LEARNING TO COORDINATE THROUGH STRATEGIC GAME SLICES

Anonymous authors
Paper under double-blind review

ABSTRACT

Coordination under private information requires agents to infer partners' preferences and select actions using those inferences. We study these capabilities in a finite negotiation game with private preferences and mixed incentives. A reference-policy oracle supplies posterior judgments and counterfactual action values, enabling matched diagnostic interventions. For Qwen3-4B-Instruct-2507, these probes reveal errors in both preference inference and action selection: supplying an oracle judgment does not reliably eliminate action errors. We use the same oracle to construct game slices and train a shared model on direct action selection, preference inference, and planning. With approximately 6M generated response tokens, joint slice training retains 47 of 72 meetings in a 24-game, four-agent sequential calendar suite adapted from CalBench, compared with 37 for both the unchanged model and complete-game self-play. These results provide initial evidence that targeted supervision from controlled strategic games can improve coordination in an environment with different interaction mechanics.

1 INTRODUCTION

Large language models (LLMs) increasingly operate in multi-agent systems that distribute work and coordinate decisions through interaction (Zhu et al., 2025; Chen et al., 2026). Yet interaction can fail at several distinct points. In tasks with distributed information, agents leave partners' private evidence unexplored (Li et al., 2026a); in distributed computation, they exchange messages without turning them into effective reasoning (Zhang et al., 2026b); in negotiation, sharing the relevant information does not by itself ensure a coordinated plan (Yao et al., 2026). A broader analysis likewise identifies inter-agent misalignment as a distinct failure class (Cemri et al., 2025). These findings raise a basic question: what prevents individually capable agents from coordinating effectively?

Interactive decision theory provides a starting point. In an interactive partially observable Markov decision process (I-POMDP), an agent maintains beliefs about both the physical environment and other agents, and uses those beliefs to choose actions (Gmytrasiewicz & Doshi, 2005). Opponent-modeling work similarly links inferences about others' goals or policies to the planning of a response (Carmel & Markovitch, 1996; Huang et al., 2024). Building on this distinction, we focus on two capabilities: interactive-state inference, which estimates partners' task-relevant hidden states from the available evidence, and interaction planning, which uses those estimates to evaluate and select actions. We refer to them as *inference* and *planning*. They form a feedback loop: an inference informs an action, which shapes partners' responses and the evidence available for subsequent decisions. Studies of LLM agents find that accurately predicting another agent's state or behavior need not lead to an effective response (Riemer et al., 2025; Sobotka et al., 2026). This gap motivates evaluating both the inference itself and the decisions made from it.

Diagnosing these capabilities in realistic interactions is difficult because it is hard to extract a clear answer from free-form interactions. Coordination with private calendars, strategic coalition formation, and commerce on behalf of different principals illustrate this complexity (Zou et al., 2026; O'Neill et al., 2026; Li et al., 2026b). Games offer a controlled setting: their information structure, utilities, legal actions, and strategic consequences can be specified and verified. Recent work has used multi-agent games and self-play to train strategic or transferable reasoning in LLMs (Yuan et al., 2026; Liu et al., 2026; Lyu et al., 2026; He et al., 2026b). Yet a game-level outcome alone does not reveal whether an agent misread a partner or failed to act effectively on an adequate belief. In so-

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

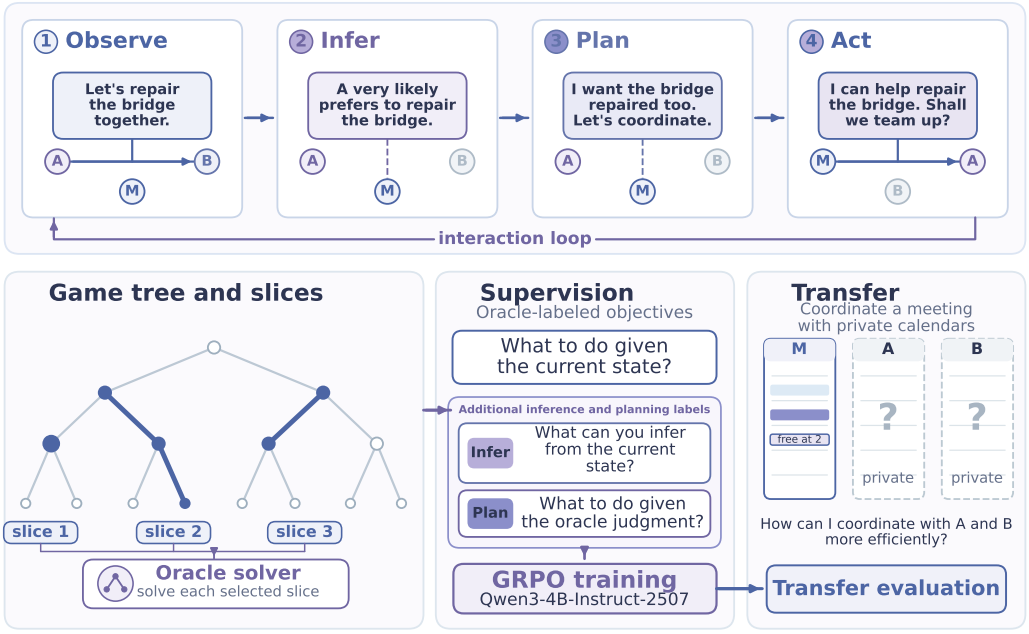


Figure 1: From interaction to targeted training and transfer. Agent M observes its partners, infers their preferences, plans a response, and acts to create new observations. We select game slices (colored nodes and paths) and solve them with an oracle to supervise action selection, inference, and planning. These objectives train a shared model with GRPO. We then evaluate transfer to meeting coordination, where agents use private calendars and exchanged messages to agree on meeting times.

cial deduction games, for example, a single trajectory combines inference, deception, and collective decision-making (Light et al., 2023). We seek a setting that retains strategic uncertainty while allowing beliefs and actions to be evaluated separately. To this end, we extend the commitment-based multi-party negotiation game of Benac et al. (2026) with private preferences and costly investigation actions. Hidden preferences create an inference problem; their alignment or conflict determines the value of possible plans.

We use an exact reference-policy solver to diagnose interactive-state inference and belief-conditioned planning separately, turn the same strategically consequential decision points into targeted training examples, and test whether this supervision transfers beyond the training game (Figure 1). At matched decisions, we evaluate both capabilities and intervene by replacing the model's inference with an oracle judgment. This diagnosis reveals failures in both capabilities: supplying the oracle judgment often changes the model's choices but does not reliably resolve planning errors. We then use the same oracle to supervise *game slices*, contiguous segments of one or more decisions drawn from deliberately structured games. These slices provide targeted training in direct action selection, inference, and planning conditioned on an oracle judgment. Under the same response-token budget, slice training yields stronger transfer than complete-game self-play to coordination with private calendars, where agents must reconcile private constraints and maintain commitments across successive meetings. Together, these findings suggest that carefully designed games and strategically selected slices can turn identifiable interaction challenges into effective supervision for more complex multi-agent environments.

2 Motivation and Methods

2.1 From Interactive Uncertainty to Two Capabilities

Consider three coding agents working on the same Python backend. In an Anthropic experiment motivated by behavior observed in deployment, each was instructed to migrate it to a different language. The agents mistook the resulting interference for deliberate obstruction and escalated to repeatedly killing competing processes and, in some runs, revoking one another's access (Anthropic, 2026).

They depended on a usable shared backend, yet pursued incompatible local objectives without knowing the others' instructions. Such coordination is naturally an interactive partially observable decision problem. For agent i, an I-POMDP is $\mathcal{I}_i = \langle IS_i, A, T_i, \Omega_i, O_i, R_i \rangle, IS_i = S \times \prod_{j \neq i} M_j$ where S is the physical state, M_j models agent j, A is the joint action space, and T_i, O_i, R_i specify transition, observation, and reward (Gmytrasiewicz & Doshi, 2005). Uncertainty therefore covers not only the world but also others' information, preferences, and policies. This view underlies opponent modelling and belief-space coordination (Carmel & Markovitch, 1996; Huang et al., 2024; Goel et al., 2026); the observation–belief–action links nevertheless remain fragile for LLM agents (Sobotka et al., 2026; Jha et al., 2026). Let $b_t^i \in \Delta(IS_i)$ be agent i's belief conditioned on public history h_t, its own preferences v_i, and its private observations r_t^i. An observation updates it as $b_{t+1}^i = \tau_i(b_t^i, a_t^i, o_{t+1}^i)$, while an action is selected by belief-conditioned planning: $a_t^{i,*} \in \arg\max_{a \in A_i} \mathbb{E}\Big[\sum_{k=t}^{T} \gamma^{k-t} R_i(x_k, a_k) \mid b_t^i, a_t^i = a, \pi_{-i}\Big]$. We call these computations *inference*, which maintains the part of b_t^i predictive of others, and *planning*, which evaluates actions under that belief. They form a loop: inference informs an action, the action determines the next observation, and that observation informs the next inference. For this distinction to be experimentally useful, the hidden state must be both inferable and consequential: interaction must provide evidence that supports a reasonable belief about another agent, and different beliefs must affect the relative value of available actions. We therefore need decision points at which evidence, private information, and action consequences are all specified well enough to evaluate the belief and the choice separately. In the game below, we operationalize inference with an inference target and planning with counterfactual action values.

2.2 Negotiation with Private Preferences and Mixed Incentives

To make inference and planning separately observable while retaining exact supervision, we extend the commitment-based negotiation game of Benac et al. (2026) with private preferences and costly information gathering. An instance of this game is

$$\mathcal{E} = \langle \mathcal{N}, \mathbf{K}, \mathcal{G}, \Gamma, \delta, \rho, C_0, \mathcal{L}, \{\Theta_i\}_{i \in \mathcal{N}}, \mu_0, \mathbf{q}_0 \rangle. \tag{1}$$

Here $\mathcal{N}$ is the player set and $\mathbf{K} = (K_i)_{i \in \mathcal{N}}$ gives each player's number of binary commitment coordinates. Each public goal $g \in \mathcal{G}$ has a nonempty requirement set $\Gamma_g \subseteq \{(i,k) : i \in \mathcal{N},\ 0 \leq k < K_i\}$ and a scoring mode $\delta_g \in \{\text{binary}, \text{linear}\}$. The public schedule $\rho = (\rho_t)_{t=0}^{T-1}$ specifies the proposer at each of T proposal opportunities, and $\mathcal{L}$ gives the legal actions at each information set. The commitment state is $C = (c_{ik})$, with $c_{ik} \in \{0, 1\}$; initially $C_0 = 0$, and a commitment cannot be withdrawn. Each player privately observes its preference vector $v_i \in \Theta_i \subseteq \{-1, 0, +1\}^{|\mathcal{G}|}$. The joint type $v = (v_i)_{i \in \mathcal{N}}$ is drawn from the common prior $\mu_0 \in \Delta(\prod_{i \in \mathcal{N}} \Theta_i)$; entries encode AVOID, NEUTRAL, and WANT, respectively. $\mathbf{q}_0 = (q_{i,0})_{i \in \mathcal{N}}$ gives the initial investigation budgets. For any commitment state C, goal satisfaction and terminal utility are

$$S_g(C) = \begin{cases} \mathbf{1}[\, c_{ik} = 1,\ \forall (i,k) \in \Gamma_g \,], & \delta_g = \text{binary}, \\ \dfrac{1}{|\Gamma_g|} \sum_{(i,k) \in \Gamma_g} c_{ik}, & \delta_g = \text{linear}, \end{cases} \qquad U_i(C; v_i) = \sum_{g \in \mathcal{G}} v_{ig} S_g(C). \tag{2}$$

At initialization, the goals, prior, legal-action rules, and proposer schedule are public, while some of the preferences are hidden; every player starts with one investigation opportunity ($q_{i,0} = 1$). Play follows ρ. At each proposal opportunity, the scheduled player may PASS, make a bilateral OFFER to a chosen partner, or, if $q_{i,t} > 0$, use INVESTIGATE(j, g) to learn v_{jg}. Passing leaves commitments unchanged. Investigation also leaves them unchanged, uses that proposal opportunity and one unit of budget, and reveals the answer only to the investigator; the query target is public. An offer specifies new commitments for the proposer, the partner, or both. The partner immediately chooses ACCEPT or REJECT, acceptance makes all proposed additions binding and irreversible, whereas rejection changes no commitments. The response does not use another proposal opportunity. Play then proceeds to the next scheduled proposer and ends after the final opportunity, when each player receives $U_i(C_T; v_i)$. To improve its utility, a player infers partners' preferences from offers, responses, and any private investigation answers, then plans whom to approach, what to offer, and whether to accept, accounting for goals on which preferences align or conflict. Figure 2 illustrates these transitions in a three-player game. Appendix A provides the full formal specification.

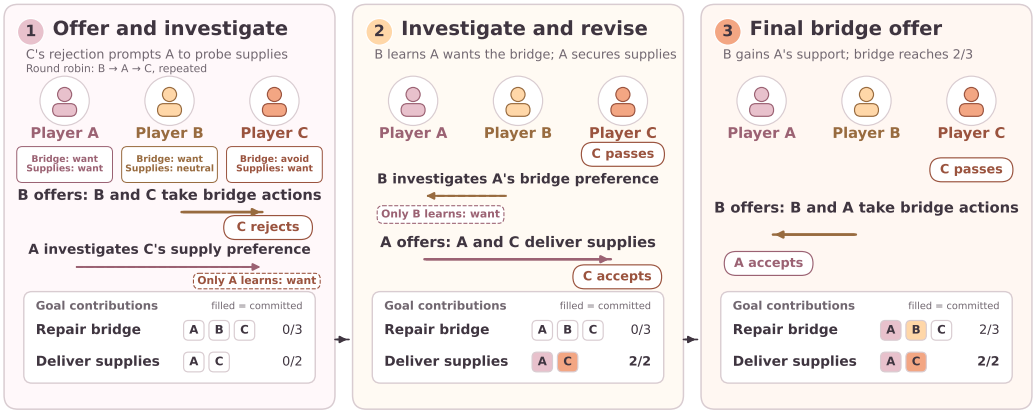


Figure 2: A three-player negotiation with private preferences and two linearly scored goals. Filled letters indicate commitments; fractions show goal progress. Offers take effect only upon acceptance, while investigations privately reveal preferences and consume a proposer turn. Proposer turns follow the fixed round-robin order B–A–C.

2.3 Game Slices and Oracle Solver

A *game slice* is a consecutive segment of play that begins at a chosen decision and may include later decisions. For example, it can contain an offer, the partner's response, and the next proposal. The slice records which actions occurred and what each player observed following the game rule. Whereas a full game starts with zero commitments, a slice may start from a legal history prefix or an explicitly specified exogenous state, with its current commitments and remaining investigation budgets recorded. At decision s, the acting player i knows the public history h_s, its own preferences v_i, and its private answers r_s^i. We write this view as $I_s^i = (h_s, v_i, r_s^i)$, an *information set* that identifies the game states the player cannot distinguish (Kuhn, 1953). The observed sequence locates the decisions to be labeled. For each such decision, the oracle also evaluates actions that were not taken and their possible continuations; it does not give the acting player access to the partners' hidden preferences. A formal slice definition appears in Appendix B. Following the Bayesian representation of private types (Harsanyi, 1967), the oracle solves the finite tree over possible preferences and actions under the common prior μ_0. The oracle labels are exact relative to the public prior, the specified partner policies, and the fixed continuation policy. A *policy* $\pi_i(a \mid I)$ assigns a probability to each legal action $a \in \mathcal{L}(I)$ given what player i can know. Let Π_i contain all such policies, including plans for later decisions after each possible observation; π_{-i} denotes the other players' policies. Holding π_{-i} fixed, a *best response* maximizes player i's expected terminal utility over these complete plans. Starting from uniform legal actions, we update every player against the same previous policy profile:

$$\mathrm{BR}_i(\pi_{-i}) \in \arg\max_{\widetilde{\pi}_i \in \Pi_i} \mathbb{E}_{\mu_0,(\widetilde{\pi}_i,\pi_{-i})}[U_i(C_T; v_i)],$$
$$\pi_i^{(0)}(a \mid I) = |\mathcal{L}(I)|^{-1}, \qquad \pi_i^{(k+1)} = \mathrm{BR}_i(\pi_{-i}^{(k)}). \tag{3}$$

We retain a fixed profile $\pi^\star$ only if it passes a no-profitable-unilateral-deviation check, following the stability criterion of Nash (1950): in every initial private-information class, no player can improve its expected utility by changing its complete plan while the others keep theirs. At ACCEPT/REJECT responses, expected utility to the other players breaks ties in the responder's own utility; remaining ties are mixed uniformly. The reference policy also turns interaction into an inference target. Let $b_s^{i,\star}(v_{-i}) = \Pr_{\mu_0,\pi^\star}(v_{-i} \mid I_s^i)$ be player i's *posterior*, its probability distribution over the others' hidden preferences. If partner j autonomously chooses a_s^j, Bayes' rule gives

$$b_{s+1}^{i,\star}(v_{-i}) \;\propto\; b_s^{i,\star}(v_{-i})\pi_j^\star(a_s^j \mid I_s^j(v_i, v_{-i})), \qquad j \neq i. \tag{4}$$

An action is evidence to the extent that the reference policy makes it more likely under one candidate preference assignment than another. For planning, we instead ask what would happen if i chose each legal action a now:

$$Q_i^\star(I_s^i, a) = \mathbb{E}_{\mu_0,\pi^\star}[U_i(C_T; v_i) \mid I_s^i, \mathrm{do}(a_s^i = a)]. \tag{5}$$

The forced action is followed by the reference policy for the rest of the game. Across a longer slice, each player's posterior is updated from the observations available to that player and informs its later decisions. Appendix B details the computation and validity checks; Section 4.1 describes slice construction and sampling.

3 DIAGNOSIS

An unsuccessful interaction does not reveal whether an agent misread its partner or chose poorly despite an adequate understanding. We diagnose these two sources of error in the Qwen3-4B-Instruct-2507 model (Yang et al., 2025). At matched decision points, we ask whether the model can infer a partner's preference from available evidence, choose a good action given an oracle judgment, and improve its choice when an incorrect inference is replaced by an oracle judgment.

Intervening on the inference supplied to planning. Each case elicits four responses. The model first reports a preference judgment $\widehat{J}_c$ from the history and available private information. Two separate planning calls then receive the same current state, with either that judgment ($\mathrm{Plan}_{\mathrm{model}}$) or the oracle judgment $J_c^\star$ ($\mathrm{Plan}_{\mathrm{oracle}}$). We also evaluate *direct action selection* from the original state, without a supplied judgment. The paired planning calls share their tools, decoding settings, and seed. We remove the behavioral history from both calls so that planning uses the supplied inference rather than reconstructing it from the original evidence. To ensure that this removal does not make the task underspecified, we retain only cases in which the oracle judgment determines the optimal action set and removing history preserves the continuation payoff table.

Evidence and oracle labels. The suite contains 22 two-player cases spanning seven game structures. Six test use of a preference revealed by investigation; six require ruling out preferences after one observed partner choice; six retain uncertainty after one choice; and four require combining successive observations. Each case has one unknown partner–goal preference, taking values WANT, NEUTRAL, or AVOID. The oracle conditions on all available evidence under the public prior and reference policy of Section 2.3. Its label specifies the *support*, the preferences with positive posterior probability, and the *most likely preference*. We report the latter only when its posterior probability exceeds the next highest by more than 0.1; otherwise, we report UNDETERMINED. These labels describe what the evidence supports and we verify that all posteriors compatible with each oracle label yield the same optimal action set. Appendix C details these constructions.

We report joint inference correctness: a judgment is correct only when both its support and most likely preference match the oracle label. Action correctness uses the oracle's acceptable-action set, with the utility tolerance specified in Appendix C. To measure the cost of an error and the effect of replacing a judgment, let $V_c(a) = Q_i^\star(I_c, a)$ be the expected terminal utility of action a under the oracle posterior and reference continuation. We report

$$\mathrm{Reg}_c(a) = \max_{a' \in \mathcal{L}(I_c)} V_c(a') - V_c(a), \qquad \Delta_c = V_c(\widehat{a}_c(J_c^\star)) - V_c\Big(\widehat{a}_c(\widehat{J}_c)\Big), \tag{6}$$

where $\widehat{a}_c(J)$ is the model's planning response to judgment J. Both actions are scored under the same posterior and continuation, $\Delta_c > 0$ means that replacing the judgment improves the selected action's value.

Inference and planning failures. The model recovers all six directly revealed preferences but fails all ten single-choice elimination and multi-step judgments (Table 1). Even with the oracle judgment, planning succeeds in only $11/22$ cases, with mean regret 0.567. Among the ten valid pairs with an incorrect initial judgment, oracle replacement changes the selected action in nine. Across all ten pairs, three replacements improve utility, six leave it unchanged, and one reduces it. Across all 21 valid pairs, mean regret falls from 0.684 to 0.541 despite unchanged aggregate action accuracy. Thus both behavioral inference and action selection exhibit failures; replacing the supplied inference with the oracle judgment can improve choices but does not reliably resolve planning errors. Appendix C provides the breakdown by sensitivity to partner preferences.

An illustrative intervention. In Table 2, Blair asks Alex for a commitment that makes completing Harbor also complete Orchard. Since Blair could instead secure Harbor alone, this choice rules out

Table 1: Diagnosis of the Qwen3-4B-Instruct-2507 model. Top: correct responses out of all cases. Joint inference requires both the support and most likely preference to be correct; Direct selects an action from the original state. Bottom: paired intervention results, grouped by whether the model's initial judgment was correct. One invalid $\mathrm{Plan}_{\mathrm{model}}$ response counts as incorrect above and is excluded below. Equal action accuracy can conceal changes in action value.

	Inference		Action	
Evidence	Joint	Direct	$\mathrm{Plan}_{\mathrm{model}}$	$\mathrm{Plan}_{\mathrm{oracle}}$
Direct feedback	6/6	2/6	3/6	3/6
Single-choice elimination	0/6	3/6	2/6	3/6
Ambiguous single choice	5/6	2/6	2/6	2/6
Multi-step update	0/4	4/4	4/4	3/4
All cases	11/22	11/22	11/22	11/22

Initial judgment	Pairs	Action changes	$\Delta > 0$	$\Delta = 0$	$\Delta < 0$	Mean Δ
Incorrect	10	9	3	6	1	0.300
Correct	11	0	0	11	0	0.000
All valid pairs	21	9	3	17	1	0.143

Table 2: A two-goal inference error and its correction. Both goals are binary and require both listed commitments. Alex wants both goals; only Blair's Orchard preference is hidden. The two planning calls use the same model and state, with different supplied judgments. Utilities are evaluated under the oracle posterior and reference policy.

(a) Goals and preferences

	Required commitments		Preferences	
Goal	Alex	Blair	Alex	Blair
Orchard	Cedar	Cedar	Want	Hidden
Harbor	Maple	Cedar	Want	Want

History	Blair offers to add nothing and asks Alex to add Cedar; Alex accepts.
Current decision	Alex has Cedar; Blair has none. Alex has the final proposal turn.
Key implication	After Alex adds Cedar, any completion of Harbor also completes Orchard. A Blair that avoids Orchard could instead have secured Harbor alone.
Model judgment	Possible preferences: Avoid only; most likely: Avoid.
Oracle judgment	Possible preferences: Want or Neutral; most likely: Want.

(b) Final offers and outcomes

Action source	Alex adds	Blair adds	Goals completed	U_{Alex}
$\mathrm{Plan}_{\mathrm{model}}$	None	Maple	None	0
$\mathrm{Plan}_{\mathrm{oracle}}$	Maple	Cedar	Orchard and Harbor	2

AVOID for Orchard under the reference policy. The model infers the opposite from Blair's lack of its own commitment. Replacing that judgment changes its offer from completing neither goal to completing both, raising utility from 0 to 2. The improved offer is also optimal under the model's original judgment, revealing a planning error alongside the inference error. Appendix C.1 gives the derivation and outputs; Appendix C.2 provides a case where correction leaves a planning error.

Together, these probes reveal failures in extracting preference information and in using sufficient information to choose an action. The intervention connects the two at the explicit judgment–action interface. These findings motivate training both capabilities alongside direct action selection; the effects of training and transfer are evaluated separately below.

4 EXPERIMENTS

4.1 DATA AND TRAINING

Data generation. We construct 100 game slices from the games in Section 2.2, selecting for identifiable inference and planning demands. Inference coverage includes evidence that changes or preserves a preference judgment and investigation outcomes that inform subsequent choices. Planning coverage includes commitments that shape later opportunities and agreements that depend on multiple partners' responses and interdependent commitments. We include information contrasts re-

Table 3: Same-model negotiation: mean team utility on 16 games per team size (↑).

Model	Two players	Three players	Four players
Baseline	0.969	0.583	1.177
Self-play	1.063	0.542	1.250
GAMESLICE-A	1.238	0.911	2.135
GAMESLICE-AP	1.375	1.438	1.583
GAMESLICE-JOINT	1.481	1.615	1.594

quiring different actions, alongside controls in which actions should remain unchanged. For scored action tasks, we require $\varnothing \neq \mathcal{A}_\epsilon(I) \subsetneq \mathcal{L}(I)$, where $\mathcal{A}_\epsilon(I)$ is the oracle's acceptable-action set under the scoring tolerance: at least one legal action must be acceptable and one unacceptable. Slices are executed with the oracle to advance game states and generate decision-level supervision. Histories are checked by legal replay. The oracle supplies three supervision types: direct action selection, inference of possible and most likely preferences, and planning given an oracle judgment. Failed reference computations are excluded, and planning labels ambiguous under the supplied judgment are masked. Appendix D.1 details these structural selection and coverage criteria.

Training. All trained variants start from Qwen3-4B-Instruct-2507 (Yang et al., 2025). Self-play (SP) learns from each player's terminal utility in complete games, starting from 100 initial game configurations. Slice training instead uses oracle-scored responses: GAMESLICE-A trains direct action selection, GAMESLICE-AP trains direct action selection and conditional planning, and GAMESLICE-JOINT combines all three tasks. These objectives update a single model without imposing an inference-to-planning pipeline at test time. Slice training uses grouped policy optimization with eight responses per item and a KL penalty toward the frozen base model. We use BF16 and an initial learning rate of 10^{-6}. All training variants use a budget of approximately 6M generated response tokens, and we evaluate the final checkpoint of each training run. The training settings are listed in Tables 6 and 7 in Appendix D.

4.2 SAME-MODEL NEGOTIATION

We evaluate complete games with two, three, and four players, using 16 fixed instances per team size and one greedy rollout per instance. All players use the same checkpoint, each observing only its own private information and the public history. Instances are shared across models within each size. We report mean team utility, the sum of players' signed terminal utilities averaged over games. Table 3 reports the results; Appendix E provides more detailed statistics.

Results. All slice-trained variants exceed the baseline's mean utility in each suite, but adding supervision objectives does not yield monotonic gains. Self-play remains close to the baseline. In four-player games, GAMESLICE-A secures more accepted offers than Self-play despite making fewer proposals (38/59 versus 25/78). One rollout shows how successive bilateral agreements accumulate the commitments needed for two goals, yielding team utility 4 versus Baseline's 0.

4.3 TRANSFER TO SEQUENTIAL CALENDAR COORDINATION

CalBench (Zou et al., 2026) evaluates decentralized meeting coordination: agents with private calendars and rescheduling costs communicate to agree on meeting times. A feasible agreement requires agents to learn their partners' constraints and reconcile them with their own schedules, potentially moving private errands or existing meetings. We evaluate transfer on a 24-game sequential adaptation, with four agents coordinating three meetings per game. Meetings arrive sequentially, so accommodating a new request must also account for earlier commitments. Each team uses the same model for all agents, and all models face identical scenarios and evaluation settings. Detailed environment and inference settings are provided in Appendix F.1.

Evaluation. We report retained meetings (out of 72), fully successful games (out of 24), per-agent excess rescheduling cost, communication overhead, and burden imbalance. Success is determined from the final calendars. Excess cost is measured relative to the minimum feasible team cost and averaged only over successful games. A retained meeting must occupy a common slot for every par-

Table 4: CalBench transfer on 24 frozen sequential games. Meetings count final retained meetings; excess cost is per agent and conditional on full-game success. Communication and imbalance average native per-agent values over all games.

Model	Meetings ↑	Full games ↑	Cost ↓	Comm. ↓	Imbalance ↓
Baseline	37	5	0.100	5.663	0.443
Self-play	37	6	0.625	5.181	0.755
GAMESLICE-A	38	8	0.688	5.003	0.490
GAMESLICE-AP	44	10	0.075	4.760	0.432
GAMESLICE-JOINT	47	9	0.194	4.714	0.292
Qwen3-4B (thinking)	42	7	0.571	4.828	0.557
MARSHAL Generalist	37	6	1.333	4.429	0.875
Social-R1-4B	39	5	1.100	4.554	0.901

ticipant; a successful game must retain all three meetings while preserving blocked calendar items. Communication counts sent direct messages per agent, divided by its round-level scheduling successes (with denominator at least one). Imbalance measures deviations in per-agent excess burdens relative to a reference for the round-successful meeting subset. Both include failed games. We additionally report Qwen3-4B (thinking), MARSHAL Generalist (Yuan et al., 2026), and Social-R1-4B (Wu et al., 2026) as a separate comparison group. Exact metric definitions and model details appear in Appendix F.1.

Results. Table 4 shows that GAMESLICE-JOINT retains 47 meetings, compared with 37 for both the baseline and self-play, and completes nine games compared with five and six, respectively. It also reduces communication and burden imbalance relative to both, suggesting that its higher completion rate is accompanied by more efficient coordination and a more even distribution of costs. GAMESLICE-AP achieves the most complete games and the lowest conditional excess cost, although cost averages cover different successful subsets for each model. Both variants retain more meetings than action-only training, supporting the value of adding conditional planning supervision for transfer to this coordination task. A matched rollout makes the behavioral difference concrete. After learning that a proposed slot is blocked, GAMESLICE-JOINT checks a partner's alternative with the remaining participant. Later, the agents identify a conflict with an existing meeting and agree on another slot, retaining all three meetings. Under the same initial calendars, the baseline and self-play teams fail to establish a common slot for the second meeting and retain only the first. This trajectory illustrates two demands of coordination: using a partner's response to revise the feasible options, and checking a proposed revision against other participants' constraints and existing commitments. Appendix F.2 presents the detailed comparison with verbatim model outputs.

5 DISCUSSION

Beyond preference inference. Our games locate uncertainty primarily in partners' preferences. Real coordination also requires reasoning about what others have observed and what they believe others know, as explored in recent multi-agent theory-of-mind work (YS et al., 2026; Mu et al., 2026; Cross et al., 2025). These hidden states belong within inference, but introduce a further challenge: the same refusal may reflect conflicting preferences or missing information, calling for different responses. Future game slices could vary knowledge and preferences independently, testing whether agents distinguish these causes and choose when to clarify or disclose information. The key question is which distinctions in a partner's mental state are necessary for a good decision.

Coverage and interaction. Game slices let us select strategically consequential decisions and verify their supervision. Self-play provides coherent interaction trajectories, but visits states according to the policies currently playing, resulting in rare important decisions. This difference offers a possible explanation for the efficiency of targeted training, consistent with work on subgame curricula (Chen et al., 2024). A possible next step is to combine the two: use self-play to identify recurring failures, construct and solve slices around those decisions, and return to complete games to evaluate the learned behavior. This would test whether targeted practice can repair local weaknesses while preserving effective coordination over an entire interaction.

Inference under unfamiliar partner policies. Our oracle assumes a fixed partner policy, while self-play primarily exposes a model to copies of its evolving policy. Yet the meaning of an observed action depends on how the partner chooses: a rejection that rules out a preference under one policy may be compatible with it under another. Cross-partner coordination research motivates training with diverse partner populations (Hu et al., 2020; Strouse et al., 2021). For game slices, this means varying planning horizons, tie-breaking rules, or action noise and recomputing labels under each behavioral model. Evaluation on held-out policy families would test whether agents can interpret unfamiliar behavior and adapt their plans beyond the assumptions of a single oracle.

6 Related Work

Classical multi-agent decision making couples inference about other agents with downstream action selection, from opponent models and interactive beliefs (Carmel & Markovitch, 1996; Gmytrasiewicz & Doshi, 2005; Albrecht & Ramamoorthy, 2013) to belief-space coordination, inference from actions, and opponent-aware planning (Macindoe et al., 2012; Wu et al., 2021; Foerster et al., 2019; Huang et al., 2024). For LLM agents, explicit mental-state or behavioral representations support collaboration, adaptation, trust tracking, and planning (Li et al., 2023; Cross et al., 2025; Jha et al., 2026; Mu et al., 2026; Zhou et al., 2026; Feng et al., 2026; Goel et al., 2026; Hwang et al., 2026). However, accurate mental-state prediction need not yield useful behavior (Riemer et al., 2025; Sobotka et al., 2026).

Recent MAS benchmarks expose strategic failures under private information, mixed incentives, and imperfect communication. Commitment-based negotiation, self-negotiated contracts, and multi-party conquest model consequential agreements and coalition dynamics (Benac et al., 2026; Wyse et al., 2026; O'Neill et al., 2026); CalBench, SOTOPIA-TOM, and SNEAK make information disclosure and privacy strategically consequential (Zou et al., 2026; YS et al., 2026; Avsian & Heck, 2026). Broader evaluations cover collaboration, controlled interaction, long-horizon commerce, and communication barriers (Zhu et al., 2025; Chen et al., 2026; Li et al., 2026b; Xuan et al., 2026). Diagnostic work further finds failures in distributed information pooling, distributed computation, dynamic grounding, and belief-sensitive negotiation (Li et al., 2026a; Zhang et al., 2026b; Yao et al., 2026; Zhang et al., 2026a; Agashe et al., 2025); these complement broad failure taxonomies and accounts of long-lived peer interaction (Cemri et al., 2025; Anthropic, 2026).

Game-based training has been used to induce transferable strategic and social reasoning through self-play, multi-game reinforcement learning, and social reasoning supervision (Yuan et al., 2026; Liu et al., 2026; Lyu et al., 2026; He et al., 2026b; Hua et al., 2026; He et al., 2026a; Wu et al., 2026). Cross-environment cooperation and Other-Play study generalization across tasks and partners (Jha et al., 2025; Hu et al., 2020). Curriculum work selects terminal-proximal or high-learning-value subgames (West et al., 2020; Chen et al., 2024), while LLM curricula can also induce undesirable strategic abstractions (Madmoun & Lahlou, 2026). Similarity-based inference has also been proposed as a route to cooperation among foundation-model agents (Meulemans et al., 2026). Our work connects capability-specific diagnosis, targeted supervision, and transfer evaluation through the same oracle-labeled game slices.

7 Conclusion

We study inference and planning as distinct but coupled capabilities in multi-agent interaction. A controlled negotiation game makes both measurable, revealing that supplying the oracle judgment alone does not reliably resolve planning errors. Oracle-supervised game slices turn these capabilities into targeted training objectives and improve transfer to an external environment over complete-game self-play under the same response-token budget. These results suggest that deliberately structured games can serve as both diagnostic tools and sources of supervision for more complex multi-agent interaction.

AI USE STATEMENT

We used generative AI tools to assist with literature discovery, reference formatting, manuscript drafting and revision, and the creation and editing of scientific figures. We also used these tools to discuss and refine aspects of the methodological presentation and experimental design, and to assist in interpreting diagnostic results and comparing interaction trajectories. The authors reviewed the resulting text and figures, checked cited sources, and verified reported findings against experimental records. The authors take responsibility for the final content of this work, including its methods, claims, results, and artifacts.

REFERENCES

Saaket Agashe, Yue Fan, Anthony Reyna, and Xin Eric Wang. LLM-coordination: Evaluating and analyzing multi-agent coordination abilities in large language models. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 8053–8072. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.findings-naacl.448.

Stefano V. Albrecht and Subramanian Ramamoorthy. A game-theoretic model and best-response learning method for ad hoc coordination in multiagent systems. In *Proceedings of the 2013 International Conference on Autonomous Agents and Multi-Agent Systems*, pp. 1155–1156, 2013.

Anthropic. Patterns and problems in emerging multiagent systems. Anthropic Research blog, August 2026. URL https://www.anthropic.com/research/multiagent-systems. Published August 13, 2026.

Adar Avsian and Larry Heck. SNEAK: Evaluating strategic communication and information leakage in large language models. In *Conference on Language Modeling (COLM)*, 2026.

Leo Benac, Jonas Raedler, Zilin Ma, and Finale Doshi-Velez. A benchmark for multi-party negotiation games from real negotiation data, 2026.

David Carmel and Shaul Markovitch. Opponent modeling in multi-agent systems. In *Adaptation and Learning in Multi-Agent Systems*, volume 1042 of *Lecture Notes in Computer Science*, pp. 40–52. Springer, 1996. doi: 10.1007/3-540-60923-7_18.

Mert Cemri, Melissa Z. Pan, Shuyi Yang, Lakshya A. Agrawal, Bhavya Chopra, Rishabh Tiwari, Kurt Keutzer, Aditya Parameswaran, Dan Klein, Kannan Ramchandran, Matei Zaharia, Joseph E. Gonzalez, and Ion Stoica. Why do multi-agent LLM systems fail? In *Advances in Neural Information Processing Systems*, 2025.

Jiaju Chen, Bo Sun, Yuxuan Lu, Yun Wang, Dakuo Wang, and Bingsheng Yao. CollabSim: A CSCW-grounded methodology for investigating collaborative competence of LLM agents through controlled multi-agent experiments, 2026.

Jiayu Chen, Zelai Xu, Yunfei Li, Chao Yu, Jiaming Song, Huazhong Yang, Fei Fang, Yu Wang, and Yi Wu. Accelerate multi-agent reinforcement learning in zero-sum games with subgame curriculum learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(10):11320–11328, 2024. doi: 10.1609/aaai.v38i10.29011.

Logan Cross, Violet Xiang, Agam Bhatia, Daniel L. K. Yamins, and Nick Haber. Hypothetical minds: Scaffolding theory of mind for multi-agent tasks with large language models. In *International Conference on Learning Representations*, 2025.

Peilin Feng, Suorong Yang, and Soujanya Poria. Σ-Mem: An online reliability memory for LLM-based multi-agent systems, 2026.

Jakob Foerster, Francis Song, Edward Hughes, Neil Burch, Iain Dunning, Shimon Whiteson, Matthew Botvinick, and Michael Bowling. Bayesian action decoder for deep multi-agent reinforcement learning. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 1942–1951. PMLR, 2019.

Piotr J. Gmytrasiewicz and Prashant Doshi. A framework for sequential planning in multi-agent settings. *Journal of Artificial Intelligence Research*, 24:49–79, 2005. doi: 10.1613/JAIR.1579.

Harsh Goel, Aditya Sai Ellendula, Vaishnav Tadiparthi, Ehsan Moradi Pari, Hossein Nourkhiz Mahjoub, and Sandeep P. Chinchali. Bayesian partner modelling enables adaptive replanning for LLM coordination, 2026.

John C. Harsanyi. Games with incomplete information played by "bayesian" players, i–iii: Part i. the basic model. *Management Science*, 14(3):159–182, 1967. doi: 10.1287/mnsc.14.3.159.

Keyu He, Xuhui Zhou, and Maarten Sap. Social gym and SPaRTan: Benchmarking and improving LLM social reasoning via multi-agent game tournaments, 2026a.

Yidong He, Yutao Lai, Pengxu Yang, Jiarui Gan, Jiexin Wang, Yi Cai, and Mengchen Zhao. Stratreasoner: Reinforcing strategic reasoning of LLMs in multi-agent games. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026b.

Hengyuan Hu, Adam Lerer, Alex Peysakhovich, and Jakob Foerster. Other-Play for zero-shot coordination. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 4399–4410. PMLR, 2020.

Wenyue Hua, Zachary Huang, Tyler Payne, Safoora Yousefi, Saleema Amershi, and Asli Celikyilmaz. From passive delegates to strategic negotiators: Reinforcing social reasoning in small language models with SocialRL, 2026.

Yizhe Huang, Anji Liu, Fanqi Kong, Yaodong Yang, Song-Chun Zhu, and Xue Feng. Efficient adaptation in mixed-motive environments via hierarchical opponent modeling and planning. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 20004–20022. PMLR, 2024.

EunJeong Hwang, Yuwei Yin, Giuseppe Carenini, Peter West, and Vered Shwartz. Infusing theory of mind into socially intelligent LLM agents. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 11327–11360. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.findings-acl.551.

Kunal Jha, Wilka Carvalho, Yancheng Liang, Simon Shaolei Du, Max Kleiman-Weiner, and Natasha Jaques. Cross-environment cooperation enables zero-shot multi-agent coordination. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 27198–27220. PMLR, 2025.

Kunal Jha, Aydan Huang, Eric Ye, Natasha Jaques, and Max Kleiman-Weiner. Modeling others' minds as code. In *International Conference on Learning Representations*, 2026.

Harold W. Kuhn. Extensive games and the problem of information. In Harold W. Kuhn and Albert W. Tucker (eds.), *Contributions to the Theory of Games II*, volume 28 of *Annals of Mathematics Studies*, pp. 193–216. Princeton University Press, 1953.

Huao Li, Yu Quan Chong, Simon Stepputtis, Joseph Campbell, Dana Hughes, Michael Lewis, and Katia Sycara. Theory of mind for multi-agent collaboration via large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 180–192. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.emnlp-main.13.

Yuxuan Li, Aoi Naito, and Hirokazu Shirado. Systematic failures in collective reasoning under distributed information in multi-agent LLMs. In *Proceedings of the 43rd International Conference on Machine Learning*, volume 306 of *Proceedings of Machine Learning Research*, 2026a.

Zeyuan Li, Lukas Petersson, Alessandro Acquisti, and Michiel A. Bakker. Emergent misaligned communication in long-horizon multi-agent LLM commerce, 2026b.

Jonathan Light, Min Cai, Sheng Shen, and Ziniu Hu. AvalonBench: Evaluating LLMs playing the game of avalon, 2023.

Bo Liu, Leon Guertler, Simon Yu, Zichen Liu, Penghui Qi, Daniel Balcells, Mickel Liu, Cheston Tan, Weiyan Shi, Min Lin, Wee Sun Lee, and Natasha Jaques. SPIRAL: Self-play on zero-sum games incentivizes reasoning via multi-agent multi-turn reinforcement learning. In *International Conference on Learning Representations*, 2026.

Nuoyan Lyu, Bingbing Xu, Xueyun Tian, Weihao Meng, Yige Yuan, Yang Zhang, Zhiyong Huang, Tat-Seng Chua, and Huawei Shen. GIFT: Games as informal training for generalizable LLMs, 2026.

Owen Macindoe, Leslie Pack Kaelbling, and Tomás Lozano-Pérez. POMCoP: Belief space planning for sidekicks in cooperative games. *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, 8(1):38–43, 2012. doi: 10.1609/aiide.v8i1.12510.

Hachem Madmoun and Salem Lahlou. Communication enables cooperation in LLM agents: A comparison with curriculum-based approaches. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 307–321. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.eacl-short.23.

Alexander Meulemans, Maciej Wołczyk, Marissa A. Weis, Rajai Nasser, Roberta Rocca, Seijin Kobayashi, Guillaume Lajoie, Angelika Steger, Blake Richards, Marcus Hutter, James Manyika, Rif A. Saurous, João Sacramento, and Blaise Agüera y Arcas. A game theory for foundation models shows new paths to rational cooperation through similarity inference, 2026.

Chunjiang Mu, Ya Zeng, Qiaosheng Zhang, Kun Shao, Chen Chu, Hao Guo, Danyang Jia, Zhen Wang, and Shuyue Hu. Adaptive theory of mind for LLM-based multi-agent coordination. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 29608–29616, 2026. doi: 10.1609/aaai.v40i35.40204.

John F. Nash, Jr. Equilibrium points in N-person games. *Proceedings of the National Academy of Sciences*, 36(1):48–49, 1950. doi: 10.1073/pnas.36.1.48.

Abigail O'Neill, Alan Zhu, Mihran Miroyan, Narges Norouzi, and Joseph E. Gonzalez. Cooperate to compete: Strategic coordination in multi-agent conquest. In *Proceedings of the 2026 Conference on Empirical Methods in Natural Language Processing*, 2026.

Matthew Riemer, Zahra Ashktorab, Djallel Bouneffouf, Payel Das, Miao Liu, Justin D. Weisz, and Murray Campbell. Position: Theory of mind benchmarks are broken for large language models. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 82091–82130. PMLR, 2025.

Jan Sobotka, Mustafa O. Karabag, and Ufuk Topcu. Why do LLMs struggle in strategic play? broken links between observations, beliefs, and actions, 2026.

DJ Strouse, Kevin McKee, Matt Botvinick, Edward Hughes, and Richard Everett. Collaborating with humans without human data. In *Advances in Neural Information Processing Systems*, volume 34, 2021.

Joseph West, Frederic Maire, Cameron Browne, and Simon Denman. Improved reinforcement learning with curriculum. *Expert Systems with Applications*, 158:113515, 2020. doi: 10.1016/j.eswa.2020.113515.

Jincenzi Wu, Yuxuan Lei, Jianxun Lian, Yitian Huang, Lexin Zhou, Haotian Li, Xing Xie, and Helen Meng. Social-R1: Towards human-like social reasoning in LLMs, 2026.

Sarah A. Wu, Rose E. Wang, James A. Evans, Joshua B. Tenenbaum, David C. Parkes, and Max Kleiman-Weiner. Too many cooks: Bayesian inference for coordinating multi-agent collaboration. *Topics in Cognitive Science*, 13(2):414–432, 2021. doi: 10.1111/tops.12525.

Tim Wyse, Kaitlin Bustos, Yulia Volkova, and Max Kleiman-Weiner. Commitment to cooperation with self-negotiated contracts. In *Proceedings of the Third Conference on Language Modeling (COLM)*, 2026.

Keyang Xuan, Pengda Wang, Chongrui Ye, Haofei Yu, Tal August, and Jiaxuan You. SocialVeil: Probing social intelligence of language agents under communication barriers. In *Conference on Language Modeling (COLM)*, 2026.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report, 2025.

Yiheng Yao, Chelsea Zou, and Robert D. Hawkins. Talk is cheap, communication is hard: Dynamic grounding failures and repair in multi-agent negotiation, 2026. Accepted at the Conference on Language Modeling (COLM 2026).

Yashwanth YS, Ruichen Wang, Shihua Zeng, Xuhui Zhou, Koichi Onoue, Vasudha Varadarajan, and Maarten Sap. SOTOPIA-TOM: Evaluating privacy and information management in multi-agent interaction with theory of mind. In *Conference on Language Modeling (COLM)*, 2026.

Huining Yuan, Zelai Xu, Zheyue Tan, Xiangmin Yi, Mo Guang, Kaiwen Long, Haojia Hui, Boxun Li, Xinlei Chen, Bo Zhao, Xiao-Ping Zhang, Chao Yu, and Yu Wang. MARSHAL: Incentivizing multi-agent reasoning via self-play with strategic LLMs. In *International Conference on Learning Representations*, 2026.

Erica Zhang, Fangzhao Zhang, Aneesh Pappu, Batu El, Jose Blanchet, Susan Athey, Jiashuo Liu, and James Zou. TERMS-Bench: Diagnosing LLM negotiation agents beyond deal rate, 2026a.

Yuzhe Zhang, Feiran Liu, Yi Shan, Xinyi Huang, Xin Yang, Yueqi Zhu, Xuxin Cheng, Cao Liu, Ke Zeng, Terry Jingchen Zhang, and Wenyuan Jiang. SILO-BENCH: A scalable environment for evaluating distributed coordination in multi-agent LLM systems. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 29379–29398. Association for Computational Linguistics, 2026b. doi: 10.18653/v1/2026.acl-long.1354.

Ruiwen Zhou, Maojia Song, Xiaobao Wu, Sitao Cheng, Xunjian Yin, Yuxi Xie, Zhuoqun Hao, Wenyue Hua, Liangming Pan, Soujanya Poria, and Min-Yen Kan. Epistemic context learning: Building trust the right way in LLM-based multi-agent systems, 2026.

Kunlun Zhu, Hongyi Du, Zhaochen Hong, Xiaocheng Yang, Shuyi Guo, Zhe Wang, Zhenhailong Wang, Cheng Qian, Xiangru Tang, Heng Ji, and Jiaxuan You. MultiAgentBench: Evaluating the collaboration and competition of LLM agents. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8580–8622. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.acl-long.421.

Chelsea Zou, Yiheng Yao, Selena She, Noah D. Goodman, and Robert D. Hawkins. CalBench: Evaluating coordination-privacy trade-offs in multi-agent LLMs, 2026.

A FORMAL GAME SPECIFICATION

This section completes the game definition in Section 2.2. The commitment representation, goal satisfaction functions, and terminal utility follow the action-level negotiation framework of Benac et al. (2026). We specify the information available to each player and the transition rules used in our private-preference setting.

Public instance and private types. The public instance specifies the players, commitment coordinates, goal requirements, scoring modes, proposer schedule, and legal-action rules in Eq. 1. A commitment coordinate belongs to a player; it is not a goal, an item transferred to another player, or an exclusive choice among alternatives. Different goals may depend on the same coordinate. Once that coordinate becomes binding, it contributes to every goal that requires it. A goal's requirements do not indicate which players want that goal, the requirements and preferences are separate parts of the instance.

The preference assignment v is sampled once and remains fixed throughout play. Each player observes its own complete vector v_i. Any publicly specified preferences are common knowledge as well. Write $F = \{(i, g, z)\}$ for these fixed facts and $\mathcal{W} = \mathrm{supp}(\mu_0)$ for the admissible joint assignments. In instances generated from public categorical weights w_z, the prior is obtained by conditioning independent draws of the unfixed entries on the stated generation constraints:

$$\mu_0(v) = \frac{1}{Z}\mathbf{1}[v \text{ agrees with } F]\,\mathbf{1}[v \text{ satisfies the generation constraints}] \prod_{(i,g)\notin \mathrm{dom}(F)} w_{v_{ig}}. \tag{7}$$

Here Z normalizes the distribution. The constraints used in these instances require each player to want at least one goal and each goal to have at least one non-neutral player. The weights and fixed facts are provided with the instance; they are not inferred from the realized world. Conditioning on these constraints can create dependence between preferences, so independent background draws do not imply independent posterior beliefs.

State and decision phases. A full environment state can be written as

$$x = (t, C, p, \mathbf{q}, v, \mathbf{r}, h), \tag{8}$$

where t indexes proposal opportunities, C is the binding commitment matrix, p is either an unresolved offer or $\emptyset$, $\mathbf{q}$ contains remaining investigation budgets, $\mathbf{r} = (r^i)_i$ contains private investigation records, and h is the public event history. This full state is available to the environment, not to every player. When $p = \emptyset$ and $t < T$, the actor is ρ_t. When an offer is pending, its recipient acts instead. The index t does not advance until that offer is accepted or rejected. Thus a proposal and its response count as two decisions but only one proposal opportunity.

Legal offers. At an opportunity of player i, an offer is $o = (j, D_i, D_j)$, where $j \neq i$ and D_i, D_j are sets of new commitment coordinates owned by the two participants. Let m denote the per-player, per-offer addition limit specified by $\mathcal{L}$. Legality requires

$$D_\ell \subseteq \{k : c_{\ell k} = 0\}, \qquad |D_\ell| \leq m \quad (\ell \in \{i, j\}), \qquad |D_i| + |D_j| > 0. \tag{9}$$

Any additional forbidden coordinates specified by the instance are excluded. One participant's addition set may be empty. The limit applies separately to each participant in this offer. Previously binding coordinates cannot be removed or submitted as new additions. The native state representation retains those coordinates when constructing the resulting commitment vectors. The game uses bilateral offers.

Transitions. When no offer is pending, the scheduled proposer may take one of the following actions:

- PASS: append the public action to h, leave C unchanged, and advance t by one.
- OFFER(j, D_i, D_j): append the offer to h and set $p = o$. No commitment binds yet and t is unchanged. The recipient must then choose ACCEPT or REJECT. Acceptance sets $c_{\ell k} = 1$ for $k \in D_\ell$, $\ell \in \{i, j\}$; rejection leaves C unchanged. Either response is public, clears p, and advances t.

- INVESTIGATE(j, g): legal only if $j \neq i$ and $q_i > 0$. The environment appends the public query (i, j, g) to h, privately appends (j, g, v_{jg}) to r^i, reduces q_i by one, and advances t. Commitments are unchanged. Other players' private records are unchanged.

Investigation is allowed even when the queried preference is already known or does not affect the current decision. Its cost is the consumed proposal opportunity and budget. Responses do not consume investigation budgets.

Observations and information sets. All players observe the public rules, fixed preference facts, commitments, proposer schedule, remaining budgets, offers, responses, passes, and query identities. An investigation answer is revealed only to its investigator. Player i's view is $I_s^i = (h_s, v_i, r_s^i)$, with the public instance understood as fixed context. Two histories in the full game belong to the same information set for i when these views agree.

Termination and incentives. A full game starts with $C_0 = 0$, no pending offer, no private query records, and $q_{i,0} = 1$. It ends when all T proposal opportunities have been used, including the response to an offer made at the final opportunity. Utilities are then evaluated by Eq. 2; there are no intermediate payoffs. For a binary goal, partial completion yields zero. For a linear goal, each required coordinate contributes an equal fraction of completion. An unachieved goal contributes zero even if it is wanted or avoided. Each player sums its own signed contributions across all goals. Utilities need not sum to zero, and one player's improvement need not imply either an improvement or a loss for the others.

Starting a slice. A slice records its starting commitments, schedule position, pending offer if any, remaining budgets, and the observations available at that position. A state reached by a legal prefix retains the corresponding public history and each player's own private observations. Only actual observed choices provide behavioral evidence under the reference policy. The reference-policy computation and conditioning conventions are specified in Appendix B.

B GAME SLICES AND ORACLE SOLVER DETAILS

Multi-decision slices and information. Let $\mathcal{W} = \mathrm{supp}(\mu_0) \subseteq \prod_{i \in \mathcal{N}} \Theta_i$ be the finite set of admissible joint preference assignments, with prior μ_0 from Eq. 1. For a segment spanning decisions t through $u-1$, a teacher-side record is

$$\sigma_{t:u} = \big(\mathcal{E}, v, h_t, (a_s, o_{s+1}, m_s)_{s=t}^{u-1}\big), \qquad t < u, \tag{10}$$

where $v \in \mathcal{W}$ is the sampled joint preference assignment, h_t is the starting public history, a_s is the action actually taken, and $o_{s+1} = (o_{s+1}^j)_{j \in \mathcal{N}}$ records what each player observes afterward. The flag m_s records whether a_s is an autonomous player action or an externally imposed move; only autonomous partner actions contribute behavioral likelihood. The sampled world determines private investigation answers and terminal payoffs, whereas the actions are generated by the acting players or specified by an external starting context; v alone does not determine the action sequence. Both proposals and responses count as decisions, so $u = t + 1$ gives a single-decision slice. Here the decision indices t, u are distinct from the proposal-opportunity index in Appendix A.

The public history h_s determines commitments, the pending offer, the proposer schedule, and remaining investigation budgets. Player j additionally knows only v_j and its own private answers r_s^j. A slice can contain decisions by several players, each labeled from that player's information set I_s^j; the sampled world is never disclosed as part of this view. The observed sequence selects decision points to label, while the oracle retains all worlds in the prior's support and all legal terminal continuations from each such point.

Finite-tree policy selection. A policy must assign the same action probabilities in any two worlds that a player cannot distinguish. We expand the public action tree for every $v \in \mathcal{W}$, retaining world-dependent private investigation answers. For a policy profile π, its terminal utility is $V_i^\pi(n, v) = U_i(C_T(n); v_i)$. At a nonterminal node n whose actor is j, the continuation values satisfy

$$V_i^\pi(n, v) = \sum_{a \in \mathcal{L}(n)} \pi_j(a \mid I^j(n, v))\, V_i^\pi(n[a, v], v), \tag{11}$$

where $n[a, v]$ is the successor after action a in world v. The private answer to an investigation affects the investigator's later information set, not the public information available to everyone else. Here $\mathcal{L}(n)$ denotes the legal actions at node n.

In Eq. 3, BR_i is a best response over i's *complete contingent policy*: it can revise later decisions after every possible observation, not merely the next action. We compute it by backward induction while keeping the other players' policies fixed. Since the plan can depend on i's initial private information, the maximization separates across root information classes with positive prior probability. At node n in world v, define the counterfactual reach weight

$$\lambda_{-i}(n, v) = \mu_0(v) \prod_{\substack{r \prec n:\, j_r \neq i \\ r \text{ autonomous}}} \pi_{j_r}(a_r \mid I_r^{j_r}(v)), \tag{12}$$

where $r \prec n$ ranges over earlier decisions on the path to n and j_r is the player acting at decision r. Conditioned on i's information set, these weights determine the expected value of each candidate action. The product omits i's own past choices, so an alternative plan remains evaluable even when it differs from i's current plan; imposed moves contribute no behavioral likelihood. At a zero-reach information set, we use the root prior restricted to i's own type and private investigation answers. We first maximize expected own terminal utility. Only at an ACCEPT/REJECT response do own-utility ties favor the action with greater expected total utility to others. Residual ties receive equal probability.

The updates for all players read the same previous profile and are applied simultaneously. A fixed profile is accepted only if, for every player and root private-information class,

$$\max_{\widetilde{\pi}_i \in \Pi_i} \mathbb{E}_{(\widetilde{\pi}_i, \pi^\star_{-i})}[U_i(C_T; v_i) \mid I^i_{\mathrm{root}}] - \mathbb{E}_{\pi^\star}[U_i(C_T; v_i) \mid I^i_{\mathrm{root}}] \leq \varepsilon, \tag{13}$$

where Π_i contains legal information-respecting contingent policies and ε is the numerical tolerance. The check evaluates a full unilateral deviation, rather than only a one-step alternative. We discard cycles, timeouts, and profiles that fail the check. The resulting $\pi^\star$ is a selected reference profile, with no claim of uniqueness or of describing arbitrary human or LLM partners.

Belief filtering and action values. Let $b^{i,\star}_s(v_{-i})$ be the joint posterior over all other players' types at one decision in the slice. If player $j \neq i$ autonomously takes public action a^j_s, define $\ell^\star_s(v_{-i}) = \pi^\star_j(a^j_s \mid I^j_s(v_i, v_{-i}))$. The candidate world determines what j privately knows, including answers to its own investigations. The exact update is

$$b^{i,\star}_{s+1}(v_{-i}) = \frac{b^{i,\star}_s(v_{-i})\ell^\star_s(v_{-i})}{\sum_{\widetilde{v}_{-i}} b^{i,\star}_s(\widetilde{v}_{-i})\ell^\star_s(\widetilde{v}_{-i})}. \tag{14}$$

An action by i or an imposed move changes the game state but contributes no likelihood about another player's type. A privately received investigation answer instead removes worlds inconsistent with that answer. We exclude observed actions whose total likelihood is zero. For every legal action at a selected decision, we force that action once and then follow the fixed reference profile:

$$Q^\star_i(I^i_s, a) = \sum_{v_{-i}} b^{i,\star}_s(v_{-i}) V^{\pi^\star}_i(n_s[a, (v_i, v_{-i})], (v_i, v_{-i})). \tag{15}$$

For a queried partner–goal preference v_{jg}, the inference target is the positive-support set of $p_{jg}(z) = \sum_{v_{-i}: v_{jg} = z} b^{i,\star}_s(v_{-i})$. We report the mode as the most likely preference only when its posterior probability exceeds the runner-up by more than 0.1; otherwise, we report UNDETERMINED. Planning actions are evaluated using $Q^\star_i$, including ties among optimal actions. The student sees neither the exact posterior nor action values. We use the same selected policy throughout a slice, while each player's posterior incorporates only its own observations. The stronger guarantee that a qualitative judgment alone determines the best action is certified separately for the selected partner-history diagnostic in Appendix C.

C DIAGNOSTIC PROTOCOL AND DETAILED RESULTS

Cases and evidence. The main diagnostic uses 22 two-player cases spanning seven distinct game geometries. Its 12 basic cases comprise six direct feedback, three single-choice elimination, and

three ambiguous single-choice cases, with two or three binary goals per basic case. Ten challenge cases comprise three additional elimination cases, three additional ambiguous cases, and four multi-step updates. Cases sharing a geometry are related probes rather than independent draws from a task population.

Initial commitments are declared starting conditions, not fictitious partner choices. Direct feedback comes from a legal focal-player investigation with a truthful private answer. That query is an intervention and need not be optimal; it does not supply behavioral evidence about the partner. Other cases use observed choices with positive probability under the specified partner policy. Labels condition on public facts, own preferences, private observations, and the relevant behavioral evidence.

Judgments and planning sufficiency. Each case queries exactly one unknown partner–goal preference, with candidate values WANT, NEUTRAL, and AVOID. The oracle judgment contains its positive-probability support and the most likely preference. We report the latter only when its posterior probability exceeds the runner-up by more than 0.1; otherwise, we report UNDETERMINED. This label does not imply equal posterior probabilities. We report joint correctness, requiring both components to match the oracle label. All labels are relative to the public prior and specified reference policy.

Planning uses the standard history-free training renderer, including tables of goal requirements and preferences and qualitative supplied beliefs. Behavioral history and numerical posterior probabilities are omitted. For each oracle judgment, exact rational polytope-vertex checks verify that all compatible posteriors induce the same optimal-action set. The admission checks also verify the scorer's tolerance-based acceptable-action set and recompute continuation payoffs from the current state without the removed history. Every retained case passes these checks. Correctness follows the certified acceptable-action set; regret uses the exact expected-utility gap. The scorer allows an own-utility gap of at most 0.1 from the best action. At offer responses, it additionally allows a gap of at most 0.1 in the other players' total utility relative to actions tied in own utility. The oracle policy itself uses exact utility maximization and the response tie rule; these scoring tolerances do not alter that policy.

For each possible partner preference, the qualification procedure computes the acceptable-action set under the same per-world payoff table. A case is repair-sensitive if at least one such set is disjoint from the set accepted under the oracle judgment. The remaining cases are labeled action controls. This label denotes the complement of the repair-sensitive group, not a negative control for the model's response to its input. There are nine repair-sensitive cases and thirteen controls. All four multi-step cases are controls and each admits an acceptable action shared across all three preferences, they therefore do not require a belief-dependent change of action.

Requests and execution. Every case has four calls: inference, direct action selection, $\mathrm{Plan}_{\mathrm{oracle}}$, and $\mathrm{Plan}_{\mathrm{model}}$. The two planning requests share the current state, tools, decoding settings, and request seed; only the supplied judgment differs. Inference requests require a brief explanation followed by one belief-submission tool call, with explicit set semantics. Action calls use the native action tools. The intervention changes an explicit input between fresh calls rather than manipulating an internal model representation.

The reported run evaluates Qwen3-4B-Instruct-2507 once per case, at temperature zero, top-$p = 1$, top-$k = -1$, and a 4,096-token completion limit. One $\mathrm{Plan}_{\mathrm{model}}$ call has a format failure. It counts as incorrect for accuracy and is excluded from paired utility comparisons, leaving 21 valid pairs. No repeated-run uncertainty estimate is claimed.

Aggregation and intervention accounting. The repair-sensitive and control panels are reported separately. Overall counts are descriptive: joint inference correctness is $11/22$; direct action selection, $\mathrm{Plan}_{\mathrm{oracle}}$, and $\mathrm{Plan}_{\mathrm{model}}$ are each correct in $11/22$. Basic-case joint inference and $\mathrm{Plan}_{\mathrm{oracle}}$ correctness are $9/12$ and $7/12$, versus $2/10$ and $4/10$ on challenge cases.

Across 21 valid pairs, three replacements increase utility, seventeen leave it unchanged, and one reduces it. Among the ten pairs with a changed judgment, these counts are three, six, and one, respectively. There are nine action changes, including five that preserve utility. All eleven pairs with unchanged judgments reproduce the same action. Mean paired regret falls from 0.684 to 0.541, a value gain of 0.143. The all-case $\mathrm{Plan}_{\mathrm{oracle}}$ mean regret is 0.567 and uses a different denominator.

Table 5: Diagnostic subgroups defined by the oracle action sets. Entries are correct responses out of all cases; invalid outputs count as incorrect. The groups partition the same 22 cases as Table 1.

	Inference		Action	
Case group	Joint	Direct	$\mathrm{Plan}_{\mathrm{model}}$	$\mathrm{Plan}_{\mathrm{oracle}}$
Repair-sensitive	5/9	3/9	3/9	3/9
Action control	6/13	8/13	8/13	8/13

For each pair, the value gain equals model-judgment regret minus oracle-judgment regret under the same scoring posterior. These effects can have either sign and are not an additive decomposition of the direct action-selection error.

The repair-sensitive panel has one positive and seven zero gains among its eight valid pairs. The control panel has two positive, ten zero, and one negative gain among thirteen pairs. In the single-choice elimination layer, mean repair gain is 0.667 across six cases, but two of its three positive gains occur in controls. This distinction prevents improvements in controls from being presented as evidence that the task requires preference-dependent action selection. Of the eight valid repair-sensitive pairs, five already have the correct judgment. Among the three with an incorrect judgment, two actions change; one gains 1 in utility and two have zero gain. The mean gain is therefore 0.333 over corrected judgments and 0.125 over all valid pairs in this group. In the control group, seven judgments are incorrect; all seven actions change, with two gains, four ties, and one loss. The corresponding means are 0.286 and 0.154 over corrected judgments and all thirteen valid pairs, respectively.

C.1 Two-goal example: ruling out an avoided goal

Table 2 uses case starting with no commitments, Blair acts first and Alex acts last; neither has an investigation use remaining. The prior over Blair's Orchard preference is uniform. Recomputing the reference policy gives likelihoods $(1/6, 1/9, 0)$ for the observed offer under Want, Neutral, and Avoid. Alex accepts with probability one, so the posterior is $(0.6, 0.4, 0)$ and the oracle judgment is Want or Neutral, with Want most likely. Excluding Avoid has the simple payoff argument given in the main text; the unequal weights on the two remaining preferences follow from the reference policy's uniform mixing over tied actions.

The model instead states, "Blair's behavior (not adding Cedar) suggests they do not want Orchard," and submits only `avoid`. Its planning call requests `self_commitments=[]` and `partner_commitments=[Maple]`, completing no goal regardless of Blair's response. With the oracle judgment, the call requests `self_commitments=[Maple]` and `partner_commitments=[Cedar]`. Blair accepts, completing both goals and giving Alex 2. The two offers have regrets 2 and 0, respectively, giving an intervention gain of 2. The latter offer is also acceptable if Blair avoids Orchard: its own utility then ties rejection at 0, and the response tie rule favors Alex's gain. This explains the case's action-control classification and limits the conclusion to the observed benefit of correcting the model's supplied judgment.

C.2 A policy-dependent example of partial correction

This additional example has two proposal opportunities which are Blair followed by Alex, starting with no binding commitments. Each offer can add at most one commitment per player, and commitments cannot be removed. All four goals are binary, with requirements and preferences as follows. Only Blair's preference for Orchard is hidden; its prior is uniform over Want, Neutral, and Avoid.

Goal	Alex must add	Blair must add	Alex's preference	Blair's preference
Orchard	Maple	Maple	Want	Hidden
Harbor	Cedar	Maple	Want	Want
Library	Maple	Maple	Neutral	Avoid
Garden	Cedar	Cedar	Avoid	Avoid

Observed history and oracle judgment. Blair proposes no commitment for itself and Cedar for Alex; Alex accepts. To evaluate this choice as evidence, the oracle compares its continuation value

with the other actions Blair could have taken. Under the selected reference policy, Alex's subsequent proposal adds Maple for both players. If Blair wants Orchard, accepting completes Orchard, Harbor, and Library, giving Blair $1 + 1 - 1 = 1$. If it is neutral, accepting gives $0 + 1 - 1 = 0$; rejection also gives 0, so the response tie rule favors acceptance for Alex's benefit. If Blair avoids Orchard, accepting gives $-1 + 1 - 1 = -1$, so it rejects and receives 0.

Blair has an alternative first offer: add Maple for itself and Cedar for Alex. Alex accepts, immediately completing Harbor. On Alex's last turn, adding its Maple would also complete Orchard and Library. Blair accepts this addition if it wants Orchard and otherwise rejects it. Thus the alternative secures Blair terminal utility 1 under every preference. The observed offer is optimal only under WANT, where all thirteen legal first actions tie and the reference policy mixes uniformly. Its likelihoods are therefore

$$\begin{aligned} \Pr(\text{observed offer} \mid \textsc{Want}) &= \tfrac{1}{13}, \\ \Pr(\text{observed offer} \mid \textsc{Neutral}) = \Pr(\text{observed offer} \mid \textsc{Avoid}) &= 0. \end{aligned} \tag{16}$$

Alex accepts the observed offer with probability one. Bayes' rule maps the prior $(1/3, 1/3, 1/3)$ to posterior $(1, 0, 0)$, yielding support $\{\textsc{Want}\}$ and most likely preference WANT. This inference follows from the stated reference policy and continuation, rather than from a general assumption that any offer reveals a unique preference.

Recorded model responses and their consequences. The model's inference response instead states, "There is no evidence suggesting Blair wants or avoids Orchard," and submits all three values with the most likely preference reported as UNDETERMINED. With that judgment, its planning tool call requests `self_commitments=[Maple], partner_commitments=[Cedar]`. Accepting would complete Garden, giving Blair -1, so Blair rejects; Alex's terminal utility is 0. With the oracle judgment, the tool call changes to `self_commitments=[], partner_commitments=[Maple]`. Blair accepts because Harbor gives it 1; Alex also receives 1. The optimal call would add Maple for both players, completing Orchard, Harbor, and Library. Its utilities are $1 + 1 + 0 = 2$ for Alex and $1 + 1 - 1 = 1$ for Blair, so Blair accepts. The three oracle action values are therefore 0, 1, and 2, respectively. The two recorded planning calls have regrets 2 and 1, giving intervention gain $\Delta_c = 1$.

D TRAINING DATA AND CONFIGURATION

Slice training uses 100 game slices, and self-play starts from 100 initial game configurations. All training variants use a budget of approximately 6M generated response tokens, and we evaluate the final checkpoint of each training run. Tables 6 and 7 report the settings for slice training and self-play, respectively.

Table 6: Optimization and decoding settings for slice training.

Parameter	Setting
Shared slice-training configuration	
Initial model / KL reference	Qwen3-4B-Instruct-2507 / frozen initial model
Precision	BF16; gradient accumulation and reduction in FP32
Optimizer	Adam, Megatron backend
Initial learning rate	10^{-6}
Adam coefficients; weight decay	$(0.9, 0.95)$; 0
Policy updates per rollout batch	1 epoch
Policy-ratio clipping	0.2
KL loss coefficient	0.01
Loss aggregation	Mean over tokens within each response, then over responses
Sequence / prompt limit	4096 / 3072 tokens
Maximum generated response	1024 tokens per call
Sampling	Temperature 1; top-$p = 1$; top-k disabled
Repetition penalty	1
Responses per question group	8

Continued on next page

Table 6 (continued)

Parameter	Setting
Invalid or truncated output	Independent negative advantage, coefficient 0.2
Random seed	42

Self-play training. All players act using the current policy, with each receiving only its own observation and available interaction history. Training collects complete games and assigns each player's terminal utility to that player's decisions. The environment supplies the reward; oracle judgments and action labels are not used. Four stochastic trajectories are collected for each sampled initial game, with returns associated with the same initial game and player seat. The number of decisions varies across trajectories, so collection is governed by generated response tokens rather than a fixed number of decision items.

Table 7: Self-play training settings. Each rollout covers a complete game; all player roles share the current policy.

Parameter	Setting
Initial model / KL reference	Qwen3-4B-Instruct-2507 / frozen initial model
Trajectories per sampled initial game	4
Task reward	Each player's own terminal utility
Precision	BF16; gradient accumulation and reduction in FP32
Optimizer	Adam, Megatron backend
Initial learning rate	10^{-6}
Adam coefficients; weight decay	$(0.9, 0.95)$; 0
Policy updates per rollout batch	1 epoch
Policy-ratio clipping	0.2
KL loss coefficient	0.01
Sequence / prompt limit	4096 / 3072 tokens
Maximum generated response	1024 tokens per decision call
Sampling	Temperature 1; top-$p = 1$; top-k disabled
Repetition penalty	1
Invalid or truncated output	One retry; independent negative advantage with coefficient 0.2
Random seed	42

D.1 Structural criteria for slice selection

Strategic coverage concerns the relationship between available evidence, partner responses, and the consequences of a decision. We distinguish eligibility checks, verified relations between candidate decisions, and structural coverage dimensions (Table 8). These criteria describe candidate construction and coverage; they are not an exhaustive specification of subsequent training-pool selection. Coverage is defined across the collection, rather than requiring every slice to exhibit every property.

For a decision information set I, let $\mathcal{A}_\epsilon(I)$ denote the legal actions accepted by the oracle scorer, retaining all actions within the utility tolerance $\epsilon = 0.1$ and applying the stated response tie-breaking rule. Action-supervision eligibility requires $\varnothing \neq \mathcal{A}_\epsilon(I) \subsetneq \mathcal{L}(I)$. This excludes decisions with no reliable acceptable action or with no distinction between acceptable and unacceptable choices. It does not require a unique best action, and it is not a filter on inference-only questions.

For an information contrast (I, I'), disjoint acceptable-action sets, $\mathcal{A}_\epsilon(I) \cap \mathcal{A}_\epsilon(I') = \varnothing$, certify that no single action is acceptable in both conditions. Such contrasts need not be consecutive decisions: a contiguous slice follows one legal trajectory, whereas counterfactual branches provide related supervision items. Cross-turn and multi-party coverage are structural descriptions; neither a longer remaining horizon nor a larger player count alone certifies that short-horizon or single-partner reasoning would fail. We do not impose such an ablation-based requirement on every selected slice.

Table 8: Structural selection criteria and coverage dimensions. Verified contrasts are defined under the reference policy; planning coverage describes dependencies in the game structure.

Criterion	Check or coverage target
Decision validity	Replay the history under the native rules, verify the current state and legal actions, and retain the acting player's available information.
Action discrimination	Require a nonempty acceptable-action set that is a proper subset of the legal actions for scored action tasks.
Judgment update or preservation	Compare oracle judgments at linked histories. Include both informative observations that change the judgment and observations for which it remains unchanged. Attribute private feedback to the corresponding investigation.
Information–action dependence	Include controlled information contrasts with different acceptable actions, as well as action-invariant controls. For certified must-change pairs, the acceptable-action sets are disjoint while continuation payoffs are held fixed.
Cross-turn planning	Cover decisions where irreversible commitments or spending an opportunity on investigation affect later feasible choices. Evaluate actions through their reference-policy continuation to terminal utility.
Multi-party planning	Cover goal and commitment dependencies across players, including agreements whose consequences depend on several partners' responses. Players still pursue their own utility.
Label reliability	Exclude failed reference computations. For planning with a qualitative judgment, retain reliable action supervision and mask labels that remain ambiguous under the supplied information.
Coverage sampling	Favor underrepresented combinations of evidence type, proposal/response/investigation decisions, remaining decision stages, and preference-sensitive or insensitive cases.

D.2 Illustrative training prompts

The templates below summarize the instructions and information supplied to the model. They are explanatory renderings of the training prompts, with instance-specific content replaced by bracketed placeholders. We use the paper's terminology for preference judgments rather than implementation field names. Slices are executed with the oracle, which advances the game and updates each player's information along the trajectory. The resulting decision points supply the model's supervision requests. The three templates describe supervision views of those decisions; a model-generated inference response is not automatically passed to the planning view, which receives the oracle judgment.

Shared context and response format. Each slice prompt specifies the acting player, game rules, goal requirements, own preferences, binding commitments, pending offer (if any), remaining proposal order, and investigation budget. Goal requirements describe the commitments needed for completion, not players' preferences. The prompt also states the reference partner policy, including its response tie-breaking rule. Legal actions and their arguments are supplied through tool schemas. The shared system instruction is:

> You are answering a reasoning question about a negotiation game. Use the stated rules and visible information. Briefly explain your answer, then make exactly one requested tool call.

Direct action selection. The model receives the public history and its own private observations, together with the public preference prior. No oracle judgment is supplied.

> **Context:** [Game rules, goals, own preferences, public prior, partner policy, visible interaction history, private investigation answers, and current decision state.]
>
> **Task:** Choose your next action. Pursue the goals you want and avoid bringing about goals you dislike. When accepting and rejecting are equally good for you, choose the response that helps the other players more.
>
> **Response:** Briefly explain your decision, then make exactly one legal game-action tool call. Keep the explanation outside the tool arguments.

Inference. This request uses the available evidence to query a particular partner's preference for a particular goal. Its output is a judgment, not a game action.

> **Context:** [Game and decision information, public prior, partner policy, visible history, and private observations.]
>
> **Task:** Assess [partner]'s preference for [goal]. Use the observed behavior and revealed information under the stated partner policy. Report every preference still possible, including unlikely ones, and the most likely preference, or `undetermined` when no preference is clearly better supported under the specified reporting rule.
>
> **Response:** Briefly explain, then make exactly one `SUBMIT_BELIEFS` call containing the requested judgment.

Planning given an oracle judgment. Here, the oracle judgment is an input; the model must choose an action conditioned on that judgment. The template shows the planning view that supplies the current decision state and qualitative judgments directly, without requiring reconstruction of the evidence history.

> **Context:** [Game rules, goals, own preferences, partner policy, current decision state, and legal actions.]
>
> **Supplied oracle judgment:** [For each specified partner–goal pair: possible preferences and most likely preference, or `undetermined`.]
>
> **Task:** Use the supplied judgment directly to choose your next action. Pursue your own goals under the stated continuation policy, including its response tie-breaking rule.
>
> **Response:** Briefly explain your decision, then make exactly one legal game-action tool call.

Both action tasks also state the scoring tolerance: an action may lose at most 0.1 utility relative to the best action; for responses tied in own utility, the corresponding allowance in other players' total utility is 0.1. The acceptable-action labels remain with the scorer and are not included in the model's input.

Self-play. At each decision in a complete game, the acting player receives the game rules, its own preferences and private observations, the public history, and the current state and legal tools. Its instruction is summarized below.

> Your objective is to maximize your own expected final score. Other players' preferences may differ from yours. Briefly explain your decision using the information available to you, then make exactly one legal game-action tool call. Your explanation is private and is not shown to other players.

The selected action advances the game, and the next player receives its own observation. Unlike the slice tasks, these requests contain no supplied oracle judgment or reference-policy action target; training feedback comes from the player's terminal utility.

E SAME-MODEL NEGOTIATION EVALUATION

Instances and execution. Each team size has 16 frozen instances. The two- and four-player suites cross binary/linear scoring, two/four proposal opportunities per player, and balanced/avoid-heavy priors, with two instances per combination. Prior weights for WANT, NEUTRAL, and AVOID are $(1, 1, 1)$ and $(1, 1, 2)$, conditioned on the native generation constraints. The three-player suite retains its existing 16 instances. All games have three goals and two commitment coordinates per player.

Metrics and incomplete games. Team utility is $\sum_i U_i(C_T; v_i)$, averaged over the full suite. For GAMESLICE-A's sole incomplete three-player rollout, Table 3 uses a conservative team-utility lower bound of 0.5 derived from its last valid state's irreversible commitments, retaining all 16 games in the denominator and yielding a lower-bound mean of 0.911. Negative-outcome counts below use completed games only and explicitly show their player-level denominators. They measure negative terminal utility. A legal terminal state is not a cooperation-success criterion, and completing more goals need not improve signed utility.

Table 9: Negative-utility players among completed games. Counts expose individual costs that can be obscured by aggregate team utility.

Model	Two players	Three players	Four players
Baseline	5/32	5/48	11/64
Self-play	3/32	5/48	11/64
GAMESLICE-A	2/32	15/45	14/64
GAMESLICE-AP	4/32	8/48	12/64
GAMESLICE-JOINT	2/32	3/48	10/64

F CALBENCH EVALUATION AND ROLLOUT COMPARISON

F.1 ENVIRONMENT AND EVALUATION SETTINGS

Task and private information. CalBench (Zou et al., 2026) models personal assistants coordinating a stream of meeting requests on behalf of different users. Each agent controls one private calendar with discrete time slots. A slot can be free, contain a movable errand or a previously scheduled meeting, or be blocked and therefore unavailable for either scheduling or rescheduling. The agent sees its own calendar, item-specific displacement costs, the current meeting request, and messages delivered to it; it cannot inspect another agent's calendar. Agents are instructed to minimize their own user's cumulative displacement cost while coordinating feasible meetings and avoiding disclosure of exact costs or unnecessary private details. Thus a mutually feasible slot may require one user to move an errand or several users to move a shared meeting, creating a need to negotiate who accommodates whom. Scheduling a new meeting itself incurs no displacement cost; moving an existing item does.

One meeting round. Each incoming request identifies a meeting and its participants. The calendar state persists across rounds, and a verbal agreement alone does not change it. Our evaluation retains the native four-phase interaction:

1. **Communication.** Agents exchange private direct messages to propose slots, explain constraints, and negotiate calendar changes. A participant may also contact an agent outside the current meeting when an earlier shared commitment needs to move.
2. **Voluntary rescheduling.** A non-participant who received a message during the round can submit changes to its own calendar. This phase allows an earlier meeting's participants to help relocate it even when they are absent from the new meeting.
3. **Independent decisions.** Each current participant submits a batch of `schedule` and `reschedule` actions for its own calendar. A reschedule identifies the item, its source and destination slots, and a justification. The batch is validated as a whole: the new meeting must occupy a free slot after the proposed moves, and blocked items cannot be moved or overwritten. No agent can directly edit a partner's calendar.
4. **Resolution.** The environment validates the submitted actions and checks agreement across calendars. All participants must place the new meeting in the same slot, and any relocated meeting must remain consistent across its original participants. Agreement in messages is therefore insufficient if the agents submit incompatible calendar actions.

Our sequential evaluation suite. We evaluate homogeneous teams: the same model controls all four agents in a game. Each game has eight slots and three single-slot meetings, with participant sets $\{0, 1, 2\}$, $\{1, 2, 3\}$, and $\{0, 2, 3\}$ in that order. There are no preloaded meetings; shared commitments arise from the agents' own earlier decisions. Future requests and evaluator reference solutions are withheld from agents. The frozen suite crosses four conditions with two errand-cost regimes and three scenario seeds, giving $4 \times 2 \times 3 = 24$ games:

- **Loose availability:** generator density 0.6, with movable errands and no blocked items, tests scheduling with spare calendar capacity.
- **Dense calendars:** generator density 1.0, with no blocked items, increases the need to accommodate existing commitments. Density is applied to residual capacity after reserved slots, rather than to the raw fraction of occupied slots.

- **Blocked errands:** density 1.0 and two immovable items per agent combine hard availability constraints with movable errands.
- **Replanning control:** an authored calendar configuration makes an earlier meeting's placement consequential for a later request. Placing the first meeting at slot 0 requires moving it to slot 1 when the second request arrives, including contacting agent 0, who is absent from that request. Placing it at slot 1 initially avoids this branch; successful completion does not by itself demonstrate replanning.

Movable errands cost 1 to relocate in the uniform regime and have integer costs from 1 to 8 in the varied regime. Moving an existing meeting costs 1 per participant. The three scenario seeds change private occupancy and costs, rather than merely repeating stochastic model decoding. The replanning controls admit a zero-cost solution in both cost regimes. All cases are verified as feasible by reference schedules and native sequential replay. This suite is a small-model adaptation of CalBench; we report coordination outcomes rather than reproduce its original aggregate benchmark score. Although the native prompts retain privacy instructions, we do not measure privacy leakage, and message volume is used only as a communication metric.

Inference settings and comparison models. All models use the same cases, direct messages only, temperature zero, a 4,096-token output limit, and a 32,768-token context limit. Each meeting allows four communication turns per participant and one decision retry; fallback and reflection are disabled. The Qwen3-4B thinking model, MARSHAL Generalist (Yuan et al., 2026), and Social-R1-4B (Wu et al., 2026) form a separate comparison group because they use thinking-mode checkpoints; *Baseline* denotes the unchanged Qwen3-4B-Instruct-2507 model, and *Self-play* denotes its complete-game self-play-trained counterpart. The slice-trained variants are GAMESLICE-A (direct action selection), GAMESLICE-AP (direct action selection and conditional planning), and GAMESLICE-JOINT (all three objectives), initialized from the same instruct model. These names match Table 4.

Metric definitions. A meeting is retained only if it remains in one common slot for all participants in the final calendars. A game is successful only if all three meetings remain valid and blocked items are preserved. These counts have maxima of 72 meetings and 24 games. For a successful game g, per-agent excess cost is $(C_g - C_g^\star)/4$, where C_g is the realized team rescheduling cost and $C_g^\star$ is the minimum full-stream cost, established by the lower-bound and attainability argument below. We average this quantity over successful games only, so models may be evaluated on different subsets of games. The cost reference has access to the full scenario after the fact and is not an online policy.

Minimum-cost reference. Every pair of meetings shares a participant, so their final slots must be distinct. We enumerate all $8 \cdot 7 \cdot 6 = 336$ assignments of the three meetings to distinct slots, excluding assignments that occupy a participant's blocked slot. For each remaining assignment, the sum of the displacement costs of initial errands occupying its meeting slots is a lower bound: each such errand must move at least once. Minimizing over these assignments therefore gives a lower bound on every successful execution. For every evaluation scenario, a native sequential execution attains this bound, establishing $C_g^\star$. Replay proves attainability; enumeration supplies the lower bound needed for optimality. This reference uses all three requests in hindsight, whereas agents receive them sequentially.

Communication overhead. Let M_{ig} be the number of direct messages sent by agent i throughout game g, including messages during unsuccessful rounds and messages sent as a non-participant. Received messages are not counted. Let R_{ig} be the number of meeting rounds in which i participates and the round resolution declares the meeting successfully coordinated. These are round-level successes, not finally retained meetings: a later inconsistency does not remove an earlier success from this denominator. We report

$$\mathrm{Comm} = \frac{1}{24}\sum_{g=1}^{24}\frac{1}{4}\sum_{i=1}^{4}\frac{M_{ig}}{\max(R_{ig},1)}. \tag{17}$$

When $R_{ig} = 0$, the agent contributes its full sent-message count, including zero if it sent nothing. All games enter this average, regardless of final success.

Burden imbalance. Let S_g contain meetings declared coordinated at their respective round resolutions, and let c_{ig} be agent i's total realized displacement cost over the entire game, including costs incurred during unsuccessful rounds. The native assignment solver is run on the initial calendars for S_g. Its selected reference assignment is replayed in request order to obtain per-agent reference costs $r_{ig}(S_g)$. This is the cost allocation of the selected team-cost reference, not each agent's independently minimized cost or an equal share of the team cost. Define

$$b_{ig} = c_{ig} - r_{ig}(S_g), \qquad \bar{b}_g = \frac{1}{4}\sum_{i=1}^{4} b_{ig}, \qquad \text{Imbalance} = \frac{1}{24}\sum_{g=1}^{24}\frac{1}{4}\sum_{i=1}^{4}|b_{ig} - \bar{b}_g|. \tag{18}$$

Individual excess burdens may be negative and are not clipped. Failed games remain in the average with their realized costs and their own S_g; if S_g is empty, all reference costs are zero. The implementation also uses zero reference costs if the selected assignment cannot be replayed. Thus imbalance uses a round-success-subset reference, whereas the reported excess-cost column uses the full-stream reference and successful games only.

F.2 Rollout comparison with verbatim model outputs

The following excerpts come from the frozen `replan_varied_s1` scenario. Baseline, Self-play, and GameSlice-Joint received identical initial calendars. Slot tuples below come from the environment's round resolutions, ordered by the meeting's participants, rather than from model explanations.

Negotiating the first meeting. GameSlice-Joint's agent 0 first proposed slot 2. Agent 1 replied:

> "Hi, slot 2 is blocked for me, so I can't use it. Slot 1 is free for me and would be a good alternative. Would that work for you?"

Agent 0 then asked agent 2:

> "Hi, Agent 1 suggested slot 1. Is that free for you? I'm open to slot 1 if it works for both of you."

Agent 2 confirmed slot 1, and the three agents' scheduling actions all chose slot 1; the resolution tuple was $(1, 1, 1)$.

A disagreement in the other runs. Baseline and Self-play both placed the first meeting at slot 0. During negotiation for the second meeting, Baseline's agent 3 reported to agent 1:

> "Slot 1 is blocked for me, so I can't use it. How about slot 2? It's free and doesn't conflict with my schedule. Would that work for you?"

Baseline's agent 1 later responded:

> "I see slot 1 is blocked for you, but it's free for me. I don't have any other free slots available. Could you please confirm if you're able to adjust your schedule to slot 1? I'm happy to help coordinate if needed, but we need a shared free slot."

Self-play's agent 1 made the corresponding request to agent 3:

> "I see you're suggesting slot 2, but it's blocked for me and cannot be used. Slot 1 is free and available for me without any conflict. Would you be able to join us there? If not, let me know what alternative works best for you."

In both runs, the second meeting's resolution tuple was $(1, 2, 2)$, so no common slot was established. Baseline's agent 1 actually returned an empty action list at this decision; its resolution entry should not be read as an intentional slot-1 commitment.

Preserving a shared commitment. In the GAMESLICE-JOINT run, the second meeting was placed at slot 0 by all three participants. When agent 0 proposed slot 0 for the third meeting, agent 2 replied:

> "Slot 0 is taken by another meeting. I suggest slot 2 instead, which is free and doesn't conflict with any of my commitments. Would that work for you?"

Agent 3 also identified the prior meeting at slot 0. The final resolution tuple was $(2, 2, 2)$, and all three meetings remained valid. This trajectory shows coordination around an existing commitment; it contains no action that reschedules an earlier meeting.