–5 s), our method (green) maintains deviations below 0.25 m, demonstrating stable coordination. In contrast, Nash (blue) and Stackelberg (orange) exhibit peaks up to 0.55 m around $t = 3$–4 s, indicating coordination breakdown, while vanilla MCTS peaks around 0.38 m. The shaded confidence intervals highlight the superior stability of our approach.

(iii) Minimum distance distribution (Fig. 4(c)) The minimum distance distribution measures each agent's closest approach to others, providing a direct safety indicator. With $H = 9$, our method yields only 1.7% near-collision events and a median separation of 7 m, demonstrating safe and consistent behavior. The $H = 4$ variant shows 5.0% violations. In contrast, Stackelberg exhibits 21.7% violations, while Nash and vanilla MCTS reach 15.0% and 13.3%, respectively. These results confirm that Level-k reasoning with sufficient horizon provides superior safety margins while maintaining coordination stability and efficiency.

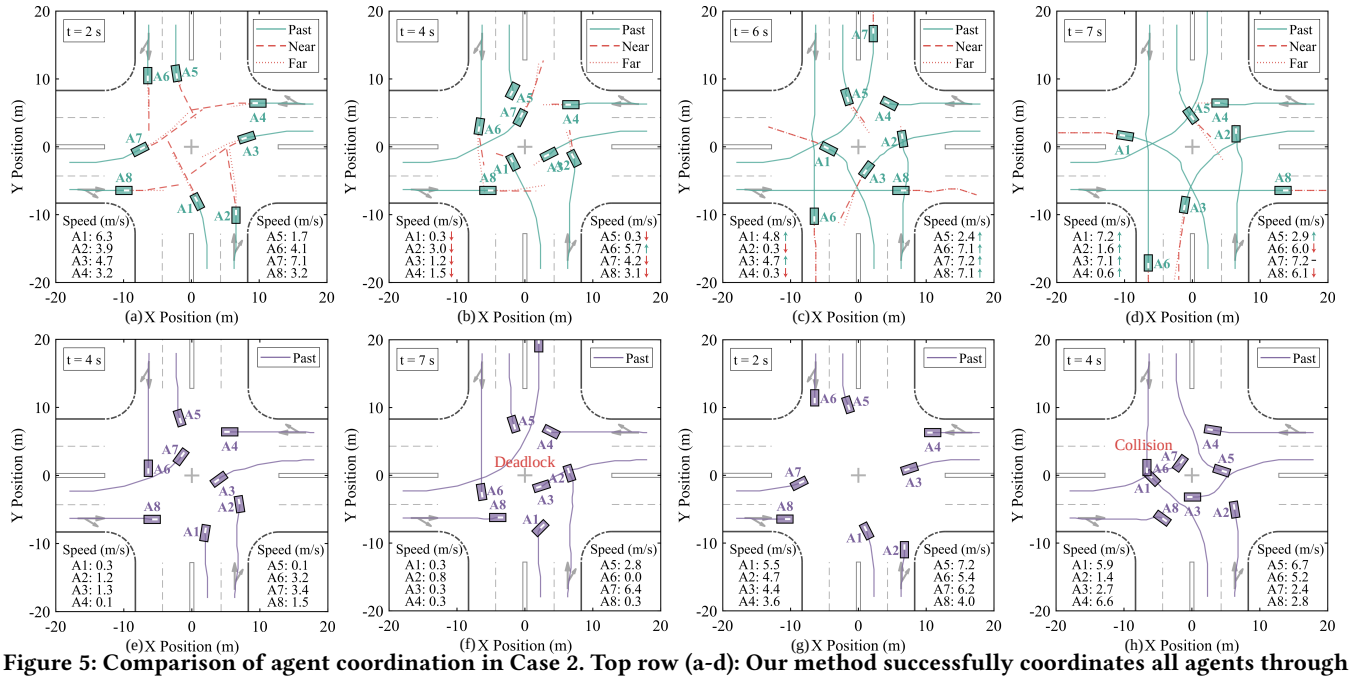


Figure 5: Comparison of agent coordination in Case 2. Top row (a-d): Our method successfully coordinates all agents through the intersection. Bottom row (e-h): Vanilla MCTS exhibits failure modes: deadlock (e-f) or collision (g-h).

Table 2: Performance comparison in Case 2: Heterogeneous Maneuvers in Symmetric Intersection

Method	Average Speed (m/s)	Trajectory Deviation (m)	Minimum Distance (m)	Collision Rate (%)	Arrival Rate (%)	Travel Time (s)	Computation Time (ms)	Maximum Iterations
Stackelberg	3.94 ± 1.78)	0.423 ± 0.418	5.439 ± 2.926	47.8 ± 21.8	34.7 ± 26.3	14.3 ± 2.3	72.23 ± 21.10	1000
Nash	4.17 ± 1.73	0.398 ± 0.354	5.069 ± 2.985	35.2 ± 16.8	53.1 ± 17.2	13.8 ± 2.2	89.23 ± 22.46	1000
MCTS	3.89 ± 1.65	0.266 ± 0.164	5.518 ± 3.204	21.6 ± 7.7	65.1 ± 10.6	13.2 ± 2.4	117.18 ± 23.24	1000
Ours	**5.20 ± 2.15**	**0.191 ± 0.113**	**6.804 ± 3.408**	**0**	**95.3 ± 4.1**	**9.8 ± 1.8**	**61.35 ± 12.79**	**300**

6.2 Case 2: Heterogeneous Maneuvers

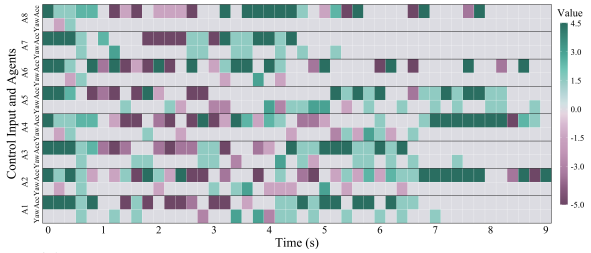
Scenario description. This scenario introduces maneuver diversity: inner-lane agents (A1, A3, A5, A7) turn left, while outer-lane agents (A2, A4, A6, A8) go straight. Despite geometric symmetry, this mix induces asymmetric crossings and tighter interaction timing, increasing the difficulty of coordination.

Qualitative comparison. Figure 5 compares coordination in heterogeneous scenarios with mixed turn and straight maneuvers. The Level-k framework (top row) enables proactive coordination: turning agents adapt trajectories to straight agents' intentions, while straight agents adjust speeds in anticipation. This mutual reasoning yields efficient passing patterns, for example, by $t = 4$ s, A5–A6 proceed while A1–A2 slow to 0.3 m/s, establishing safe sequences. Cascading safety margins from Level-0 initialization ensure turning agents complete 90° rotations by $t = 6$ s without blocking straight paths. In contrast, vanilla MCTS (bottom row) suffers from reactive limitations. Lacking anticipation, agents fail to form passing orders, causing deadlock (A3–A4 nearly stop at $t = 6$ s) or collisions (at $t = 4$ s). Its 21.6% failure rate versus our 0% highlights the value of cognitive hierarchy: Level-k reasoning turns joint optimization into tractable sequential planning, ensuring safety through conservative initialization rather than post hoc constraints.

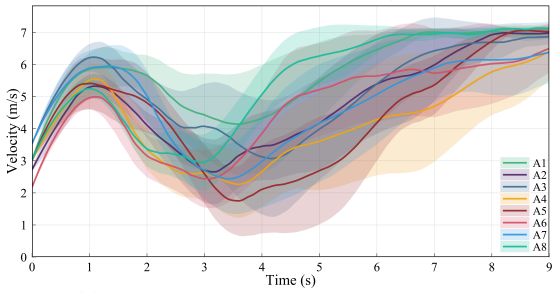
Quantitative results. Table 2 confirms robustness in heterogeneous scenarios: we achieve 0% collision rate and 95.3% arrival rate, significantly outperforming MCTS (21.6% collisions, 65.1% arrivals) and Stackelberg (47.8% collisions). Despite handling mixed left-turn and straight trajectories, our method maintains an average speed of 5.2 m/s with a computation time of 61.35 ms. This is 48% faster than vanilla MCTS while using 70% fewer iterations (300 compared to 1000), demonstrating both safety and efficiency in complex multi-agent coordination.

Statistical analysis. In this heterogeneous maneuvers scenario, we additionally examine control inputs and velocity profiles to reveal coordination strategies.

(i) Temporal dynamics and control analysis (Figure 6) Figure 6 illustrates coordination complexity under mixed maneuvers. The control heatmap (Figure 6(a)) shows temporal sequencing: A5–A7 begin decelerating at $t = 2$, s (-3 to -5, m/s^2) for early left turns, while A1–A3 delay to $t = 3$–4, s, forming natural separation. Straight-going agents (A2, A4, A6, A8) maintain moderate inputs, briefly



(a) Heatmap of control inputs for all agents over time.



(b) Velocity profiles of all agents over time.

Figure 6: Temporal dynamics and control analysis in Case 2.

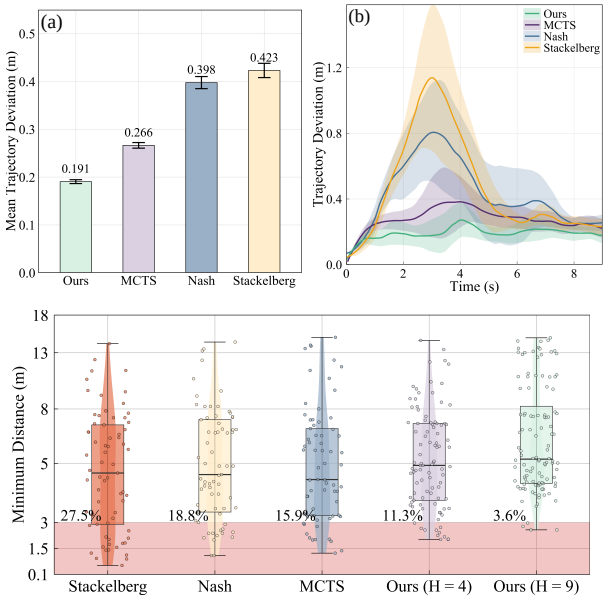


Figure 7: Statistical analysis in Case 2.

yielding when turning vehicles cross. Although 18% of actions exceed $3, \mathrm{m/s}^2$ (vs. 12% for uniform maneuvers), extremes remain under 5%, confirming anticipatory rather than reactive control. The velocity profiles (Figure 6(b)) further reflect this sequencing: turning agents decelerate to 2.0–2.5, m/s during 90° turns before recovering to 6–7, m/s, while straight-going agents sustain 4–5, m/s with minimal variation. Level-k reasoning thus naturally sequences conflicting trajectories, enabling collision-free navigation.

(ii) Trajectory deviation (Figure 7(a)) Our method achieves the lowest deviation at 0.191 m, improving over MCTS (0.266 m, 28%) and Stackelberg (0.423 m, 52%), despite the complexity of coordinating 90° turns with straight paths.

(iii) Temporal evolution (Figure 7(b)) The temporal profile shows our method maintains deviations below 0.4 m during the critical turning phase ($t = 3$–5 s), while Nash and Stackelberg spike to 1.0–1.2 m due to coordination failures.

(iv) Minimum distance distribution (Figure 7(c)) The minimum distance distribution further validates safety: our approach yields only 3.6% violations, compared to 15.9% (MCTS), 27.5% (Stackelberg), and 18.8% (Nash), with a median separation of 6 m. These results confirm the robustness of Level-k reasoning in mixed-trajectory coordination, achieving a 77% reduction in safety violations.

6.3 Complexity Analysis

To better illustrate the efficiency of our method, Table 3 contrasts its computational complexity with common baselines.

Joint optimization over $N = 8$ agents and horizon $H = 9$ requires exploring $O(|\mathcal{A}|^{NH}) \approx 10^{85}$ possible trajectories. Our approach achieves a 21-order complexity reduction through two key mechanisms: (i) Level-k decomposition (17) transforms the exponential joint problem into N sequential single-agent problems, each solving an induced MDP (18) with complexity $O(K_{\text{iter}} \cdot H)$; (ii) Safety-aware pruning based on constraints (6) removes $\sim 70\%$ of infeasible actions, reducing the effective branching factor to $b_{\text{eff}} \approx 4.5$. This

Table 3: Theoretical Computational Complexity Comparison

Method	Complexity Formula	Operations
Joint Optimization	$15^{8\times9}$	$\approx 10^{85}$
Game-Theoretic Nash	$O(15^{72} \cdot I_{\text{Nash}})$	$> 10^{87}$
Level-k (Exhaustive)	8×15^9	$\approx 3.1 \times 10^{11}$
Level-k + MCTS	$8 \times 300 \times 15 \times 9$	$\approx 3.2 \times 10^5$
Level-k + MCTS + Pruning	$8 \times 300 \times 4.5 \times 9$	$\approx 9.7 \times 10^4$

enables sub-100 ms planning cycles while maintaining optimality, as pruned unsafe actions cannot be part of optimal trajectories.

7 CONCLUSION

In this paper, we have shown that cognitive models originally developed to explain human bounded rationality can be repurposed as effective algorithms for multi-agent coordination. By reconceptualizing Level-k reasoning and replacing the traditional random Level-0 assumption with a safety-aware initialization, our framework allows safety to emerge organically through recursive reasoning rather than relying on explicit constraints. When integrated with MCTS, augmented with safety-aware pruning and dual filtering, the approach achieves real-time performance while maintaining structural safety properties. Our method reduces computational complexity by 81 orders of magnitude (from $\sim 10^{85}$ to $\sim 10^4$ operations), enabling sub-100ms planning. Experiments show that the framework delivers robust, interpretable coordination without requiring communication, adapts seamlessly to varying agent densities, and outperforms vanilla MCTS and game-theoretic methods.

The success of this approach suggests that other bounded rationality models may be repurposed for practical coordination. The framework currently assumes global observability. Future work will address partial observability, asynchronous planning, and mixed human-robot settings, where cognitive modeling can further enhance safety, interpretability, and scalability.

REFERENCES

[1] Guy Avni, Martin Kurečka, Kaushik Mallik, Petr Novotný, and Suman Sadhukhan. 2025. Bidding Games on Markov Decision Processes with Quantitative Reachability Objectives. In *Proceedings of the 24th International Conference on Autonomous Agents and Multiagent Systems* (Detroit, MI, USA) *(AAMAS '25)*. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 161–169.

[2] Lorenzo Bonanni, Daniele Meli, Alberto Castellini, and Alessandro Farinelli. 2025. Monte Carlo Tree Search with Velocity Obstacles for Safe and Efficient Motion Planning in Dynamic Environments. In *Proceedings of the 24th International Conference on Autonomous Agents and Multiagent Systems* (Detroit, MI, USA) *(AAMAS '25)*. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 371–380.

[3] Maxime Bouton, Alireza Nakhaei, David Isele, Kikuo Fujimura, and Mykel J Kochenderfer. 2020. Reinforcement learning with iterative reasoning for merging in dense traffic. In *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, IEEE, Rhodes Island, Greece, 1–6.

[4] C. B. Browne, E. Powley, D. Whitehouse, S. M. Lucas, et al. 2012. A Survey of Monte Carlo Tree Search Methods. *IEEE Trans. Comput. Intell. AI Games* 4, 1 (2012), 1–43. https://doi.org/10.1109/TCIAIG.2012.2186810

[5] Axel Brunnbauer, Julian Lemmel, Zahra Babaiee, Sophie A. Neubauer, and Radu Grosu. 2025. Scalable Offline Reinforcement Learning for Mean Field Games. In *Proceedings of the 24th International Conference on Autonomous Agents and Multiagent Systems* (Detroit, MI, USA) *(AAMAS '25)*. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 408–417.

[6] T-H. Hubert Chan, Qipeng Kuang, and Quan Xue. 2025. Game-Theoretically Secure Distributed Protocols for Fair Allocation in Coalitional Games. In *Proceedings of the 24th International Conference on Autonomous Agents and Multiagent Systems* (Detroit, MI, USA) *(AAMAS '25)*. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 463–471.

[7] Andrei Constantinescu and Roger Wattenhofer. 2025. Byzantine Game Theory: Sun Tzu's Boxes. In *Proceedings of the 24th International Conference on Autonomous Agents and Multiagent Systems* (Detroit, MI, USA) *(AAMAS '25)*. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 519–528.

[8] P. Hang et al. 2022. Driving Conflict Resolution of Autonomous Vehicles at Unsignalized Intersections: A Differential Game Approach. *IEEE/ASME Trans. Mechatron.* 27, 6 (2022), 5136–5146. https://doi.org/10.1109/TMECH.2022.3174273

[9] Rustam Galimullin, Maksim Gladyshev, Munyque Mittelmann, and Nima Motamed. 2025. Changing the Rules of the Game: Reasoning About Dynamic Phenomena in Multi-Agent Systems. In *Proceedings of the 24th International Conference on Autonomous Agents and Multiagent Systems* (Detroit, MI, USA) *(AAMAS '25)*. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 829–838.

[10] Peng Hang, Chao Huang, Zhongxu Hu, Yang Xing, and Chen Lv. 2021. Decision making of connected automated vehicles at an unsignalized roundabout considering personalized driving behaviours. *IEEE Trans. Veh. Technol.* 70, 5 (2021), 4051–4064.

[11] Shahab Karimi, Arash Karimi, and Ardalan Vahidi. 2023. Level-K Reasoning, Deep Reinforcement Learning, and Monte Carlo Decision Process for Fast and Safe Automated Lane Change and Speed Management. *IEEE Trans. on Intell. Veh.* 8, 6 (2023), 3556–3571.

[12] S. Karimi and A. Vahidi. 2021. Monte Carlo Tree Search and Cognitive Hierarchy Theory for Interactive-Behavior Prediction in Fast Trajectory Planning and Automated Lane Change. *ASME J. Auton. Veh. Syst.* 1, 1 (2021), 011008. https://doi.org/10.1115/1.4050042

[13] Levente Kocsis and Csaba Szepesvári. 2006. Bandit Based Monte-Carlo Planning. In *Machine Learning: ECML 2006*, Johannes Fürnkranz, Tobias Scheffer, and Myra Spiliopoulou (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 282–293.

[14] Vojtěch Kovařík, Nathaniel Sauerberg, Lewis Hammond, and Vincent Conitzer. 2025. Game Theory with Simulation in the Presence of Unpredictable Randomisation. In *Proceedings of the 24th International Conference on Autonomous Agents and Multiagent Systems* (Detroit, MI, USA) *(AAMAS '25)*. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 1191–1199.

[15] Duong Le and Erion Plaku. 2019. Multi-Robot Motion Planning With Dynamics via Coordinated Sampling-Based Expansion Guided by Multi-Agent Search. *IEEE Robot. Autom. Lett.* 4, 2 (2019), 1868–1875. https://doi.org/10.1109/LRA.2019.2898087

[16] G. H. Lee, D.-H. Kim, J. M. Pak, and C. K. Ahn. 2025. Vehicle Sideslip Angle Estimation Using Finite Memory Estimation and Dynamics/Kinematics Model Fusion Based on Neural Networks. *IEEE Trans. Intell. Transp. Syst.* 26, 2 (2025), 2157–2168. https://doi.org/10.1109/TITS.2024.3500794

[17] Jean Leneutre, Vadim Malvone, and James Ortiz. 2025. Timed Obstruction Logic: A Timed Approach to Dynamic Game Reasoning. In *Proceedings of the 24th International Conference on Autonomous Agents and Multiagent Systems* (Detroit, MI, USA) *(AAMAS '25)*. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 1272–1281.

[18] D. Lenz, T. Kessler, and A. Knoll. 2016. Tactical Cooperative Planning for Autonomous Highway Driving Using Monte-Carlo Tree Search. In *Proc. IEEE Intelligent Vehicles Symposium (IV)*. IEEE, Gothenburg, Sweden, 447–453.

[19] D. Li, J. Zhang, and G. Liu. 2024. Autonomous Driving Decision Algorithm for Complex Multi-Vehicle Interactions: An Efficient Approach Based on Global Sorting and Local Gaming. *IEEE Trans. Intell. Transp. Syst.* 25, 7 (2024), 6927–6937. https://doi.org/10.1109/TITS.2023.3346048

[20] Nan Li, Yu Yao, Ilya Kolmanovsky, Ella Atkins, and Anouck R. Girard. 2022. Game-Theoretic Modeling of Multi-Vehicle Interactions at Uncontrolled Intersections. *IEEE Trans. Intell. Transp. Syst.* 23, 2 (2022), 1428–1442. https://doi.org/10.1109/TITS.2020.3026160

[21] Zhihao Lin, Jianglin Lan, Christos Anagnostopoulos, Zhen Tian, and David Flynn. 2025. Safety-Critical Multi-Agent MCTS for Mixed Traffic Coordination at Unsignalized Intersections. *IEEE Transactions on Intelligent Transportation Systems* 1 (2025), 1–15.

[22] Z. Lin and Z. Tian. 2025. Scenario-based Decision-Making Using Game Theory for Interactive Autonomous Driving: A Survey. *arXiv preprint arXiv:2509.05777* abs/2509.05777, 2509.05777 (2025), 1–21.

[23] Wenliang Liu, Suhail Alsalehi, Noushin Mehdipour, Ezio Bartocci, and Calin Belta. 2025. Quantifying the Satisfaction of Spatio-Temporal Logic Specifications for Multiagent Control. *IEEE Trans. Autom. Control* 70, 8 (2025), 5098–5113. https://doi.org/10.1109/TAC.2025.3538747

[24] C. Ma and Others. 2021. Trajectory Planning for Connected and Automated Vehicles at Isolated Signalized Intersections Under Mixed Traffic Environment. *Transp. Res. Part C Emerg. Technol.* 130 (2021), 103309. https://doi.org/10.1016/j.trc.2021.103309

[25] Negar Mehr, Mingyu Wang, Maulik Bhatt, and Mac Schwager. 2023. Maximum-Entropy Multi-Agent Dynamic Games: Forward and Inverse Solutions. *IEEE Trans. Robot.* 39, 3 (2023), 1801–1815. https://doi.org/10.1109/TRO.2022.3232300

[26] Hongsheng Qi and Xianbiao Hu. 2019. Monte Carlo Tree Search-based intersection signal optimization model with channelized section spillover. *Transp. Res. Part C Emerg. Technol.* 106 (2019), 281–302.

[27] E. Sebastián, T. Duong, N. Atanasov, E. Montijano, and C. Sagüés. 2025. Physics-Informed Multiagent Reinforcement Learning for Distributed Multirobot Problems. *IEEE Trans. Robot.* 41 (2025), 4499–4517. https://doi.org/10.1109/TRO.2025.3582836

[28] E. Seraj, L. Chen, and M. C. Gombolay. 2022. A Hierarchical Coordination Framework for Joint Perception-Action Tasks in Composite Robot Teams. *IEEE Trans. Robot.* 38, 1 (2022), 139–158. https://doi.org/10.1109/TRO.2021.3096069

[29] Esmaeil Seraj, Rohan Paleja, Luis Pimentel, Kin Man Lee, Zheyuan Wang, Daniel Martin, Matthew Sklar, John Zhang, Zahi Kakish, and Matthew Gombolay. 2024. Heterogeneous Policy Networks for Composite Robot Team Communication and Coordination. *IEEE Trans. Robot.* 40 (2024), 3833–3849. https://doi.org/10.1109/TRO.2024.3431829

[30] Y. Shoham and K. Leyton-Brown. 2008. *Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations*. Cambridge University Press, Cambridge, UK.

[31] Zhen Tian, Dezong Zhao, Zhihao Lin, David Flynn, Wenjing Zhao, and Daxin Tian. 2024. Balanced Reward-Inspired Reinforcement Learning for Autonomous Vehicle Racing. In *Proceedings of the 6th Annual Learning for Dynamics and Control Conference*. PMLR, Oxford, UK, 628–640.

[32] Shuling Wang, Wei Huang, and Hong K. Lo. 2020. Combining shockwave analysis and Bayesian Network for traffic parameter estimation at signalized intersections considering queue spillback. *Transp. Res. Part C: Emerg. Technol.* 120 (2020), 102807.

[33] X. Wang, J. Gao, X. Zhou, and X. Gu. 2024. Path Planning for the Gantry Welding Robot System Based on Improved RRT*. *Robot. Comput.-Integr. Manuf.* 85 (2024), 102643. https://doi.org/10.1016/j.rcim.2023.102643

[34] H. Xu, Y. Zhang, C. G. Cassandras, L. Li, and S. Feng. 2020. A Bi-Level Cooperative Driving Strategy Allowing Lane Changes. *Transp. Res. Part C Emerg. Technol.* 120 (2020), 102773. https://doi.org/10.1016/j.trc.2020.102773

[35] J. Yan, X. Lin, Z. Ren, S. Zhao, J. Yu, C. Cao, P. Yin, J. Zhang, and S. Scherer. 2023. MUI-TARE: Cooperative Multi-Agent Exploration With Unknown Initial Position. *IEEE Robot. Autom. Lett.* 8, 7 (2023), 4299–4306. https://doi.org/10.1109/LRA.2023.3281262

[36] M. Yuan, J. Shan, and H. Schofield. 2024. Scalable Game-Theoretic Decision-Making for Self-Driving Cars at Unsignalized Intersections. *IEEE Trans. Ind. Electron.* 71, 6 (2024), 5920–5930. https://doi.org/10.1109/TIE.2023.3290255

[37] X. Zhang, L. Wu, H. Liu, Y. Wang, H. Li, and B. Xu. 2023. High-Speed Ramp Merging Behavior Decision for Autonomous Vehicles Based on Multiagent Reinforcement Learning. *IEEE Internet Things J.* 10, 24 (2023), 22664–22672.

[38] R. Zhao, K. Wang, Y. Li, Y. Fan, F. Gao, and Z. Gao. 2025. Safe Multi-Agent Deep Reinforcement Learning for the Management of Autonomous Connected Vehicles at Future Intersections. *IEEE Trans. Parallel Distrib. Syst.* 36, 8 (2025), 1744–1761. https://doi.org/10.1109/TPDS.2025.3580092

[39] J. Zhu, K. Gao, H. Li, Z. He, and C. O. Monreal. 2023. Bi-Level Ramp Merging Coordination for Dense Mixed Traffic Conditions. *Fundamental Research* 3, 2 (2023), 259–269.