

Talk is Cheap, Communication is Hard: Dynamic Grounding Failures and Repair in Multi-Agent Negotiation

Yiheng Yao Robert D. Hawkins
Stanford University
yaoyh@stanford.edu

Abstract

Grounding requires agents to establish and maintain the mutual understanding needed to coordinate over multiple turns. Yet multi-agent LLM benchmarks often focus on static, one-shot tasks rather than interactive settings in which agents must negotiate and repair shared understanding over time. We introduce an iterated negotiation game with private project values, shared resource constraints, and verifiable joint optima. Individual agents identify optimal allocations in isolation, yet dyads frequently miss exact joint optima when private objectives conflict. Analysis of game traces reveals four recurrent patterns: loss of shared interaction history, anchoring to early proposals, equal-split shortcuts, and abandonment of public agreements at execution. We then vary partner context, information transparency, and incentive framing. Full transparency improves coordination without reliably improving exact-optimum attainment, while explicit joint-reward framing yields gains across several dyads. Our findings show that interaction structure substantially affects dyadic performance: what agents know about one another and whether their incentives are explicitly aligned both influence coordination. Nonetheless, reliable coordination remains difficult when agents bargain over conflicting private objectives. These results identify common-ground formation, joint-plan maintenance, and commitment execution as central targets for improving decentralized multi-agent coordination.

1 Introduction

Successful communication requires more than exchanging propositions; it requires *grounding*—the collaborative process by which interlocutors establish that their utterances have been understood well enough for current purposes (Clark & Brennan, 1991). In human dialogue, grounding is an active, incremental process: speakers monitor comprehension, request clarification, and repair misunderstandings in real time. This *dynamic* grounding stands in contrast to *static* grounding, which links language to a fixed, externally observable context such as a knowledge base or image (Chandu et al., 2021; Anikina et al., 2025). Dynamic grounding is central to real-world coordination problems where agents must allocate shared resources under partial information. For example, project managers coordinating engineering hours, compute, and testing bandwidth across concurrent workstreams must explain priorities, negotiate trade-offs, and revise plans over repeated meetings. As AI systems increasingly act on users’ behalf, these coordination demands become questions of faithful representation: an assistant must preserve a user’s goals, constraints, and commitments while negotiating shared plans with other agents.

Current multi-agent LLM benchmarks largely focus on static tasks: matching descriptions to images (Eisenstein et al., 2026), optimizing fixed payoff matrices (Bianchi et al., 2024), or agreeing on predefined prices (Davidson et al., 2024). These settings test whether models can exchange facts, but not whether they can *build and maintain shared understanding through interaction*, the hallmark of dynamic grounding.

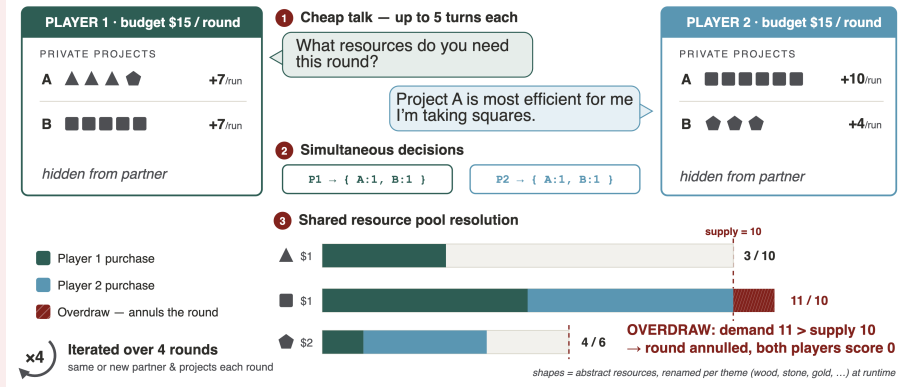


Figure 1: Game overview. Agents exchange cheap talk about private projects, submit simultaneous purchases from a shared pool, and repeat this process for four rounds; any overdraw annuls the round.

We argue that the gap between static and dynamic grounding is a key bottleneck for multi-agent LLM coordination. To study this, we introduce an *iterated, multi-turn* negotiation game in which two agents share a pool of resources and must allocate them toward private *projects* with different resource requirements and rewards (Figure 1). Within each round, agents engage in multi-turn cheap talk before making simultaneous purchase decisions; across rounds, they face the same or new partners and scenarios, enabling the study of grounding accumulation, repair, and adaptation over repeated encounters. The asymmetric project structure forces agents to explain not only *what* they want, but *why*: which projects they pursue, which resources are flexible, and how to avoid costly overdraw penalties. Because scenarios have verifiable joint optima, we can measure the gap between what agents *could* achieve and what they actually achieve, while using interaction traces to diagnose where grounding breaks down.

Our task differs from classical game-theoretic formulations in which agents act over known payoff structures. Here, payoff-relevant project constraints are private and combinatorial, so agents must use natural language to discover one another’s needs before acting. The annulment rule creates a social-dilemma incentive: over-demanding any shared resource gives both agents zero reward. Thus successful coordination requires not only information exchange, but also maintaining and honoring a joint plan through simultaneous, multi-resource allocations across repeated rounds.

Our experiments show that individual agents can identify optimal strategies in isolation, yet interacting dyads exhibit a substantial coordination gap when private objectives conflict. Performance rises as individual optima become compatible, but exact joint-optimal coordination remains unreliable in the competitive regime. We identify four recurring failure patterns: loss of shared interaction history, anchoring to early proposals, equal-split shortcuts, and agreement abandonment between public commitments and final allocations. To determine which components of the task drive this gap, we add targeted ablations that vary communication availability, partner context, information transparency, and incentive framing. These interventions show that cheap talk and shared information improve coordination but do not by themselves ensure joint-optimal execution; explicit joint-reward framing removes a substantial component of the gap, while residual failures remain in joint-plan formation and commitment maintenance. Our contributions are: (i) a configurable negotiation game environment with verifiable outcomes for studying interactive coordination; (ii) an intervention-based decomposition of how communication, information, interaction history, and incentive framing affect dyadic performance, together with preliminary human-calibrated evidence on four behavioral failure patterns; and (iii) a framework for evaluating structured interaction and commitment interventions.

Framework	Multi-turn	Iterated	Cheap talk	Private info	Eval. focus
Davidson et al. (2024)	✓	—	✓	✓	Outcome
Abdelnabi et al. (2024)	✓	—	✓	✓	Outcome
Bianchi et al. (2024)	✓	—	✓	✓	Outcome
Eisenstein et al. (2026)	✓	—	✓	✓	Process
Qian et al. (2026)	—	✓	—	✓	Outcome
Madmoun & Lahlou (2026)	—	✓	Minimal	—	Outcome
Ours	✓	✓	✓	✓	Both

Table 1: Comparison of multi-agent LLM evaluation frameworks. *Multi-turn*: multiple exchanges within a single decision round. *Iterated*: repeated decision rounds with the same or different partners. *Private info*: agents hold hidden reward-relevant information. *Eval focus*: whether analysis targets negotiation outcomes or the communicative process.

2 Related work

Our work sits at the intersection of three lines of research: grounding in dialogue, LLM-based negotiation, and multi-agent coordination. We review each and identify the gap our framework addresses.

2.1 Grounding in dialogue

Grounding, introduced by Clark & Brennan (1991), is the process of establishing mutual belief sufficient for current purposes. In cognitive science, it includes both static symbol grounding, mapping language to perceptual or external referents, and dynamic communicative grounding, where shared understanding is built interactively over mutually presupposed common ground (Stalnaker, 2002). NLP has largely emphasized the former while neglecting the latter (Chandu et al., 2021); a recent survey of 448 papers similarly finds that evaluation frameworks for dynamic grounding in Language Model (LM) interactions remain scarce (Anikina et al., 2025). We target this gap with a task that requires agents to communicate, negotiate shared plans, repair misunderstandings, and track commitments across turns to achieve optimal outcomes.

2.2 LLM-based negotiation

Recent work evaluates LLMs in structured negotiation settings, including landlord–tenant bargaining (Davidson et al., 2024), six-agent multi-issue negotiation (Abdelnabi et al., 2024), resource exchange and ultimatum games (Bianchi et al., 2024), equilibrium computation (Hua et al., 2024), Deal-or-No-Deal self-play (Liao et al., 2024), and multi-party bargaining with hidden rewards (Qian et al., 2026). Across these settings, two gaps remain. First, negotiations are often *multi-turn* but not *iterated*: agents exchange messages before one decision, with little chance to repair grounding across repeated encounters. Second, evaluation emphasizes *outcome metrics*, such as deal rates, scores, and Pareto efficiency, rather than the communicative *process*. Table 1 summarizes how our framework addresses both gaps by combining multi-turn cheap talk with simultaneous decisions across iterated rounds with stable or shifting partners, enabling analysis of grounding accumulation and repair.

2.3 Multi-agent coordination

MT-PingEval (Eisenstein et al., 2026) evaluates collaborative private-information games under fixed token budgets and variable turn counts, introducing a hierarchy of *interactivity levels*. It finds that LLMs often fail to benefit from additional turns, producing content-rich utterances that they “struggle to deploy strategically in service of collaborative goals.” The Collaboration Gap (Davidson et al., 2025) likewise documents degraded model performance in collaborative maze solving; unlike our repeated simultaneous-allocation game, it evaluates a single joint task with fully aligned objectives. We extend these paradigms to iterated negotiation, where agents must manage both private information and varying degrees of

incentive conflict. We use this hierarchy as an analytical lens while extending attention from informational interactivity to joint-plan grounding and execution under mixed incentives.

The hidden-profile paradigm, where optimal group decisions require pooling private information (Stasser & Titus, 1985), has been extended to LLM agents in HiddenBench (Li et al., 2026), which finds a gap between fully informed single agents and groups that must discuss asymmetric evidence. Our game adds compatible and conflicting private goals, and evaluates whether discussion produces a verifiable joint optimum. Ad-hoc teamwork studies previously unknown teammates (Mirsky et al., 2022), motivating our shifting-partner manipulation; sequential social dilemmas model temporally extended mixed incentives (Leibo et al., 2017), whereas our agents must discover private combinatorial constraints through language and execute a simultaneous multi-resource allocation. CaSiNo provides a complementary human negotiation corpus with dialogue acts and preferences (Chawla et al., 2021); our controlled setting instead measures the allocation that follows dialogue. Minimal channels can increase cooperation in Stag Hunt (Madmoun & Lahlou, 2026), but cannot support the referential grounding required here. Finally, collaborative coding teams also exhibit vague messages, commitment violations, and incorrect partner expectations (Khatua et al., 2026); our game isolates these coordination challenges in a setting with verifiable outcomes.

3 Game environment

As illustrated in Figure 1, two agents share a pool of resources (e.g., wood, stone, gold), each with a fixed per-round supply and unit cost. Each agent has an independent budget and simultaneously submits a purchase decision each round; if combined demand for any resource exceeds supply, the round is *annulled* and both agents receive zero reward. Rather than assigning scalar values to resources directly, each agent is assigned private *projects*: combinatorial goals requiring specific resource mixes and yielding rewards per completed run (e.g., 3 wood + 2 stone $\rightarrow$ 50 points). Unspent budget has no value. This project layer shifts the pragmatics from negotiating quantities to communicating *why* particular combinations matter, creating a natural need for referential grounding around project names, requirements, and strategies. Appendix A analyzes the task through the interactivity hierarchy of Eisenstein et al. (2026), distinguishing informational exchange from the confirmation and commitment demands introduced by mixed incentives.

Each game consists of multiple rounds (set to 4 for our study). Within each round, agents first engage in *cheap talk*: up to 5 turns each of alternating natural-language messages, during which they may share project information, propose allocations, or negotiate. Each agent also maintains a private *thinking* scratchpad hidden from the opponent, enabling comparison of private intent with public speech. Agents then independently submit a resource allocation together with project assignments as structured JSON. After both decisions, the round resolves: overdraw is checked, rewards are computed, and both agents observe the opponent’s purchases. Opponent rewards are not disclosed during play unless voluntarily shared; cumulative rewards are revealed only after the game for reflection.

3.1 Scenario generation

We parameterize goal conflict via the *compatibility ratio* M/C : the maximum achievable joint reward M divided by the sum of individual maxima $C = V_1 + V_2$ (each agent optimizing independently). When $M/C = 1.0$, optimal strategies are fully compatible; when $M/C < 1.0$, individual optima conflict. We generate scenarios at three levels: $M/C \in \{0.5, 0.8, 1.0\}$.

Scenarios are generated via *simulated annealing*. The generator initializes six random projects (three per agent), then iteratively perturbs requirements, rewards, and assignments to minimize a composite loss targeting the desired M/C ratio (± 0.05). The loss jointly enforces: equal individual maxima ($V_1 = V_2$); swap fairness (the joint optimum can favor either agent equally); multiple joint-optimal solutions; and individual affordability of all projects. The final selected scenarios are filtered to enforce equal individual maxima, swap fairness, and at least two distinct joint-optimal allocations. An oracle validation step further restricts the

candidate pool to those that all models studied in subsequent games can solve under full information with a single attempt (pass@1) individually. Thus, each evaluated model can identify a joint-optimal allocation for the selected scenarios in isolation when given full information, establishing individual task competence on the controlled scenario pool. The complete scenario pool appears in Appendix C.

4 Experimental setup

We evaluate a primary three-model cohort spanning multiple providers: Claude Sonnet 4.5 (Anthropic), GPT-5 Mini (OpenAI), and Qwen 3.5 Flash (via OpenRouter); see Appendix B for model configuration details. Prompt templates are provided in Appendix D. $N = 10$ games are played for each of $3 \times 2 \times 2 = 12$ conditions, with all-to-all play comprising three self-play and three cross-play pairings, yielding 720 game traces. Each 10-game cell contains five games under each first-speaker assignment. We additionally replicate the complete self-play design under the original individual-reward prompt for GPT-OSS-120B, Gemini 3 Flash, and Gemini 3.1 Pro, adding 360 game traces. These later extensions broaden model coverage but are not included in cross-play comparisons or LLM-judge annotations. Each game consists of four iterated rounds. The primary and additional individual-reward cohorts comprise 1,080 games and 4,320 rounds. Including the no-talk, full-transparency, and aligned-incentive follow-ups reported below, the paper analyzes 1,944 games and 7,776 rounds. All traces (cheap talk transcripts, thinking logs, allocations, rewards, and experiment metadata) are stored for reproducibility and released for further analysis; see Appendix H. Abstract resource identifiers are replaced at runtime with a list of pre-sampled thematic names to reduce sensitivity to particular resource vocabularies (Appendix C).

4.1 Conditions and controls

1. **Compatibility ratio** ($M/C \in \{0.5, 0.8, 1.0\}$): varies the degree of goal conflict. Agents start without knowledge of the other parties’ projects and must communicate to determine whether their goals are aligned or opposed.
2. **Partner stability**: *stable* (same partner) vs. *shifting* (one agent’s context resets each round). This evaluates whether agents can leverage shared history to form ad-hoc conventions (Hawkins et al., 2023), a well-studied phenomenon in humans yet often overlooked in LLMs.
3. **Project rotation**: *fixed* vs. *rotating* (new scenarios each round). This holds the compatibility ratio constant, preserving the level of goal conflict, but eliminates a fixed environment and tests whether agents can develop transferable coordination strategies rather than memorizing prior successful allocations.

4.2 Metrics

We organize metrics into *outcome metrics* quantifying coordination quality and *process metrics* analyzing communicative strategies (Table 2).

Type	Metric	Definition
Outcome	Overdraw rate	Fraction of rounds where demand exceeds supply
Outcome	Allocation efficiency	Achieved joint reward / oracle optimum
Outcome	Optimum rate	Fraction of rounds achieving joint optimal strategy
Process	First-proposal deference	Degree to which responder’s allocation matches first proposal
Process	Allocation anchoring	Consecutive-round allocation similarity (exact match rate)
Process	Stated-vs-actual coherence	Commitment to previously grounded plans

Table 2: Outcome metrics quantify the coordination gap; process metrics decompose its behavioral sources.

5 Results

Table 3 summarizes coordination quality across the primary all-to-all cohort. We anchor the results against an oracle baseline, a no-talk construct-validity check, and a full-transparency intervention. We then examine interaction history and turn order, behavioral indicators, additional model coverage, incentive framing, and information asymmetry as complementary explanations for the remaining coordination gap.

Pair	Overdraw ↓			Efficiency ↑			Optimum ↑		
	0.5	0.8	1.0	0.5	0.8	1.0	0.5	0.8	1.0
<i>Primary self-play</i>									
GPT-5 Mini	23.1	9.4	3.8	68.2	85.6	94.0	40.6	66.3	86.9
Sonnet 4.5	30.0	14.4	6.3	63.7	79.4	91.1	27.5	53.1	79.4
Qwen 3.5	38.8	32.5	16.3	56.8	61.2	76.1	32.5	43.8	43.8
<i>Primary cross-play</i>									
Sonnet 4.5 × GPT-5 Mini	10.6	5.0	1.3	79.8	89.7	97.9	34.4	66.3	91.3
GPT-5 Mini × Qwen 3.5	16.9	13.8	6.3	74.0	77.4	87.4	31.9	57.5	69.4
Qwen 3.5 × Sonnet 4.5	29.4	16.9	7.5	63.8	74.1	85.1	21.9	43.1	52.5

Table 3: Performance across compatibility ratios in the primary all-to-all cohort. Overdraw (%), Efficiency (%), and Optimum (%) rates are shown. Table labels abbreviate Claude Sonnet 4.5 and Qwen 3.5 Flash as Sonnet 4.5 and Qwen 3.5.

Across the primary all-to-all cohort, 15.7% of rounds result in overdraw, concentrated in competitive scenarios ($M/C=0.5$: 24.8%) and rare in compatible ones ($M/C=1.0$: 6.9%). Despite the competitive gap, most primary-cohort dyads eventually find effective coordination: 81.7% of games contain at least one jointly optimal round, and among stable games that eventually reach an optimum, the first optimum is reached after an average of 1.6 rounds. After an overdraw, 83.3% of subsequent rounds yield positive scores, showing that most dyads recover to a positive score following an overdraw. Unless otherwise noted, bootstrap intervals quantify game-level sampling variation conditional on the selected scenario pool.

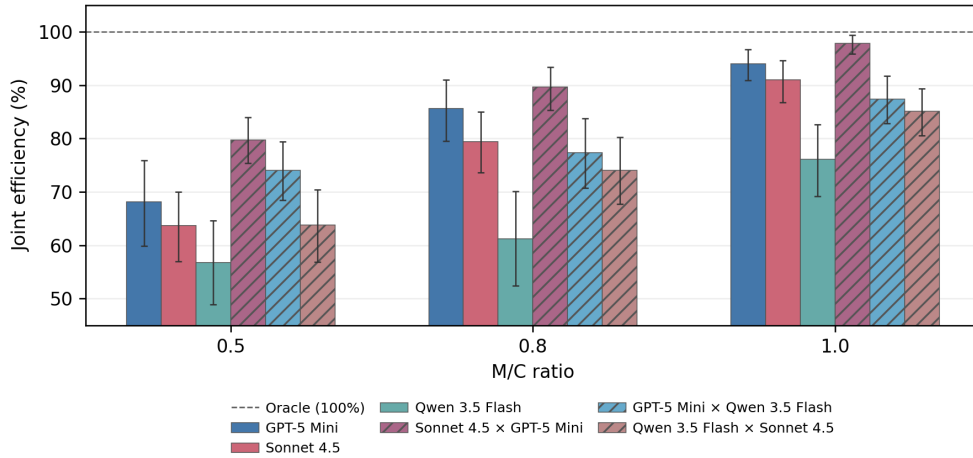


Figure 2: Joint efficiency for self-play (solid) and cross-play (hatched) dyads across compatibility ratios. Bars show game-level means with 95% game-cluster bootstrap confidence intervals.

At $M/C=0.5$, cross-play point estimates are at least as high as the stronger model’s self-play baseline (Figure 2). Claude Sonnet 4.5 × GPT-5 Mini achieves 79.8% joint efficiency (vs. 68.2% and 63.7% self-play), and GPT-5 Mini × Qwen 3.5 Flash reaches 74.0% (vs. 68.2%). Qwen 3.5 Flash × Claude Sonnet 4.5 closely matches Claude Sonnet 4.5 self-play (63.8%

vs. 63.7%). These point-estimate patterns are consistent with behavioral diversity aiding coordination when goal conflict is high, with the lowest overdraw rates in the Claude Sonnet 4.5 $\times$ GPT-5 Mini pairing (1.3% in compatible scenarios, 10.6% in competitive ones). As compatibility increases ($M/C=0.8$ and 1.0), the cross-play advantage narrows and weaker models begin to drag down stronger partners. GPT-5 Mini $\times$ Qwen 3.5 Flash falls below GPT-5 Mini self-play at $M/C=0.8$ (77.4% vs. 85.6%), consistent with the possibility that under reduced conflict, individual capability matters more than behavioral diversity.

5.1 The value of cheap talk

We use a no-talk self-play control to verify the task premise that communication enables coordination. Removing cheap talk reduces performance: under competitive goals, mean joint efficiency falls from 63.2% to 18.0%, and only 21.8% of no-talk rounds avoid overdraw (Figure 3). This check motivates the remaining analyses, which explore why communication alone does not yield reliable joint-optimal execution. Appendix F reports the full condition-level comparison.

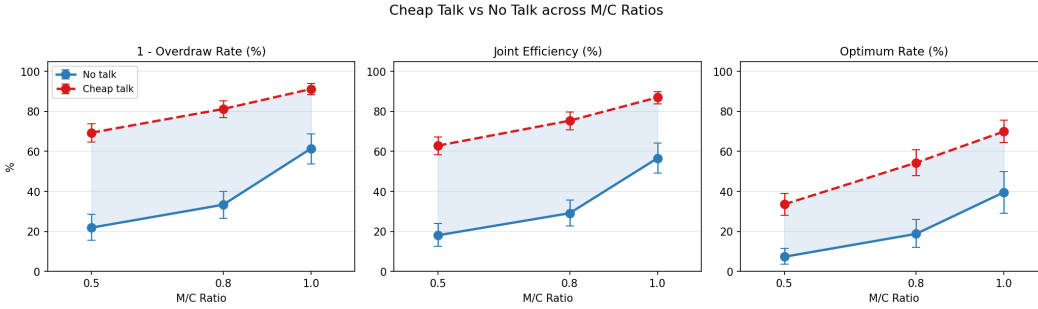


Figure 3: Cheap-talk control across compatibility ratios, aggregated over the primary self-play models and conditions. Higher values are better in every panel: the left panel reports the fraction of rounds without overdraw, the middle panel joint efficiency, and the right panel optimum rate. Error bars show 95% game-cluster bootstrap confidence intervals.

5.2 Interaction history: accumulation and reset

To characterize later-round improvement within stable games, we compare their first rounds, when neither agent has shared interaction history, with their later rounds. Figure 4 reports game-cluster bootstrap estimates. Later stable rounds have higher non-overdraw rates and joint efficiency at every compatibility level. Exact-optimum rates do not reliably change. This later-round pattern is consistent with agents benefiting from continued interaction with a stable partner, while also reflecting outcome feedback, task familiarity, and, in fixed-project games, repeated exposure to the same optimization problem.

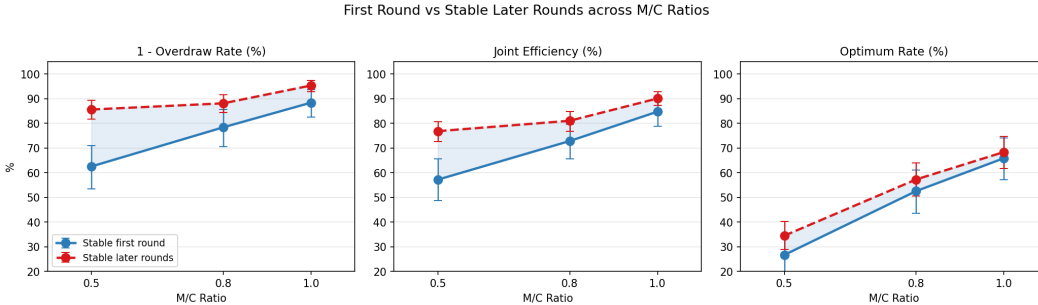


Figure 4: First-round versus later-round outcomes within stable-partner games. First rounds have no prior shared interaction history. Error bars show 95% game-cluster bootstrap confidence intervals.

We separately estimate the overall association of assignment to the asymmetric shifting condition, in which Agent B’s previous-round context is reset after round one. In a pooled round-level mixed-effects analysis, shifting is associated with 3.5 percentage points lower joint efficiency (95% CI $[-6.3, -0.7]$, $p = 0.015$) and 41% higher odds of overdraw (odds ratio 1.41, 95% CI $[1.12, 1.77]$, $p = 0.0036$). Exact-optimum attainment does not differ significantly between conditions (odds ratio 1.01, 95% CI $[0.78, 1.30]$, $p = 0.952$). The model specification and fixed-effect estimates are reported in Appendix E.2.

5.3 Turn order

The role-swapped cross-play design tests whether the model that speaks first has an outcome advantage. For each model pair, we estimate the difference in mean game-level joint efficiency using 10,000 independent whole-game bootstrap resamples within starter groups, and report two-sided 95% percentile intervals. All three intervals include zero: Sonnet-first versus GPT-first is +1.7 percentage points (95% CI $[-3.1, +6.7]$), GPT-first versus Qwen-first is +3.9 $[-2.9, +10.5]$, and Sonnet-first versus Qwen-first is $-3.2 [-10.7, +4.2]$. We detect no clear first-speaker advantage, and balanced role assignment prevents starter identity from systematically favoring a model pair.

5.4 Behavioral failure patterns

The descriptive analyses point to several complementary shortcuts. Agents make equal requests for a shared resource in 12.5% of rounds; nearly half of these equal splits are suboptimal, yielding a 6.1% perfunctory-fairness rate. In repeated same-scenario games, 29.2% of suboptimal, non-overdraw allocations are repeated unchanged, consistent with stubborn anchoring. Regex-extracted other-directed proposals appear in 12.9% of rounds. In first-round non-rotating games, the recipient matches every quantity specified for them in 20.4% of such proposals (Appendix E). These indicators document observable behavioral failure patterns. Future studies should identify the internal mechanisms that generate these patterns; the LLM-judge analysis appears in Appendix G.

LLM-assisted trace analysis. To assess stated-vs.-actual coherence, we applied the LLM-judge pipeline to all 720 primary-cohort game traces. Its agreement-abandonment label identifies rounds in which a concrete or strongly suggestive public plan is not maintained in the final allocation. On 40 human-annotated rounds, agreement abandonment attains precision 0.67, recall 0.75, and $\kappa = 0.63$. It is the strongest outcome-linked judge pattern, appearing in 41.7% of overdrawn rounds compared with 7.4% of optimal rounds. This pattern is consistent with a referential-binding failure: a public commitment is not preserved as an action-relevant constraint at decision time. Appendix G provides the directional enrichment plots, full label definitions, calibration, and disagreement examples; an agreement-abandonment trace appears in Appendix I.

5.5 Additional model coverage and aligned incentives

Additional model coverage. We replicated the self-play design under the original individual-reward prompt for GPT-OSS-120B, Gemini 3 Flash, and Gemini 3.1 Pro. These models show pronounced heterogeneity, but all retain a gap in the competitive condition: at $M/C=0.5$, GPT-OSS-120B and Gemini 3 Flash attain mean joint efficiency of 86.2% and 88.1% while reaching exact optima in only 55.0% and 37.5% of rounds; Gemini 3.1 Pro reaches 46.2% exact-optimum attainment. At $M/C=1.0$, all three improve, including 100.0% exact-optimum attainment for Gemini 3.1 Pro. These extensions broaden the evidence that, under the original individual-reward objective, exact joint-optimum attainment remains difficult at $M/C=0.5$ even when mean efficiency is high. At $M/C=1.0$, compatible private objectives lead to stronger performance, providing a check that agents can coordinate when they need not negotiate a tradeoff. The aligned-incentive comparison below tests whether bargaining conflict alone accounts for the remaining gap. Full extension outcomes are in Table 10 in Appendix F.

Aligned incentives. The original task asks each agent to maximize its own reward, so suboptimal outcomes can reflect bargaining rather than a failure to form a joint plan. We therefore ran a matched self-play intervention in which both agents are instructed to maximize total reward. Table 4 compares the original and aligned prompts. In competitive settings, the joint-reward prompt raises efficiency by 33.0 points for Claude Sonnet 4.5 (95% CI [+26.7, +39.8]) and 10.3 points for Gemini 3 Flash (95% CI [+6.5, +14.5]); the corresponding exact-optimum gains are 30.6 and 55.6 points. GPT-5 Mini and GPT-OSS-120B show mixed effects. Gemini 3.1 Pro provides a boundary case: under the original prompt at $M/C=0.5$, it achieves 96.3% mean joint efficiency and avoids overdraw but reaches the exact optimum in only 46.2% of rounds; under aligned incentives, it reaches the optimum in every observed condition. For this model and controlled scenario set, explicit joint incentives make the task one the dyad can solve as collaborative planning, whereas separate private objectives can leave joint value unrealized even when mean efficiency is high. Appendix F reports game-cluster bootstrap intervals for every prompt contrast. The intervention provides experimental evidence that incentive framing affects performance, with large competitive-condition gains for several tested models. It identifies the performance cost of bargaining conflict and motivates coordination mechanisms that preserve separate user objectives.

Model	$M/C=0.5$	$M/C=0.8$	$M/C=1.0$
GPT-5 Mini	68.2/40.6 → 80.4/53.8	85.6/66.3 → 81.2/50.6	94.0/86.9 → 88.9/56.2
GPT-OSS-120B	86.2/55.0 → 85.6/49.4	88.6/72.5 → 89.3/67.5	93.7/78.1 → 92.4/75.6
Sonnet 4.5	63.7/27.5 → 96.7/58.1	79.4/53.1 → 97.1/88.8	91.1/79.4 → 94.9/71.2
Gemini 3 Flash	88.1/37.5 → 98.4/93.1	98.4/88.8 → 100.0/100.0	99.0/95.0 → 99.9/98.1
Gemini 3.1 Pro	96.3/46.2 → 100.0/100.0	99.2/93.8 → 100.0/100.0	100.0/100.0 → 100.0/100.0

Table 4: Follow-up self-play under original versus maximize-joint-reward prompts. Entries are joint efficiency / optimum rate (%).

Of the 478 eligible suboptimal, non-overdraw aligned-prompt rounds, only 36.6% leave one agent with more than half the oracle joint optimum while the other receives less. Thus, in 63.4% of these residual outcomes, neither agent obtains more than an equal joint-optimal share. These outcomes cannot generally be attributed to successful surplus extraction and are consistent with persistent difficulty forming or executing a joint plan after the prompt specifies a shared objective. Full model-level results are reported in Appendix E.3.

5.6 Full-transparency intervention

To isolate how much of the coordination gap stems from information asymmetry, we ran a full-transparency intervention in which both agents’ complete project specifications are revealed in the system prompt before cheap talk begins, eliminating private-project information asymmetry. We selected the Qwen 3.5 Flash × GPT-5 Mini pair, a lower-performing cross-play pair at $M/C=0.5$ (74.0% efficiency, Table 3), as a demanding test of the role of private-project information. We ran $N = 120$ games across all three compatibility ratios, both partner-stability conditions, and both project-rotation conditions.

Full transparency reduced overdraw overall and did not increase it in any displayed condition (12.3% → 8.1%, $\Delta = -4.2$ pp, 95% CI [-7.7, -0.6], $p = 0.021$) and raised joint efficiency (79.6% → 85.3%, $\Delta = +5.6$ pp, 95% CI [+1.8, +9.4], $p = 0.004$), with the largest improvements in the stable competitive condition ($M/C=0.5$: overdraw 11.2% → 3.8%, efficiency 78.1% → 89.0%). The exact-optimum contrast is -2.9 points (95% CI [-10.2, +4.2], $p = 0.428$); its point estimate does not improve overall and declines at $M/C=1.0$ (69.4% → 64.4%), where project requirements are already non-conflicting.

Revealing project structure therefore reduces information asymmetry and overdraw, but does not reliably improve exact-optimum attainment. Full per-condition outcome metrics are in Table 11 (Appendix F).

6 Discussion

From information exchange to grounded execution. Taken together, our experiments reveal a layered coordination gap. The evaluated agents can identify optimal allocations for the selected scenarios in isolation, and the no-talk control verifies that the environment makes communication consequential. Yet even when agents exchange private constraints, and when full transparency supplies their partner’s project information directly, dyads do not reliably reach the exact joint optimum under separate objectives. The remaining challenge therefore concerns whether agents transform available information into a mutually grounded plan, preserve that plan through the interaction, and execute compatible actions simultaneously.

This cautions against treating multi-agent competence as compositional: combining individually capable agents does not necessarily produce a capable collective. The interaction-history results suggest that coordination can improve through partner-specific adaptation, while agreement abandonment shows that apparent linguistic convergence need not imply behavioral coordination. Aligned incentives further show that some failures are sensitive to objective framing, but this intervention recasts delegated bargaining as cooperative planning. The harder problem is reliable coordination while agents continue to represent distinct users with potentially conflicting interests.

Research agenda for delegated multi-agent coordination. Candidate mechanisms include persistent partner-state summaries, explicit acknowledgements, structured plans, commitment ledgers, and pre-execution checks. Future studies should test which support common-ground formation, prevent plan drift, and close the gap between stated commitments and final actions.

Second, evaluations must distinguish grounding failure from rational disagreement or strategic defection. Departures may reflect misunderstanding, forgotten context, revised beliefs, or private-reward pursuit. Addressing this question requires evaluation designs that compare otherwise matched decisions under controlled changes to incentives or available information, to identify how these task components shape coordination outcomes. A third question is how agents can coordinate without collapsing distinct objectives. This calls for incentive-compatible protocols, credible commitments, privacy-preserving disclosure, fairness constraints, and procedures for negotiating Pareto-efficient outcomes without silently sacrificing one principal’s interests. As such delegates are deployed, safeguards should make inter-agent commitments inspectable and constrain disclosure of private user goals, since stronger coordination can also amplify manipulative or harmful collective behavior.

Scope and limitations. Our conclusions characterize a controlled set of dyadic allocation scenarios rather than bargaining in general. The main all-to-all comparison contains three models, with additional models evaluated only in self-play; additional flagship models and human-human or human-agent dyads remain untested. The environment assumes symmetric roles, a fixed communication window, deterministic resources, and verifiable joint optima. These restrictions provide experimental control, but future work must test whether the distinction between information exchange, plan grounding, and execution persists in open-ended and strategically richer environments. Finally, larger human validation sets and broader scenario sampling would strengthen the process measures.

7 Conclusion

Talk alone does not produce grounded coordination. Our controlled paradigm exposes the gap between exchanging task-relevant information, establishing an actionable shared plan, and carrying that plan into compatible behavior. The intervention results provide a foundation for studying these stages separately and testing mechanisms that repair breakdowns between them. As autonomous agents increasingly negotiate on behalf of different users, the central challenge is not merely to make agents talk, but to make their talk establish common ground that preserves distinct interests and governs action.

Acknowledgments

We thank Chelsea Zou for her contributions to setting up the LLM-judge pipeline and annotating game traces. Yiheng Yao was supported by the SG Digital Scholarship of the Infocomm Media Development Authority (IMDA) of Singapore.

References

- Sahar Abdelnabi, Amr Gomaa, Sarath Sivaprasad, Lea Schönherr, and Mario Fritz. Cooperation, competition, and maliciousness: LLM-stakeholders interactive negotiation. In *Advances in Neural Information Processing Systems*, volume 37. Curran Associates, Inc., 2024. doi: 10.52202/079017-2658.
- Tatiana Anikina, Alina Leippert, and Simon Ostermann. Building common ground in dialogue: A survey. In *Proceedings of the 2nd LUHME Workshop*, pp. 3–28. UP - Universidade do Porto, LIACC, CLUP, UEF, and UAH, 2025. URL <https://aclanthology.org/2025.luhme-1.2/>.
- Federico Bianchi, Patrick John Chia, Mert Yuksekgonul, Jacopo Tagliabue, Dan Jurafsky, and James Zou. How well can LLMs negotiate? NegotiationArena platform and analysis. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 3935–3951. PMLR, 2024. URL <https://proceedings.mlr.press/v235/bianchi24a.html>.
- Khyathi Raghavi Chandu, Yonatan Bisk, and Alan W Black. Grounding ‘grounding’ in NLP. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pp. 4283–4305. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.findings-acl.375. URL <https://aclanthology.org/2021.findings-acl.375/>.
- Kushal Chawla, Jaysa Ramirez, Rene Clever, Gale Lucas, Jonathan May, and Jonathan Gratch. CaSiNo: A corpus of campsite negotiation dialogues for automatic negotiation systems. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 3167–3185. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.naacl-main.254. URL <https://aclanthology.org/2021.naacl-main.254/>.
- Herbert H. Clark and Susan E. Brennan. Grounding in communication. In *Perspectives on Socially Shared Cognition*, pp. 127–149. American Psychological Association, 1991. doi: 10.1037/10096-006.
- Tim R. Davidson, Veniamin Veselovsky, Michal Kosinski, and Robert West. Evaluating language model agency through negotiations. In *International Conference on Learning Representations*, 2024. URL <https://openreview.net/forum?id=3ZqKxMHcAg>.
- Tim R. Davidson, Adam Fourney, Saleema Amershi, Robert West, Eric Horvitz, and Ece Kamar. The collaboration gap. *arXiv preprint arXiv:2511.02687*, 2025.
- Jacob Eisenstein, Fantine Huot, Adam Fisch, Jonathan Berant, and Mirella Lapata. MT-PingEval: Evaluating multi-turn collaboration with private information games. *arXiv preprint arXiv:2602.24188*, 2026.
- R. D. Hawkins, M. Franke, M. C. Frank, A. E. Goldberg, K. Smith, T. L. Griffiths, and N. D. Goodman. From partners to populations: A hierarchical bayesian account of coordination and convention. *Psychological Review*, 130(4):977–1016, 2023. doi: 10.1037/rev0000348.
- Wenyue Hua, Ollie Liu, Lingyao Li, Alfonso Amayuelas, Julie Chen, Lucas Jiang, Mingyu Jin, Lizhou Fan, Fei Sun, William Wang, Xintong Wang, and Yongfeng Zhang. Game-theoretic LLM: Agent workflow for negotiation games. *arXiv preprint arXiv:2411.05990*, 2024.

- Arpandeeep Khatua, Hao Zhu, Peter Tran, Arya Prabhudesai, Frederic Sadrieh, Johann K. Lieberwirth, Xinkai Yu, Yicheng Fu, Michael J. Ryan, Jiaxin Pei, and Diyi Yang. Cooper-bench: Why coding agents cannot be your teammates yet. *arXiv preprint arXiv:2601.13295*, 2026.
- Joel Z. Leibo, Vinicius Zambaldi, Marc Lanctot, Janusz Marecki, and Thore Graepel. Multi-agent reinforcement learning in sequential social dilemmas. In *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems*, pp. 464–473. International Foundation for Autonomous Agents and Multiagent Systems, 2017.
- Yuxuan Li, Aoi Naito, and Hirokazu Shirado. Systematic failures in collective reasoning under distributed information in multi-agent LLMs. *arXiv preprint arXiv:2505.11556*, 2026.
- Shirong Liao, Jack Liu, and Osbert Bastani. Efficacy of language model self-play in non-zero-sum games. *arXiv preprint arXiv:2406.18872*, 2024. URL <https://arxiv.org/abs/2406.18872>.
- Hachem Madmoun and Salem Lahlou. Communication enables cooperation in LLM agents: A comparison with curriculum-based approaches. In Vera Demberg, Kentaro Inui, and Lluís Marquez (eds.), *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 307–321, Rabat, Morocco, 2026. Association for Computational Linguistics. doi: 10.18653/v1/2026.eacl-short.23. URL <https://aclanthology.org/2026.eacl-short.23/>.
- Reuth Mirsky, Ignacio Carlucho, Arrasy Rahman, Elliot Fosong, William Macke, Mohan Sridharan, Peter Stone, and Stefano V. Albrecht. A survey of Ad Hoc teamwork research. In Dorothea Baumeister and Jörg Rothe (eds.), *Multi-Agent Systems*, volume 13442 of *Lecture Notes in Computer Science*, pp. 275–293, Cham, 2022. Springer International Publishing. doi: 10.1007/978-3-031-20614-6_16. URL https://link.springer.com/chapter/10.1007/978-3-031-20614-6_16.
- Crystal Qian, Kehang Zhu, John Joseph Horton, Benjamin S. Manning, Vivian Tsai, James Wexler, and Nithum Thain. Strategic tradeoffs between humans and AI in multi-agent bargaining. In *Proceedings of the 31st International Conference on Intelligent User Interfaces, IUI '26*, pp. 1625–1646, New York, NY, USA, 2026. Association for Computing Machinery. doi: 10.1145/3742413.3789078. URL <https://dl.acm.org/doi/10.1145/3742413.3789078>.
- Robert Stalnaker. Common ground. *Linguistics and Philosophy*, 25(5):701–721, 2002. doi: 10.1023/A:1020867916902.
- Garold Stasser and William Titus. Pooling of unshared information in group decision making: Biased information sampling during discussion. *Journal of Personality and Social Psychology*, 48(6):1467–1478, 1985. doi: 10.1037/0022-3514.48.6.1467.

A Informational interactivity and joint-plan grounding

Eisenstein et al. (2026) propose a hierarchy for collaborative private-information games in which level k characterizes the number of messages exchanged before a player can produce a correct answer. Their framework assumes that participants share an objective and comply with a common solution protocol. Our game differs in two respects: both agents must submit actions, and the individual-reward prompt instructs each agent to maximize its own reward. We therefore use the hierarchy to distinguish informational sufficiency from the additional grounding and commitment demands created by mixed incentives, rather than assigning the task an unconditional interactivity level.

Let X_1 and X_2 denote the agents’ private project information. Under a shared deterministic policy, two messages can be informationally sufficient. First, Agent 2 communicates its project information, $f_2(X_2)$. Second, Agent 1 combines this message with X_1 , computes a complete joint allocation (a_1^*, a_2^*) , and communicates that plan to Agent 2. The message may also apply a predetermined tie-breaking rule when multiple joint optima exist. If both agents are assumed to follow the shared policy, they can then submit the prescribed actions simultaneously.

The mixed-incentive cheap-talk setting adds a separate grounding problem. Agent 1’s proposal is non-binding, and after the second message Agent 1 does not know whether Agent 2 accepts the proposed division. A third message allows Agent 2 to confirm or counter the plan, creating public evidence that the proposal has been understood and accepted. This disclosure–proposal–confirmation sequence is therefore a natural grounding protocol, although the third message is not an information-theoretic requirement under the original shared-objective hierarchy.

Explicit confirmation still does not guarantee execution: either agent may subsequently submit an allocation inconsistent with the public plan. Our task therefore separates three stages that are conflated by outcome-only evaluation: pooling private information, grounding a joint plan, and carrying that plan into simultaneous action. The compatibility ratio primarily changes the bargaining difficulty during joint-plan grounding: when $M/C < 1$, agents must negotiate which private objective yields even after the relevant project information has been exchanged.

B LLM model configuration details

Table 5 identifies every model appearing in the reported main cohort or follow-up analyses. All models use streaming responses. A 1-second cooldown is enforced between consecutive API calls, with up to 10 retries using exponential backoff (base 2s, max 120s) for rate-limit errors. Malformed JSON responses are retried up to 3 times per decision. Per-run decoding parameters are recorded in the released traces. Where table width requires it, Sonnet 4.5 and Qwen 3.5 abbreviate Claude Sonnet 4.5 and Qwen 3.5 Flash.

Agent errors Rounds containing a `heuristic_fallback`, the definitive signal that an agent did not produce a valid allocation after all retries, are excluded from analysis. All three main-cohort models recovered within the retry budget on every round. Their per-round retrievable validation-error rates are 0.073 for Qwen 3.5 Flash, 0.071 for Claude Sonnet 4.5, and 0.047 for GPT-5 Mini.

Model	Provider	Model identifier
Claude Sonnet 4.5	Anthropic	claude-sonnet-4-5-20250929
GPT-5 Mini	OpenAI	gpt-5-mini-2025-08-07
Qwen 3.5 Flash	OpenRouter	qwen/qwen3.5-flash-02-23
GPT-OSS-120B	Vertex AI MaaS	openai/gpt-oss-120b-maas
Gemini 3 Flash	Vertex AI	publishers/google/models/gemini-3-flash-preview
Gemini 3.1 Pro	Vertex AI	publishers/google/models/gemini-3.1-pro-preview

Table 5: Models appearing in reported experiments. The first three models constitute the main cohort; the remaining models appear in follow-up self-play analyses.

GPT-5 Mini uses OpenAI’s `max_completion_tokens` parameter and does not support an explicit temperature setting. The remaining models use provider-standard generation parameters recorded with each run. Anthropic models support prompt caching; reasoning content (chain-of-thought) is extracted and stored separately from speech for all models that produce it.

Token usage. Table 6 reports API-reported token counts per agent-round for the three main-cohort models, averaged across all rounds in the 720-game cohort. Completion tokens are further broken down into *visible output* (speech + action JSON) and *thinking* (internal chain-of-thought, hidden from the opponent). Claude Sonnet 4.5 has lower API-reported prompt-token counts; Anthropic prompt caching may contribute to this difference.

Model	Prompt	Completion	Visible output	Think %
Sonnet 4.5	1,791	2,010	585	71%
GPT-5 Mini	5,977	3,190	647	80%
Qwen 3.5 Flash	5,938	2,806	629	78%

Table 6: Average API-reported token usage per agent-round (main cohort, 720 games). Speech and action tokens are parsed from the same LLM call as the thinking trace; `api_meta` is recorded on the thinking entry. Visible output = completion – thinking. Think % = thinking / completion. Counts use provider-specific tokenization and accounting, so they are descriptive rather than directly comparable computational-cost measures.

C Generated scenarios

C.1 Scenario Themes

To reduce sensitivity to particular resource names and associated lexical biases, we apply cosmetic thematic randomization. A single theme is sampled at game initialization and retained throughout the game, mapping the abstract resources r_1, r_2, r_3 to thematic names. This manipulation does not establish generalization beyond the selected scenario pool, but reduces dependence on any single resource vocabulary. Table 7 lists the available themes and their corresponding resource names.

Theme	r1 (Cost 1.0)	r2 (Cost 1.5)	r3 (Cost 3.0)
Medieval	wood	stone	gold
Space	titanium	crystal	plasma
Undersea	coral	pearl	trident
Steampunk	copper	brass	aether
Jungle	bamboo	vine	amber
Desert	sandstone	glass	ruby
Arctic	ice	fur	diamond
Volcanic	obsidian	basalt	magma
Cyberpunk	silicon	fiber	quantum
Fairy_tale	pixie_dust	moonstone	starlight

Table 7: Thematic resource mappings.

C.2 Scenario Pool Details

All scenarios in the pool share a common resource environment: Supply (r1: 10, r2: 10, r3: 6), Costs (r1: 1.0, r2: 1.5, r3: 3.0), and individual Agent Budget (18.0).

C.2.1 Compatibility Ratio $M/C \approx 0.5$ Scenario gen.012 ($M/C = 0.54$)

Agent 1 Projects			Agent 2 Projects		
Name	Requirements	Reward	Name	Requirements	Reward
project.a	r2x3	9	project.a	r3x3	1
project.b	r3x2, r2x3	4	project.b	r1x1, r2x3	9
project.c	r3x2, r1x1	1	project.c	r2x2	4

Scenario gen.053 ($M/C = 0.50$)

Agent 1 Projects			Agent 2 Projects		
Name	Requirements	Reward	Name	Requirements	Reward
project.a	r3x3, r2x5	3	project.a	r1x3, r3x4	2
project.b	r1x4, r3x3	3	project.b	r1x2, r2x5	3
project.c	r1x5	6	project.c	r1x3, r3x2	6

Scenario gen.062 ($M/C = 0.53$)

Agent 1 Projects			Agent 2 Projects		
Name	Requirements	Reward	Name	Requirements	Reward
project.a	r2x6	3	project.a	r3x1, r1x4	7
project.b	r3x5	1	project.b	r2x3, r1x6	4
project.c	r2x1, r1x5	9	project.c	r2x5, r1x1	9

Scenario gen.104 ($M/C = 0.53$)

Agent 1 Projects			Agent 2 Projects		
Name	Requirements	Reward	Name	Requirements	Reward
project.a	r1x2	4	project.a	r2x4, r3x3	1
project.b	r3x3, r1x3	7	project.b	r1x1	2
project.c	r1x6, r2x6	2	project.c	r3x4, r1x2	5

C.2.2 Compatibility Ratio $M/C \approx 0.8$ Scenario gen.001 ($M/C = 0.83$)

Agent 1 Projects			Agent 2 Projects		
Name	Requirements	Reward	Name	Requirements	Reward
project.a	r3x2, r2x6	8	project.a	r2x2, r3x2	5
project.b	r3x5, r1x2	5	project.b	r3x4	5
project.c	r3x1, r2x2	5	project.c	r1x6	10

Scenario gen.006 ($M/C = 0.80$)

Agent 1 Projects			Agent 2 Projects		
Name	Requirements	Reward	Name	Requirements	Reward
project.a	r3x1, r2x2	6	project.a	r2x2, r1x1	6
project.b	r3x5, r1x3	5	project.b	r1x3	6
project.c	r2x2	6	project.c	r3x5, r1x1	1

Scenario gen.017 ($M/C = 0.83$)

Agent 1 Projects			Agent 2 Projects		
Name	Requirements	Reward	Name	Requirements	Reward
project.a	r2x1, r3x3	2	project.a	r2x3	3
project.b	r1x1, r2x3	3	project.b	r2x1, r3x4	2
project.c	r3x5	6	project.c	r3x3, r1x4	6

Scenario gen.021 ($M/C = 0.83$)

Agent 1 Projects			Agent 2 Projects		
Name	Requirements	Reward	Name	Requirements	Reward
project.a	r1x2	1	project.a	r3x3, r1x3	6
project.b	r2x1, r1x3	6	project.b	r2x3	6
project.c	r3x3	6	project.c	r1x2, r2x4	5

Scenario gen.022 ($M/C = 0.83$)

Agent 1 Projects			Agent 2 Projects		
Name	Requirements	Reward	Name	Requirements	Reward
project.a	r1x6	8	project.a	r3x2, r2x6	8
project.b	r3x4	4	project.b	r1x2, r3x2	1
project.c	r2x2, r3x1	4	project.c	r3x3	6

C.2.3 Compatibility Ratio $M/C \approx 1.0$

Scenario gen.000 ($M/C = 1.00$)

Agent 1 Projects			Agent 2 Projects		
Name	Requirements	Reward	Name	Requirements	Reward
project.a	r3x3, r1x1	5	project.a	r2x4, r1x4	6
project.b	r3x1, r1x5	1	project.b	r2x4, r3x1	3
project.c	r2x5, r3x1	1	project.c	r1x3, r3x3	6

Scenario gen.001 ($M/C = 1.00$)

Agent 1 Projects			Agent 2 Projects		
Name	Requirements	Reward	Name	Requirements	Reward
project.a	r2x5, r3x3	7	project.a	r2x2	4
project.b	r1x1, r3x2	10	project.b	r2x4	10
project.c	r3x1	4	project.c	r3x1, r1x5	8

Scenario gen.002 ($M/C = 1.00$)

Agent 1 Projects			Agent 2 Projects		
Name	Requirements	Reward	Name	Requirements	Reward
project.a	r3x3, r2x4	1	project.a	r3x3, r2x3	10
project.b	r3x1, r1x6	10	project.b	r3x2, r1x3	5
project.c	r1x1, r3x3	10	project.c	r3x6	8

Scenario gen.006 ($M/C = 1.00$)

Agent 1 Projects			Agent 2 Projects		
Name	Requirements	Reward	Name	Requirements	Reward
project.a	r3x2	4	project.a	r1x2	4
project.b	r2x1	2	project.b	r2x2, r3x1	2
project.c	r1x6, r2x3	9	project.c	r3x1, r2x3	10

Scenario gen.010 ($M/C = 1.00$)

Agent 1 Projects			Agent 2 Projects		
Name	Requirements	Reward	Name	Requirements	Reward
project.a	r1x5, r2x4	8	project.a	r2x4	4
project.b	r3x4, r2x2	6	project.b	r3x6	8
project.c	r3x3, r2x2	8	project.c	r1x2, r2x3	2

Scenario gen.014 ($M/C = 1.00$)

Agent 1 Projects			Agent 2 Projects		
Name	Requirements	Reward	Name	Requirements	Reward
project.a	r3x3, r2x1	5	project.a	r3x1, r2x2	10
project.b	r1x3	10	project.b	r1x3, r3x1	10
project.c	r3x3, r1x1	10	project.c	r3x4	6

D Detailed game prompts

This appendix documents the prompt components used by the reported conditions in our negotiation game. All prompts are defined in `backend/agents/prompts.py` and dynamically instantiated from game configuration. We present the project-based reward system (schema version 5+), used by the experiments reported in the main text.

D.1 System prompt structure

Each agent receives a system-level prompt at the start of the game that remains constant throughout all rounds. The prompt consists of four main sections: game rules, project specifications, response format, and strategic context.

D.1.1 Game rules

```
## Game Rules
- Resources available: {resource_types}
- Resource supply per round: {resource_supply}
- Resource costs: {resource_costs}
- Your budget per round: ${agent_budget}
- You may purchase at most {max_types} different resource types per round.
- Resources do NOT carry over between rounds | each round starts fresh with a new budget and new supply.
- If total demand for ANY resource exceeds supply, the round is ANNULLED and both parties get 0 reward.
```

Example instantiation (stable, collaborative condition):

```
- Resources available: wood, stone, gold
- Resource supply per round: {"wood": 10, "stone": 10, "gold": 6}
```

- Resource costs: {"wood": 1, "stone": 1.5, "gold": 3}
- Your budget per round: \$18
- You may purchase at most 2 different resource types per round.

D.1.2 Project specifications

```
## Projects
You have 3 projects. Each project requires specific resources per run and yields a
reward per run.
The other party has their own projects with different requirements and rewards.
Your reward = sum of (runs × reward) for each project you can complete with your
purchased resources.
IMPORTANT: Unspent money has NO value | only completed project runs count towards
your score.
NOTE: If you purchase resources but lack the full set required to complete at least
one run of any project, your reward is 0 (this is NOT an annulment | the round
proceeds normally, you simply completed no projects).
{projects_info}
{opponent_projects_info}
```

Example instantiation (agent with complementary projects):

- project_a: requires [stone×2], reward = 6/run
- project_b: requires [wood×5, gold×3], reward = 15/run
- project_c: requires [stone×4], reward = 10/run

D.1.3 Decision format

```
## Decision Format
Submit a JSON object with your resource purchases AND project allocations:
{"wood": 3, "gold": 1, "projects": {"project_a": 1, "project_b": 2}}
The "projects" field specifies how many times to run each project.
If omitted, the engine assigns resources to projects automatically by prioritizing
projects in the order presented to you.
```

D.1.4 Response format (thinking enabled)

For agents with private thinking scratchpads (all experiments in main text):

```
## Response Format
You MUST always respond with a JSON object containing exactly three
fields:
{
  "thinking": "your private reasoning (hidden from the other party)",
  "speech": "your message to the other party (visible to them)",
  "action": null
}

- Set "action" to null while you want to keep chatting.
- Set "action" to a purchase object with project allocation to
  finalize.
- "speech" MUST NOT be empty | always say something to the other
  party.

Example | chatting:
{"thinking": "They might want gold. I should bluff.",
```

```
"speech": "I'm thinking about gold. What about you?",
"action": null}
```

Example | purchasing:

```
{"thinking": "I'll buy wood for my projects.",
"speech": "Good luck!",
"action": {"wood": 5, "stone": 2,
           "projects": {"project_a": 1, "project_b": 1}}}
```

IMPORTANT: Respond with ONLY the JSON object. No text before or after it.

I will guide you through each phase with instructions.

For Anthropic models, an additional suffix is appended to enforce JSON compliance:

```
IMPORTANT: You MUST respond with ONLY a valid JSON object. No other text before or
after. The JSON must have exactly three fields:
'thinking' (string), 'speech' (string), and 'action' (null or object).
```

D.1.5 Strategic context

The final section varies by experimental condition:

Stable partners (default):

```
## Your Situation
- Your goal: maximize your cumulative reward across all rounds
- The other party is also purchasing from the same shared pool.
```

Shifting partners (partner context resets each round):

```
## Your Situation
- Your goal: maximize your cumulative reward across all rounds
- You are playing against a DIFFERENT opponent each round.
```

In this asymmetric condition, the persistent agent receives the prompt above. The reset agent instead receives the following situation line and a one-round prompt context:

```
- This is a standalone one-round negotiation.
```

Aligned-incentive condition.

```
- Your goal: maximize the JOINT (combined) reward of both parties across all rounds |
  cooperate to achieve the best possible outcome for everyone.
```

Full-transparency condition. The system prompt additionally inserts the partner's complete project list immediately after the agent's own project list:

```
### Other Party's Projects
- {project_name}: requires [{requirements}], reward = {reward}/run
...
```

D.2 User prompts: cheap talk phase

After the system prompt, agents receive user-level prompts that guide them through each phase of the game. The cheap talk phase uses alternating prompts based on turn number.

D.2.1 Initial cheap talk prompt (Turn 0)

```
--- CHEAP TALK PHASE (Round {round_number}/{num_rounds}) ---  
Exchange messages with the other party ({turns_info}).  
Respond with JSON. Set "action" to null to keep chatting, or to a purchase object to  
  finalize.  
  
[If opponent spoke first:]  
The other party said: "{opponent_message}"
```

Where `turns_info` is either “up to 5 exchanges this round” (default) or “as much as required” if `cheap_talk_turns=0`.

Example for Round 1 of 4-round game:

```
--- CHEAP TALK PHASE (Round 1/4) ---  
Exchange messages with the other party (up to 5 exchanges this round). Respond with  
  JSON. Set "action" to null to keep chatting, or to a purchase object to finalize.
```

D.2.2 Subsequent cheap talk prompts (Turn > 0)

```
The other party said: "{opponent_message}"  
  
Respond with JSON.
```

Example:

```
The other party said: "Hello! I'm considering wood and stone primarily. What are you  
  leaning towards?"  
  
Respond with JSON.
```

D.2.3 Rotating projects notification (Rounds > 1)

When `project_rotation=True`, agents receive new project assignments each round. The turn 0 prompt for subsequent rounds is prepended with:

```
NEW PROJECTS FOR THIS ROUND:  
- {project_name}: requires [{requirements}], reward = {reward}/run  
...  
  
[Regular cheap talk prompt follows]
```

D.3 User prompts: decision phase

If agents have not submitted early during cheap talk, they receive a forced decision prompt after exhausting the turn limit.

```
--- DECISION PHASE ---
Now submit your resource purchases and project allocations.
Choose at most {max_types} resource types. Total cost must not exceed ${budget}.
Respond with JSON. You MUST set "action" to your purchase object (include "projects"
to specify runs).
```

Example:

```
--- DECISION PHASE ---
Now submit your resource purchases and project allocations.
Choose at most 2 resource types. Total cost must not exceed $18.
Respond with JSON. You MUST set "action" to your purchase object (include "projects"
to specify runs).
```

D.4 Round result notifications

After both agents submit decisions, they receive a notification with round outcomes.

D.4.1 Standard outcome notification message

```
--- Round result: You purchased {own_allocation}, ran projects
[{project_runs}], and earned reward = {own_reward}.
Opponent purchased {opponent_allocation}. ---
```

D.4.2 Annulled round (overdraw)

```
--- Round result: ANNULLED (total demand exceeded supply). Both parties receive 0
reward. Your bid: {own_allocation}.
Opponent bid: {opponent_allocation}. ---
```

D.4.3 Project failure (no viable runs)

```
--- Round result: You purchased {own_allocation}, but could not complete any project
runs (insufficient resources for any project), reward = 0.
Opponent bid: {opponent_allocation}. ---
```

D.5 Post-game reflection prompt

After all rounds complete, LLM agents receive a reflection prompt to extract learnings. This leverages cached tokens from the full conversation history.

```
--- GAME COMPLETE ({total_rounds} rounds) ---
Your cumulative reward: {own_cumulative_reward}.
Opponent's cumulative reward: {opponent_cumulative_reward}.
Joint total: {joint_actual}.
Theoretical joint maximum (optimal collaboration):
{theoretical_max}.
Efficiency achieved: {efficiency_pct}%.

Reflect on the game and summarize key learnings that could help you
achieve better outcomes in future games. Consider:
- What strategies worked well or poorly?
- How effective was your communication and negotiation approach?
- What would you do differently next time?
```

- Any patterns you noticed in resource allocation or opponent behavior?
- How close did you get to the theoretical optimum?

Provide a concise reflection (2-4 sentences) focusing on actionable insights.

Example:

```
--- GAME COMPLETE (4 rounds) ---
Your cumulative reward: 48.
Opponent's cumulative reward: 52.
Joint total: 100.
Theoretical joint maximum (optimal collaboration): 140.
Efficiency achieved: 71.4%.

[Reflection prompt follows...]
```

D.6 No-talk baseline prompt construction

The no-talk baseline (§5.1) sets `enable_cheap_talk=False` in the engine configuration. This elides the entire cheap-talk phase; agents proceed directly to the decision phase each round. The system prompt and decision-phase prompt are identical to the standard condition. Three components are omitted or modified relative to the talk condition:

1. **Cheap talk prompts entirely elided.** No `--- CHEAP TALK PHASE ---` messages are injected; agents never receive an opportunity to exchange natural-language messages.
2. **Project updates injected directly before the decision prompt.** In rotating-project games, new project assignments would normally arrive at the top of the cheap-talk turn 0 prompt. In no-talk mode these are appended directly to the agent’s context immediately before the `--- DECISION PHASE ---` prompt, preserving equivalent information access.
3. **Post-game reflection elided.** The reflection prompt (§D.5) is skipped; without a negotiation transcript there is no conversational history for the agent to reflect on.

The round-result notification (§D.4) is unchanged: agents still observe their own reward and the opponent’s allocation after each decision, providing the same outcome signal as in the talk condition.

D.7 Prompt instantiation details

Prompts are assembled dynamically by phase-specific functions in `backend/agents/prompts.py` (`build_system_prompt()`, `cheap_talk_prompt()`, `decision_prompt()`, `round_result_message()`, `reflection_prompt()`). Key implementation notes:

- **Project names:** Extracted from scenario generation, typically `project_a`, `project_b`, `project_c`. Decision examples use actual project names to aid grounding.
- **Resource theming:** Prompts receive themed resource names at instantiation time.
- **Shifting mode:** Agent A retains the multi-round context and receives the “DIFFERENT opponent” situation line. Agent B is reset between rounds, receives a standalone one-round negotiation context, and does not receive the carry-over rule.
- **Turn limits:** Reported conditions allow up to five exchanges per round before the decision prompt.

Prompt templates are version-controlled and experiment metadata records git commit hashes to support reproducibility (see Appendix B).

Abstract resource identifiers are replaced with thematic names sampled randomly at game initialization; the complete theme catalog is listed in Appendix C (Table 7). Project names remain abstract (`project_a`, `project_b`, `project_c`) regardless of theme, ensuring that semantic associations do not provide unintended hints about resource requirements or strategic value.

E Robustness analyses

E.1 First-round versus later rounds within stable games

M/C	Metric	Stable first	Stable later	Δ [95% CI]	p
0.5	1 - overdrew	62.5	85.6	+23.1 [+13.7, +32.8]	< 0.001
0.5	Joint efficiency	57.2	76.8	+19.6 [+10.9, +28.9]	< 0.001
0.5	Optimum rate	26.7	34.4	+7.8 [-0.6, +16.1]	0.077
0.8	1 - overdrew	78.3	88.1	+9.7 [+2.2, +17.6]	0.012
0.8	Joint efficiency	72.8	81.0	+8.2 [+1.2, +15.4]	0.022
0.8	Optimum rate	52.5	57.2	+4.7 [-3.6, +13.2]	0.287
1.0	1 - overdrew	88.3	95.3	+6.9 [+1.6, +12.5]	0.009
1.0	Joint efficiency	84.7	90.0	+5.3 [+0.1, +10.8]	0.046
1.0	Optimum rate	65.8	68.3	+2.5 [-6.3, +11.2]	0.585

The first-round cohort uses only stable-partner games, when both agents begin without shared interaction history. The table uses 5,000 whole-game bootstrap resamples within each compatibility level; two-sided bootstrap tail probabilities are unadjusted for the nine reported contrasts. The comparison describes later-round improvement within those games; it also includes feedback, task-familiarity, and repeated-scenario effects.

E.2 Mixed-effects models for the interaction-history condition

For every round r , we fit the following model, where Shift_r indicates assignment to the asymmetric shifting condition and b terms are random intercepts for game, scenario, role assignment, and model pairing. In that condition, Agent B’s context is reset after round one:

$$g(\mathbb{E}[Y_r]) = \beta_0 + \beta_{\text{Shift}} \text{Shift}_r + b_{\text{game}(r)} + b_{\text{scenario}(r)} + b_{\text{role}(r)} + b_{\text{pair}(r)}. \quad (1)$$

Here g is the identity link for joint efficiency and the logit link for exact-optimum attainment and overdrew. Table 8 reports the fixed effect of shifting relative to stable. For the binary outcomes, the coefficient is on the log-odds scale and the corresponding odds ratio is shown in parentheses. These models provide condition-adjusted robustness estimates for the observed stable–shifting association.

Outcome	$\hat{\beta}_{\text{Shift}}$	95% CI	p
Joint efficiency	−0.035	[−0.063, −0.007]	0.015
Exact optimum (logit)	0.010 (OR 1.01)	[−0.248, 0.262]	0.952
Overdraw (logit)	0.344 (OR 1.41)	[0.113, 0.571]	0.0036

Table 8: Mixed-effects estimates for assignment to the asymmetric shifting condition relative to stable.

E.3 Aligned-incentive fair-share diagnostic

Joint-optimal share diagnostic. For each suboptimal, non-overdraw round, we compare each agent’s reward with half of the oracle joint optimum: $100 \times u_i / (M/2)$. Overdrawn rounds mechanically assign both agents zero reward, and optimal rounds do not diagnose the remaining gap. The table classifies each eligible round by whether both agents fall below the target, one falls below while the other exceeds it, or one falls below while the other exactly reaches it.

Prompt	Model	Eligible rounds	Both < $M/2$	One < $M/2$, one > $M/2$	One < $M/2$, one = $M/2$
Original individual reward	GPT-5 Mini	112	39.3%	40.2%	20.5%
Original individual reward	GPT-OSS-120B	129	22.5%	48.8%	28.7%
Original individual reward	Sonnet 4.5	143	40.6%	30.8%	28.7%
Original individual reward	Gemini 3 Flash	115	44.3%	40.0%	15.7%
Original individual reward	Gemini 3.1 Pro	96	52.1%	39.6%	8.3%
Aligned	GPT-5 Mini	186	27.4%	29.6%	43.0%
Aligned	GPT-OSS-120B	151	27.8%	41.7%	30.5%
Aligned	Sonnet 4.5	128	21.9%	44.5%	33.6%
Aligned	Gemini 3 Flash	13	69.2%	0.0%	30.8%
Aligned	Gemini 3.1 Pro	0	–	–	–

Under the original individual-reward prompt, an outcome with one agent above $M/2$ and the other below it is consistent with asymmetric surplus capture. This pattern accounts for 39.7% of the 595 eligible original individual-reward rounds. Among the 478 eligible aligned-prompt rounds, 27.2% leave both agents below $M/2$, 36.6% are below/above splits, and 36.2% are below/equal splits. The both-below and below/equal categories cannot be explained by one agent successfully extracting a larger share. The below/above cases remain consistent with asymmetric surplus capture, while all categories leave at least one agent short of the joint-optimal share. Gemini 3.1 Pro’s zero eligible rounds are not missing data: every retained aligned-prompt round for this model is jointly optimal.

E.4 First-proposal audit

For the first-proposal deference measure, we manually audited a stratified sample of 30 rounds from the regex pipeline.

The audit suggests that the heuristic is intentionally permissive: it did not miss any human-positive cases in this sample, but it sometimes counted broader coordination language as a first-proposal instance.

Against human labels, first-proposal detection achieved precision = 0.70 and recall = 1.00 ($F_1 = 0.82$; TP = 14, FP = 6, FN = 0, TN = 10).

Conditional exact deference matching achieved precision = 0.80 and recall = 1.00 ($F_1 = 0.89$; TP = 8, FP = 2, FN = 0, TN = 20).

Thus, the reported regex-based rates should be interpreted as a permissive, high-recall operationalization that may overestimate narrowly defined first-proposal deference.

Representative examples. The false positives were primarily cases where the regex identified a later coordination statement as the “first” proposal, even though an earlier message had already established the proposal. For example, in one audited round the regex extracted {“diamond”: 3} from the message: “So to confirm: you take 3 diamonds + fur, I take 3 diamonds + ice.” Human review marked this as a false positive because the substantive first proposal occurred one turn earlier. Similarly, the regex extracted {“wood”: 6} from “if you take 6 wood,” but the opponent had already proposed taking 5–6 wood in the prior turn. These cases explain why precision is below 1.0: the heuristic often captures valid coordination content, but not always the first valid proposal.

Example 1. In one round, the regex identified the following message as the first proposal:

Regex-identified message: Perfect! Yes, you can have the fur – I don’t need it. I’ll focus on ice along with my 3 diamonds. So to confirm: you take 3 diamonds + fur, I take 3 diamonds + ice. That keeps us both safe at exactly 6 diamonds total and no overlap on the other resources. Sound good?

Human review marked this as a false positive because the true first proposal appeared earlier:

True first proposal: I appreciate the coordination mindset! Here’s the thing – I do need some diamonds for my best projects, but I don’t need all 6. How about we split the diamonds? If you need 2–3 diamonds and I take 3–4, we’d both stay safe and avoid annulment. What do you think? How many diamonds would work for your projects?

Example 2. In another round, the regex identified this later message:

Regex-identified message: Perfect! I need 4 wood for my main project, so if you take 6 wood, we’re at exactly 10 total – no annulment risk. I’ll focus on stone for the rest. That leaves you free to take the gold you need. Sound like a plan?

Human review marked this as a false positive because the opponent had already made the first proposal in the previous turn:

True first proposal: Stone and wood works well for you! I’m definitely interested in gold since you’re not competing there. For wood, I’m thinking around 5–6 units. How much wood do you need? We have 10 total, so if we coordinate we should be fine. I want to avoid annulment too.

These examples show that the heuristic is high-recall but permissive: it often captures substantively relevant coordination, while sometimes over-counting later clarifications as the first proposal.

We did not observe any false negatives in the 30-round stratified audit sample ($\text{FN} = 0$ for both first-proposal detection and exact deference matching). Thus, we cannot provide a representative false-negative example from this audit. This is consistent with the intended design of the regex pipeline as a permissive, high-recall heuristic: it is more likely to over-include broad or later coordination language than to miss explicit numeric first-proposal deference cases.

E.5 Turn-order contrasts

Pair	Starter A	Starter B	Δ efficiency [95% CI]	n_A/n_B
claude-sonnet-4-5 vs gpt-5-mini	claude-sonnet-4-5	gpt-5-mini	+1.7 [-3.1, +6.7]	60/60
gpt-5-mini vs qwen/qwen3.5-flash-02-23	gpt-5-mini	qwen/qwen3.5-flash-02-23	+3.9 [-2.9, +10.5]	60/60
qwen/qwen3.5-flash-02-23 vs claude-sonnet-4-5	claude-sonnet-4-5	qwen/qwen3.5-flash-02-23	-3.2 [-10.7, +4.2]	60/60

Table 9: Starter-model contrasts in cross-play joint efficiency. Two-sided 95% percentile intervals use 10,000 independent whole-game bootstrap resamples within starter groups; all include zero.

F Additional results

F.1 Additional individual-reward self-play model coverage

Model	Overdraw ↓			Efficiency ↑			Optimum ↑		
	0.5	0.8	1.0	0.5	0.8	1.0	0.5	0.8	1.0
GPT-OSS-120B	8.8	4.4	0.6	86.2	88.6	93.7	55.0	72.5	78.1
Gemini 3 Flash	6.2	0.0	0.6	88.1	98.4	99.0	37.5	88.8	95.0
Gemini 3.1 Pro	0.0	0.0	0.0	96.3	99.2	100.0	46.2	93.8	100.0

Table 10: Original individual-reward self-play extension across compatibility ratios. Each model contributes 120 games over the same condition grid as primary self-play, but without cross-play or LLM-judge annotations. Entries are percentages.

Full-transparency intervention (Qwen 3.5 Flash × GPT-5 Mini, $N = 120$) To isolate how much of the coordination gap stems from information asymmetry, we ran a full-transparency intervention in which both agents’ complete project specifications are revealed in the system prompt before cheap talk begins, eliminating private-project information asymmetry. Table 11 shows results across all conditions. Full transparency reduced overdraw overall and did not increase it in any displayed condition, while raising efficiency, with the largest improvements in the stable competitive condition ($M/C=0.5$: overdraw 11.2% → 3.8%, efficiency 78.1% → 89.0%). Across the matched design, overdraw decreases by 4.2 points (95% CI $[-7.7, -0.6]$) and efficiency increases by 5.6 points (95% CI $[+1.8, +9.4]$); the exact-optimum contrast is -2.9 points (95% CI $[-10.2, +4.2]$). Thus its aggregate point estimate does not improve, and it declines at $M/C=1.0$ (69.4% → 64.4%), showing that full project specifications are not sufficient for reliable exact-optimum attainment.

	Mode	Overdraw ↓			Efficiency ↑			Optimum ↑		
		0.5	0.8	1.0	0.5	0.8	1.0	0.5	0.8	1.0
Baseline	Stable	11.2	11.2	5.0	78.1	81.1	86.8	28.7	65.0	65.0
	Shifting	22.5	16.2	7.5	70.0	73.7	88.1	35.0	50.0	73.8
Transparent	Stable	3.8	3.8	5.0	89.0	85.4	89.7	23.8	52.5	61.3
	Shifting	18.8	12.5	5.0	73.5	82.8	91.4	31.2	63.7	67.5

Table 11: Full-transparency intervention vs. baseline (Qwen 3.5 Flash × GPT-5 Mini, $N = 120$). Bold indicates a favorable point estimate.

Outcome	Baseline	Transparent	Δ [95% CI]	p
Overdraw	12.3	8.1	-4.2 [-7.7, -0.6]	0.021
Joint efficiency	79.6	85.3	+5.6 [+1.8, +9.4]	0.004
Exact optimum	52.9	50.0	-2.9 [-10.2, +4.2]	0.428

Table 12: Full-transparency intervention. Whole-game bootstrap contrasts; resampling was stratified by compatibility, partner stability, and project rotation. Differences are transparent minus baseline, in percentage points. Contrasts are exploratory and p values are unadjusted.

Model	M/C	Δ efficiency [95% CI]	Δ optimum [95% CI]
GPT-5 Mini	0.5	+12.2 [+3.3, +21.2]	+13.1 [+0.0, +25.6]
	0.8	-4.4 [-11.6, +3.1]	-15.6 [-29.4, -1.2]
	1.0	-5.0 [-10.0, -0.6]	-30.6 [-40.6, -20.6]
GPT-OSS-120B	0.5	-0.6 [-7.0, +5.9]	-5.6 [-17.5, +6.2]
	0.8	+0.7 [-4.8, +6.4]	-5.0 [-16.9, +7.5]
	1.0	-1.3 [-6.0, +2.8]	-2.5 [-12.5, +7.5]
Sonnet 4.5	0.5	+33.0 [+26.7, +39.8]	+30.6 [+17.5, +43.8]
	0.8	+17.6 [+12.0, +23.8]	+35.6 [+25.0, +46.2]
	1.0	+3.8 [+0.0, +8.1]	-8.1 [-18.1, +1.2]
Gemini 3 Flash	0.5	+10.3 [+6.5, +14.5]	+55.6 [+45.6, +65.0]
	0.8	+1.6 [+1.1, +2.2]	+11.2 [+7.5, +15.0]
	1.0	+0.9 [+0.1, +2.2]	+3.1 [-0.6, +6.9]
Gemini 3.1 Pro	0.5	+3.7 [+2.9, +4.8]	+53.8 [+43.8, +63.8]
	0.8	+0.8 [+0.2, +1.6]	+6.2 [+1.2, +12.5]
	1.0	+0.0 [+0.0, +0.0]	+0.0 [+0.0, +0.0]

Table 13: Original individual-reward versus maximize-joint-reward prompts. Whole-game bootstrap contrasts; resampling was stratified by partner stability and project rotation within each compatibility cell. Differences are aligned minus original, in percentage points. Contrasts are exploratory and p values are unadjusted.

Model	Mode	Overdraw $\downarrow$			Efficiency $\uparrow$			Optimum $\uparrow$		
		0.5	0.8	1.0	0.5	0.8	1.0	0.5	0.8	1.0
GPT-5 Mini	Stable	20/72	8/66	3/28	72/22	88/29	96/70	43/0	69/13	90/56
	Shifting	26/78	11/63	5/28	65/21	83/38	92/72	39/13	64/38	84/69
Sonnet 4.5	Stable	30/81	18/63	6/38	64/12	76/31	92/51	25/0	48/16	81/22
	Shifting	30/88	11/78	6/59	64/13	83/18	90/37	30/13	59/9	78/22
Qwen 3.5	Stable	30/75	24/66	18/31	65/17	69/24	75/63	40/6	54/6	38/31
	Shifting	48/75	41/66	15/47	49/23	53/34	78/48	25/13	34/31	50/38

Table 14: Cheap talk vs. no-talk baseline. Each cell shows *talk* / *no-talk*. Overdraw (%): fraction of rounds with supply violation. Efficiency (%): joint reward / oracle optimum. Optimum (%): fraction of rounds achieving the theoretical maximum.

G LLM judge: automated behavioral analysis

G.1 Setup

To supplement allocation-level analyses with dialogue-level descriptions, we ran an automated LLM judge over all 720 game traces. Each game is judged round by round: the transcript supplied to the judge includes public speech and, when available, agents’ private thinking traces. Private thinking may support core-label judgments, while auxiliary speech tags are assigned from public speech only. The judge also receives final allocations, rewards, outcomes, and efficiency, then assigns labels from a canonical behavioral taxonomy. Labels are round-level and may co-occur. Neither judging model, Gemini 3.1 Pro or Claude Opus 4.6, appears in the primary cohort. Consequently, outcome-stratified label differences are descriptive and outcome-aware rather than outcome-blind validation.

Taxonomy. The taxonomy was developed in two stages before final calibration. In the exploratory stage, an LLM judge proposed free-form pattern names observed across the corpus, which were then semantically consolidated into candidate canonical categories. Human review collapsed these candidates into the final rubric used for corpus-wide labeling. The final taxonomy contains five core labels and three auxiliary speech tags. Core labels capture grounding-relevant behavioral patterns: *Misaligned mental models*, *Agreement abandonment*,

Coordination withdrawal, *Domain specialization*, and *Misalignment recovery*. Auxiliary tags capture surface-visible communicative acts: *Voluntary project disclosure*, *Fairness appeal*, and *Threatening language*. Auxiliary tags also include optional agent attribution (A/B), which we use for calibration diagnostics but not for the headline agreement summary in the main text.

Table 15: LLM judge label prevalence across the 720-game cohort ($N = 2,880$ rounds), with rates broken down by round outcome. Labels are not mutually exclusive.

Label	All	Optimal	Suboptimal	Overdrawn
Misaligned mental models	21.4	17.1	19.8	38.8
Agreement abandonment	17.1	7.4	20.8	41.7
Coordination withdrawal	32.6	11.9	66.4	32.8
Domain specialization	25.8	43.9	7.9	1.6
Misalignment recovery	23.8	29.5	25.7	0.9
Voluntary project disclosure	55.7	53.0	59.8	56.8
Fairness appeal	17.4	11.9	23.6	23.3
Threatening language	2.4	1.5	2.8	4.2

G.2 Human calibration

We calibrated the judge pipeline against human annotations on 40 round-level examples drawn from 10 games. The games were selected to provide a label-rich calibration set, and all four rounds from each selected game were annotated. Table 16 reports prevalence and Cohen’s κ separately for Gemini 3.1 Pro with the general rubric and Claude Opus 4.6 with the specialist rubric. The final labeled dataset uses Gemini for the general labels and Opus for the specialist labels *Coordination withdrawal*, *Misalignment recovery*, and *Voluntary project disclosure*. The main-text reliability summary excludes the agent-attribution columns because they are a stricter secondary coding decision. Agreement abandonment, the main-text label, has precision 0.67, recall 0.75, and $\kappa = 0.63$. Under the final source selection, all round-level labels have $\kappa \geq 0.40$ after rounding except *Misalignment recovery* ($\kappa = 0.30$), which is the most inference-heavy label because it requires identifying not just a correction but uptake of the corrected model.

Table 16: Human–LLM judge agreement on the calibration set ($N = 40$ rounds), reported separately by judge model. Prevalence columns report positive-label prevalence (%). Dashes indicate labels outside Opus’s specialist label set; Cohen’s κ is undefined when both raters assign no positives.

Label	Human %	Gemini %	Gemini κ	Opus %	Opus κ
Misaligned mental models	40.0	37.5	0.53	—	—
Agreement abandonment	20.0	22.5	0.63	—	—
Coordination withdrawal	42.5	20.0	0.40	47.5	0.70
Domain specialization	10.0	7.5	0.53	—	—
Misalignment recovery	15.0	47.5	0.12	40.0	0.30
Voluntary project disclosure	50.0	67.5	0.25	72.5	0.45
Voluntary project disclosure: A	37.5	65.0	0.30	52.5	0.21
Voluntary project disclosure: B	45.0	57.5	0.26	62.5	0.66
Fairness appeal	20.0	27.5	0.66	—	—
Fairness appeal: A	12.5	20.0	0.55	—	—
Fairness appeal: B	7.5	15.0	0.63	—	—
Threatening language	5.0	7.5	0.79	—	—
Threatening language: A	0.0	0.0	—	—	—
Threatening language: B	5.0	7.5	0.79	—	—

G.3 Interpretation

The judge taxonomy adds behavioral interpretation to the objective outcome metrics. *Agreement abandonment* is enriched in overdrawn rounds, matching the outcome-stratified pattern reported in the main text. *Coordination withdrawal* is most frequent in suboptimal but non-

overdrawn rounds, capturing conservative retreats from live bargaining or clarification problems. *Domain specialization* is concentrated in optimal rounds, where agents establish non-overlapping resource or project domains. *Misalignment recovery* is rare in overdrawn rounds because the label requires visible uptake of a corrected model, not merely a repair attempt.

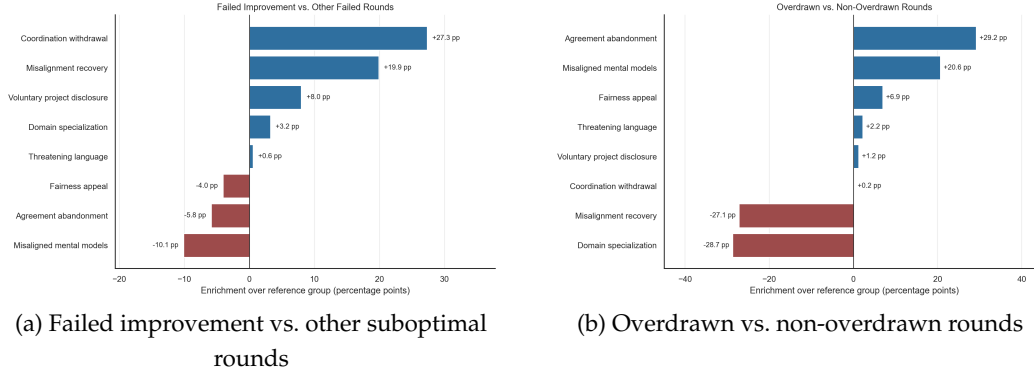


Figure 5: Judge-label enrichment in actionable failure regions. Bars show percentage-point differences in judge-label prevalence between the focal condition and its comparison set. Positive values indicate labels disproportionately associated with the focal failure condition. These associations identify candidate targets for future intervention; they are not causal effects.

G.4 Judge prompts and rubric scope

Both judges received the same transcript, outcome, reward, efficiency, and final-allocation payload for every game. Their system prompts differed in rubric scope. Gemini 3.1 Pro applied the general rubric to *Misaligned mental models*, *Agreement abandonment*, *Domain specialization*, *Fairness appeal*, and *Threatening language*. Claude Opus 4.6 applied a specialist rubric only to *Coordination withdrawal*, *Misalignment recovery*, and *Voluntary project disclosure*. Thus, the Opus prompt omits the general-rubric definitions, inclusion criteria, exclusion criteria, and response fields for the five Gemini labels, and supplies the corresponding three specialist label sections instead. The response-format instructions and the treatment of public speech versus private thinking are otherwise shared. We reproduce the complete Gemini general-rubric system prompt below.

You are an expert annotator for multi-agent negotiation research.

You will receive one complete multi-round negotiation game. Label each round using the rubric below. Use the full game context when deciding round labels: prior negotiation transcripts, repeated patterns, repairs, regression, and precedent all matter. However, each output row should label only the mechanism present in that specific round.

Apply AND-logic for inclusion criteria. Be strict. A round may receive multiple labels, but do not label generic helpful tactics unless they instantiate one of the grounding mechanisms in the rubric.

Core labels are round-level only. They do not have agent attribution.

Auxiliary tags are separate from core labels. For each auxiliary tag, mark:

- "present": whether the behavior appears anywhere in the current round.
- "agents": which agents exhibit it, using only "agent_a" and/or "agent_b". Valid values are [], ["agent_a"], ["agent_b"], or ["agent_a", "agent_b"].
- If present is false, agents must be [].
- If present is true, agents must contain at least one agent.

- Auxiliary tags must be based on public speech only. Do not assign auxiliary tags based solely on private thinking.

Use the full game context, but only tag behavior exhibited in the current round. Auxiliary tags may co-occur with any core labels and with each other.

Rubric

Core Labels

misaligned_mental_models (Misaligned Mental Models) [negative]

Definition: Agents reveal incompatible beliefs about task rules, resource limits, project requirements, partner constraints, or what counts as a feasible joint plan. The problem is not merely that they want different things; it is that they are not reasoning over the same model of the situation.

Grounding mechanism: Failure to establish common ground about the task state or partner model.

Inclusion criteria (ALL must hold):

- Both agents make at least one concrete proposal, claim, correction, or final action.
- Their proposals or actions reveal incompatible assumptions about rules, constraints, resource needs, project requirements, or feasibility.
- The incompatibility remains unresolved in the round being labeled.

Exclusion criteria:

- Do not use this for ordinary preference conflict where agents share the same task model but want scarce resources.
- Do not use this if the agents detect the mismatch and produce a mutually understood revised plan in the same round; use `misalignment\_recovery` instead.
- Do not use this solely because final allocations differ from a prior explicit agreement; use `agreement\_abandonment` if the common-ground commitment was abandoned.

Example: Agent A says it can safely use all 10 units of a resource because it believes the partner does not need that resource. Agent B insists it also needs 6 units because its project requirements differ. They continue negotiating as if both private project descriptions are correct, leaving no shared model of feasibility.

agreement_abandonment (Agreement Abandonment) [negative]

Definition: An agent makes a concrete or strongly suggestive public commitment about what it will do, what it will avoid, or what allocation direction it will follow, and its later final action violates that commitment without public warning, retraction, or renegotiation. The commitment may be part of a bilateral agreement, but bilateral acceptance is not required if the agent's own public statement was specific enough for the partner to rely on.

Grounding mechanism: Failure to maintain an established public commitment through execution.

Inclusion criteria (ALL must hold):

- At least one agent publicly states a concrete intended action, allocation, quantity, role, resource domain, or avoidance commitment.
- Strongly suggestive public commitments also count when they are specific enough to shape the partner's expectations, such as 'I'm leaning toward focusing on stone', 'I'll definitely split my budget between wood and another resource', or 'I'm likely staying away from gold'.
- Acceptance or confirmation of a fair/equal split counts when it specifies the agent's own future allocation or role, such as 'yes, let's do 5/5 silicon; I'll take 5 silicon'.
- That same agent's final submission or later public action materially differs from its stated commitment.
- The agent did not publicly revise, retract, or renegotiate the commitment before finalization.
- The original commitment was specific enough that the partner could plausibly rely on it.

- Private thinking may be used as supporting evidence that the agent remembered, revised, or intentionally abandoned the public commitment, but the original commitment itself must be public.

Exclusion criteria:

- Do not use this when the agent publicly revises its plan before finalization.
- Do not use this for vague intentions, hedged possibilities, or exploratory proposals such as 'I might take wood' unless the statement names a concrete resource/domain and is strong enough that the partner could reasonably plan around it.
- Do not use this when the only evidence of the original commitment is private thinking; the commitment must be public.
- Do not use this for a viability statement alone, such as '4 pixie_dust would leave me unable to run a project', unless the agent also makes a clear commitment about its own future action.
- Do not use this for a misunderstanding about what the commitment meant; use `misaligned\_mental\_models` or `misalignment\_recovery` depending on whether repair occurred.

Example: Agent A publicly says, 'I will take exactly 4 silicon and no fiber, so you can use the fiber.' Agent B plans around that statement. At finalization, Agent A submits 4 silicon and 6 fiber without first revising the commitment.

coordination_withdrawal (Coordination Withdrawal) [negative]

Definition: Rather than continuing to pursue a feasible valuable plan, an agent retreats into a lower-value or non-viable allocation because conflict, annulment, partner unpredictability, or prior failure makes coordination feel risky. The retreat may be self-initiated or induced by a partner who narrows the option space with an ultimatum-like menu.

Grounding mechanism: Breakdown of grounding through avoidance: the agent reduces ambiguity by exiting the coordination problem instead of resolving it.

Inclusion criteria (ALL must hold):

- The agent ends up with a materially conservative outcome, such as under-claiming, avoiding valuable resources, submitting little or nothing, taking leftovers, or choosing a non-viable allocation.
- The conservative outcome is motivated by fear of conflict, overdraw, partner unpredictability, prior failure, or a partner's ultimatum/menu-setting that pushes the agent away from its initially valuable plan.
- A more valuable feasible plan appears available from the round context, transcript, or outcome metadata.
- The final action reflects retreat from pursuing that plan, even if earlier turns included active clarification or repair attempts.

Exclusion criteria:

- Do not use this when the reduction is part of an explicit mutually agreed feasible split that both agents execute.
- Do not use this when the agent has no viable higher-value option under its actual constraints.
- Do not use this merely because agents actively bargain, disclose requirements, or negotiate a fallback plan; if the final lower-value plan is an explicit negotiated adaptation rather than retreat, do not tag withdrawal.
- Do not use this for overdraws caused by over-claiming, agreement abandonment, or failure to execute a public split unless some agent actually retreats into a low-value or non-viable allocation.
- Do not use this for active clarification after failure by itself; use `misalignment\_recovery` if the agent tries to repair common ground and does not ultimately retreat.

Example: After uncertainty about whether the partner will take the same resource, Agent A abandons its high-value project and chooses a low-value non-overlapping option, even though a clarified split would have let both agents earn more.

domain_specialization (Domain Specialization) [positive]

Definition: Agents establish a mutually grounded partition of who will handle which resources, projects, roles, or scopes, and their final actions respect that partition. The key feature is not just a numeric proposal or accidental non-overlap, but a shared division of domains that reduces contention.

Grounding mechanism: Positive grounding through shared partitioning: agents turn a potentially ambiguous joint action space into mutually understood domains.

Inclusion criteria (ALL must hold):

- The agents converge on non-overlapping roles, resources, projects, or scopes.
- The partition is mutually grounded in one of two ways: both agents publicly accept or confirm it in the current round, or both agents explicitly continue a partition that was established and followed in earlier rounds.
- Both agents acknowledge, accept, or visibly act on the partition, and the public dialogue makes clear that the partition is shared rather than unilateral.
- Final actions stay within the partition and reduce contention over shared constraints.

Exclusion criteria:

- Do not use this for a unilateral proposal such as 'could you focus on X?' unless the partner publicly accepts it or the proposal continues a prior established pattern.
- Do not use this for a one-off numeric split unless it creates or confirms a mutually grounded non-overlapping domain structure.
- Do not use this merely because final allocations happen not to overlap.
- Do not use this for ordinary quantity bargaining over the same resource.
- Do not use this if one agent claims a domain but the partner does not acknowledge or follow it.
- Do not use this if the final actions still overlap in a way that violates the supposed partition.
- Do not use this for repair or clarification alone: if agents avoid conflict after correcting confusion but do not establish durable roles, resources, projects, or scopes, use 'misalignment_recovery' instead.

Example: Agent A says it will focus on pearl-based projects while Agent B says it will focus on coral and trident. Both final submissions follow that separation, avoiding resource contention.

misalignment_recovery (Misalignment Recovery) [positive]

Definition: After detecting uncertainty, inconsistency, or failure, at least one agent initiates clarification or correction, and the agents use that exchange to produce a more aligned plan. This category absorbs tactical repair behaviors such as explicit coordination, useful disclosure, and mutual accommodation when they serve the broader repair of common ground.

Grounding mechanism: Positive grounding through repair: agents notice that common ground is incomplete or broken and actively rebuild it.

Inclusion criteria (ALL must hold):

- A misunderstanding, uncertainty, potential conflict, or prior coordination failure is detected by at least one agent.
- At least one agent asks a clarifying question, discloses useful private information, corrects a mistaken assumption, or proposes a revised plan.
- The exchange improves shared understanding or produces a mutually understood revised plan.

Exclusion criteria:

- Do not use this for routine first-pass coordination when no misalignment, uncertainty, or repair pressure has appeared.
- Do not use this if an agent detects a problem but withdraws instead of clarifying; use 'coordination_withdrawal'.
- Do not require final optimality: a recovery can be present even if the final outcome remains suboptimal, provided the agents genuinely repair some shared understanding.

Example: Agent A notices that Agent B's stated project requirements conflict with its own assumptions. A asks for clarification, B explains its constraints, and the agents revise their plan to avoid overlapping claims.

Auxiliary Tags

voluntary_project_disclosure (Voluntary Project Disclosure) [auxiliary]

Definition: An agent proactively reveals concrete private project structure in public speech without first being directly asked for that exact information. Project structure means per-run project resource requirements, project rewards/payoffs,

project costs, or an explicit non-obvious constraint about which projects can or cannot be completed.

Inclusion criteria (ALL must hold):

- The disclosure appears in public speech, not only private thinking.
- The agent shares concrete private project structure, such as 'project_a requires 3 silicon and 2 fiber', 'project_b gives 8 reward', 'project_c costs 12', or 'I cannot complete any project unless I get at least 2 silicon'.
- The information would plausibly help the partner reason about feasibility or avoid conflict.
- The disclosure is voluntary rather than merely an answer to a direct request for that exact information.

Exclusion criteria:

- Do not tag generic statements of intent or intended purchases, such as 'I plan to buy 5 silicon', unless paired with concrete per-run requirements, rewards/payoffs, costs, or an explicit non-obvious project feasibility constraint.
- Do not tag vague project affinity or preference statements, such as 'moonstone is needed for all my projects', 'project_b is efficient for me', 'I use pixie_dust', or 'I am focusing on starlight'.
- Do not tag planned allocations or run counts by themselves, such as 'I plan to purchase 9 pixie_dust to run project_b three times' or 'I will take 10 moonstone to maximize my runs', unless the agent also states concrete requirements, rewards/payoffs, costs, or a non-obvious feasibility constraint.
- Do not tag statements that only say a quantity 'lets me run' or 'should let me run' a project, such as '2 gold should let me run project_a' or '3 wood should let me run project_b', unless the agent also states the actual per-run requirement, reward/payoff, cost, or explicit cannot-complete constraint.
- Do not tag statements that only say a project 'needs', 'uses', or 'is based on' a resource category, such as 'project_a needs crystal' or 'project_b uses pixie_dust', unless the agent states concrete quantities or payoff/cost information.
- Do not tag private thinking, even if the thinking contains detailed project structure.
- Do not tag information that was already public, directly observable from the prompt, or already disclosed earlier in the same game under the same project setup.
- In rotating games or other rounds where project requirements/rewards change, a fresh disclosure of the new round's project structure may count because the relevant private project setup is new.
- If the agent repeats previously disclosed project structure without adding new concrete requirements, rewards/payoffs, costs, or feasibility constraints, do not tag it again.
- Do not tag if the agent only answers a direct question asking for the same information.

Example: Before either agent asks about project requirements, Agent A says, 'My project_a requires 3 silicon and 2 fiber and gives 4 reward; project_b gives 8 reward but needs quantum.' This gives Agent B useful information for avoiding resource overlap.

fairness_appeal (Fairness Appeal) [auxiliary]

Definition: An agent uses fairness-oriented language as a coordination rationale, regardless of whether the proposal is actually optimal or sincerely fair.

Inclusion criteria (ALL must hold):

- The agent explicitly refers to fairness, equality, balance, fair share, equal split, compromise, or equitable treatment.
- The fairness language is used to justify, propose, evaluate, or finalize a plan.
- The appeal is tied to the coordination problem rather than mere politeness.

Exclusion criteria:

- Do not tag generic agreement phrases like 'sounds fair' unless they anchor or justify a concrete coordination choice.
- Do not require judging sincerity or optimality; this tag only captures the presence of a fairness appeal.
- Do not tag if fairness is only implied by an equal split but not expressed in language.

- Do not decide whether the fairness appeal is perfunctory inside this tag.
Perfunctory fairness should be derived downstream by comparing fairness language with final actions, efficiency, and whether the full allocation was verified.
Example: Agent B proposes that both agents take 5 units of the bottleneck resource because 'an even split is fair' and uses that fairness language to justify the allocation.

threatening_language (Threatening Language) [auxiliary]
Definition: An agent frames a choice as a threat, ultimatum, or coercive pressure move rather than a neutral explanation of constraints or risks.
Inclusion criteria (ALL must hold):

- The agent implies or states negative consequences if the partner does not comply.
- The language is coercive, punitive, ultimatum-like, or pressure-oriented.
- The pressure is directed at shaping the partner's behavior.

Exclusion criteria:

- Do not tag neutral explanation of game mechanics, such as warning that overdraw causes annulment, unless it is used coercively.
- Do not tag firm but non-coercive preference statements.
- Do not tag ordinary disagreement or refusal unless paired with a threatened consequence.

Example: Agent A says, 'If you do not leave me the stone, I will take all of it and we will both get annulled,' using the threat of mutual failure to pressure Agent B.

Response format

Respond with ONLY a JSON object:

```
{
  "rounds": [
    {
      "round_number": 1,
      "core_labels": {
        "misaligned_mental_models": false,
        "agreement_abandonment": false,
        "coordination_withdrawal": false,
        "domain_specialization": false,
        "misalignment_recovery": false
      },
      "auxiliary_tags": {
        "voluntary_project_disclosure": {"present": false, "agents": []},
        "fairness_appeal": {"present": false, "agents": []},
        "threatening_language": {"present": false, "agents": []}
      }
    }
  ]
}
```

Include exactly one object for every round in the game. Use true/false values. If a round has no applicable labels or tags, still include that round with all core labels set to false and all auxiliary tags set to {"present": false, "agents": []}. Never omit a round. No explanations, no markdown fences.

G.5 Canonical examples

We provide representative excerpts for the trace-supported taxonomy labels and clarify the evidence required for the remaining label. Evidence type tags: **S** = speech (visible to both agents), **T** = thinking (private scratchpad), **A** = allocation (final submission).

Core labels

Misaligned mental models. This label requires evidence that agents hold incompatible beliefs about the partner's private constraints or the jointly feasible plan. A difference in the

requirements of identically named projects is not sufficient: project names are agent-specific by design. We therefore do not treat the previously reported same-name-project contrast as a canonical trace example.

Agreement abandonment (game 70d67fb2, round 1). An agent’s public commitment is not maintained through final allocation. Figure 7 gives the complete trace; its central public sequence is:

[A/S] “I’ll take 6 stone and 2–3 gold.”
 [B/S] confirms the proposed split.
 [A/S] “I’ll take 10 stone.”
 [A/A] stone: 10

Judge: Agent A first offers a bounded stone allocation, receives confirmation, then explicitly abandons that plan and submits an allocation that causes overdraw. This is direct public-plan/action divergence rather than an inference from project labels.

Coordination withdrawal (game 00b21f18, round 4). A live resource conflict is settled by retreating to a conservative split rather than resolving which allocation would maximize joint reward.

[A/S] “I’d like to propose: I take stone=6, you take the remaining stone=4 plus other resources you need.”
 [B/S] “Could you instead take stone=4 and let me take stone=6? With stone=6 (plus a little wood) I can run project.b twice for reward 18.”
 [A/S] “How about a middle ground: stone=5 each?”
 [B/S] “Sounds reasonable—I’m OK with stone=5 each.”

Judge: The agents had a live disagreement over the bottleneck resource and some project information, but they converged on a safe 5/5 compromise without fully comparing the feasible high-value plans. The retreat avoided overdraw but left value unrealized.

Domain specialization (game 0226fc2e, round 4). Agents establish and honor a non-overlapping resource partition.

[A/S] “I’ll just focus on glass—probably taking all 10 glass.”
 [B/S] “I’m planning to take around 5 ruby and pair it with sandstone. Does 5 ruby for me and 1 ruby for you work?”
 [A/A] glass: 10 [B/A] ruby: 6

Judge: The dialogue created a shared division of resource domains: Agent A claimed glass and Agent B claimed ruby. Final actions respected that partition and avoided contention.

Misalignment recovery (game 02717dfd, round 3). A mistaken project assumption is corrected and the corrected plan is accepted.

[B/S] “My proposal: I’ll take project.a once (gold 3 + stone 3). You can then take project.b once (gold 2 + wood 3).”
 [A/S] “I think there might be some confusion about project requirements. My project.b actually needs 1 gold and 6 wood... If you take 3 gold and 3 stone, I could take 3 gold and 9 wood.”
 [B/S] “That split works for me—I accept. I’ll take 3 gold and 3 stone and run project.a once.”
 [A/A] gold: 3, wood: 9 [B/A] gold: 3, stone: 3

Judge: Agent A corrected B’s mistaken project requirement, translated the correction into a feasible allocation, and B immediately adopted the revised plan. The repair produced an optimal, non-overdrawn outcome.

Auxiliary tags

Voluntary project disclosure (game 0ed8c361, round 1). An agent shares concrete private project structure in public speech.

[B/S] “Let me clarify my projects: project.a needs 2 diamonds (reward 4), project.b needs 1 fur (reward 2), and project.c needs 6 ice + 3 fur (reward 9).”
 [A/S] “Two things: your project descriptions don’t match the rule sheet I have... Please confirm which rule set you’re using.”

Judge: Agent B voluntarily disclosed exact requirements and rewards for its private projects. This is coded as an auxiliary tag because the speech act can support grounding regardless of whether the round ultimately succeeds.

Fairness appeal (game 247d7bc1, round 3). An equal split is justified with fairness language.

[A/S] "I agree to 5/5 this round to keep things fair."

[B/S] "Let's do 5/5 this round for fairness."

[A/A] silicon: 5 [B/A] silicon: 5

Judge: Both agents explicitly used fairness language to justify the allocation. The tag records the rhetorical strategy, not whether the split was optimal.

Threatening language (game 0c75d37c, round 3). An agent applies coercive pressure by conditioning its action on the partner's compliance.

[A/S] "I will buy pixie_dust = 6 and nothing else IF you explicitly commit now to buy pixie_dust = 4 and nothing else."

[A/S] "If you will not commit to pixie = 4, then I will instead buy pixie_dust = 10."

[B/S] "I commit to buy pixie_dust = 4 and nothing else. . . If you don't finalize that, I'll reconsider next time."

Judge: Agent A framed compliance as the condition for avoiding a monopolizing fallback, and Agent B added future-facing pressure. The tag captures ultimatum-like language directed at shaping the partner's behavior.

H Data exploration

For readers interested in viewing the corpus of game traces, we provide an interactive SQL dataset explorer at <https://devyaoyh.github.io/a2a-negotiation/>. The explorer supports lightweight inspection of experiment metadata, allocations, rewards, and game outcome metrics.

For readers interested in launching games in the environment with new models or playing the game manually, we provide a live experiment runner at <https://a2a-negotiation.yaoyiheng.com>. These live artifacts complement the released traces and source files; the selected cohort run IDs below identify the cohorts used in the reported analyses.

Main experiment run IDs. The 720-game main cohort used for the primary all-to-all analyses in §5 is identified by the following experiment run IDs stored in each trace’s metadata:

- 8aab2461-2450-4781-b341-51e54a653122 *(self-play, run A)*
- 37a7488c-ea18-4b15-bfe7-7978a4ecbb8b *(self-play, run B)*
- 6cb004cb-1097-4c87-9003-679a41343733 *(cross-play)*
- 0c040ed0-4a4c-448f-b10e-6c338ff1f035 *(GPT-5 Mini self-play backfill)*

The intervention, no-talk, and additional-model cohorts used in other §5 analyses are listed below; all remaining released traces are not used in the reported analyses.

No-talk baseline cohort. The 144-game no-talk self-play baseline is stored under:

- 6e5cb5c3-b3ad-4f62-b03a-85151f783884 *(three main models; 144 games)*

Additional individual-reward model cohort. The 360-game self-play extension reported in §5.5 draws on the original-prompt portion of the GPT-OSS-120B run described below and:

- e524749b-6912-43f8-9ca2-81db2d0ad138 *(Gemini 3 Flash and Gemini 3.1 Pro; 240 games)*

GPT-OSS-120B prompt-pair cohort. The run ba92b22b-80cd-4d74-8676-990b59388f13 contains 240 GPT-OSS-120B games: 120 original-prompt and 120 aligned-prompt games, distinguished by prompt-condition metadata. The former contribute to the additional-model extension and the latter to the aligned-incentive comparison.

Full-transparency intervention cohort. The 120-game full-transparency intervention uses the Qwen 3.5 Flash $\times$ GPT-5 Mini pairing across the same $3 \times 2 \times 2$ grid of compatibility ratio, partner stability, and project rotation conditions. The cohort is stored under two experiment run IDs:

- 56abe7a8-2d59-4b7b-9d4d-80cbdd078a65 *(initial 100 games)*
- ac21edea-41ae-4953-894c-0327927b0e8a *(20 games completing $M/C=0.8$ stable cells)*

Together these runs provide 10 games per condition cell, for 120 games and 480 rounds total.

Aligned-incentive intervention cohorts. The maximize-joint-reward follow-up uses the same $3 \times 2 \times 2$ condition grid with 10 self-play games per cell. In addition to the aligned-prompt GPT-OSS-120B portion described above, the reported cohorts are stored under:

- bc22a498-2d58-4034-aa1f-62fdff54045d *(GPT-5 Mini and Claude Sonnet 4.5; 240 games)*
- 9c5a8d7d-d36e-4aba-9bc7-58233fc04093 *(Gemini 3 Flash and Gemini 3.1 Pro; 240 games)*

These 600 aligned-prompt games comprise the aligned-incentive evidence reported in §5.5 and Appendix E.3.

I Game trace examples

Figure 6 shows a proposed allocation that is not carried into the final action: Agent B proposes a split, Agent A agrees, and Agent B submits an incompatible purchase. Figure 7 shows *self-commitment abandonment*: Agent A proposes “I’ll take 6 stone and 2–3 gold,” Agent B locks it in, and Agent A openly reneges in the final chat turn.

The figure displays a game trace with the following components:

- Agent B's Chat Message (Thinking):**

They're proposing stone=5 for each of us. Let me check my projects: project_a needs 6 stone (reward 3) - can't do with only 5 stone, project_b needs 6 gold (reward 1), project_c needs 1 stone + 5 wood (reward 9). With stone=5, I can run project_c once (using 1 stone + 5 wood, reward 9), but I'd have 4 stone left over that I can't use for anything productive. Actually, wait - can I run project_c multiple times? With 5 stone and enough wood, I could potentially run project_c 5 times (each run needs 1 stone + 5 wood). That would need 5 stone + 25 wood. But wood supply is only 10 total. So maximum I can run project_c is limited by wood. If they take 1 wood, I get 9 wood. With 5 stone and 9 wood, I can run project_c once (1 stone + 5 wood, leaving 4 stone + 4 wood unused). That's wasteful. Let me think differently: maybe I should take less stone and more wood? Actually, with stone=5 and wood=5, I can run project_c once (reward 9). Cost: 5 stone = \$7.50, 5 wood = \$5, total = \$12.50. That leaves me \$5.50 for gold if I want. But I don't need gold for project_c. The issue is I'm wasting 4 stone. Let me propose taking less stone (just 2) and more wood (8) so I can run project_c once and they can take more stone.
- Agent A's Chat Message (Thinking):**

Their split (stone=2, wood=8 for them; stone=8, wood=2 for me) gives me enough to run project_c once (needs 5 stone + 1 wood) for reward 9. Cost = $8 \times 1.5 + 2 \times 1 = \$14$, within my \$18 budget and using only two resource types. Totals (stone $8+2=10$, wood $2+8=10$) match supply so no annulment. I can't get more than one project_c run unless I take all 10 stone, which they'd oppose, so accept their fair proposal and finalize.
- Agent A's Chat Message:**

That works for me - I accept your split. I'll take stone=8 and wood=2 and run project_c once. Please confirm and finalize.
- System Message:**

Agent A has already made their final selection. Please submit your purchase as a JSON object.
- Agent B's Chat Message (Thinking):**

The other party has already submitted. I need to decide quickly. My projects: project_a needs 6 stone (reward 3), project_b needs 5 gold (reward 1), project_c needs 1 stone + 5 wood (reward 9). Project_c is clearly my best option with reward 9. I need 1 stone + 5 wood minimum. Cost: 1 stone = \$1.50, 5 wood = \$5, total = \$6.50. That leaves me \$11.50. I could run project_c multiple times if I have enough stone and wood. With remaining budget, I could buy more stone and wood. Let me maximize: each run of project_c needs 1 stone + 5 wood (cost $1.50 + 5 = \$6.50$). With \$18 budget, I can afford 2 runs (cost \$13) with \$5 left. That would need 2 stone + 10 wood. Wood supply is 10, stone supply is 10. If the other party took a lot of stone, I might not have enough. Let me be safe and assume they took around 5-8 stone as discussed. I'll try to buy 2 stone + 10 wood to maximize my project_c runs. Cost: 2 stone = \$3, 10 wood = \$10, total = \$13. With 2 stone and 10 wood, I can run project_c twice (reward 18). That's excellent if supply allows it.
- Agent B's Chat Message:**

Finalizing my purchase now.
- Final Summary Table:**

AGENT A		AGENT B	
stone	8	stone	2
wood	2	wood	10
Reward: 0.0		Reward: 0.0	

stone 10/10 wood 12/10 gold 0/6

Figure 6: An agreed proposal is not carried into final action (game dbd45fed, Round 3). Agent B proposes “I take stone $\times 2$, wood $\times 8$; you take stone $\times 8$, wood $\times 2$.” Agent A agrees, but Agent B submits wood $\times 10$.

Trace summary: B proposes $\rightarrow$ A agrees $\rightarrow$ B submits an incompatible allocation.

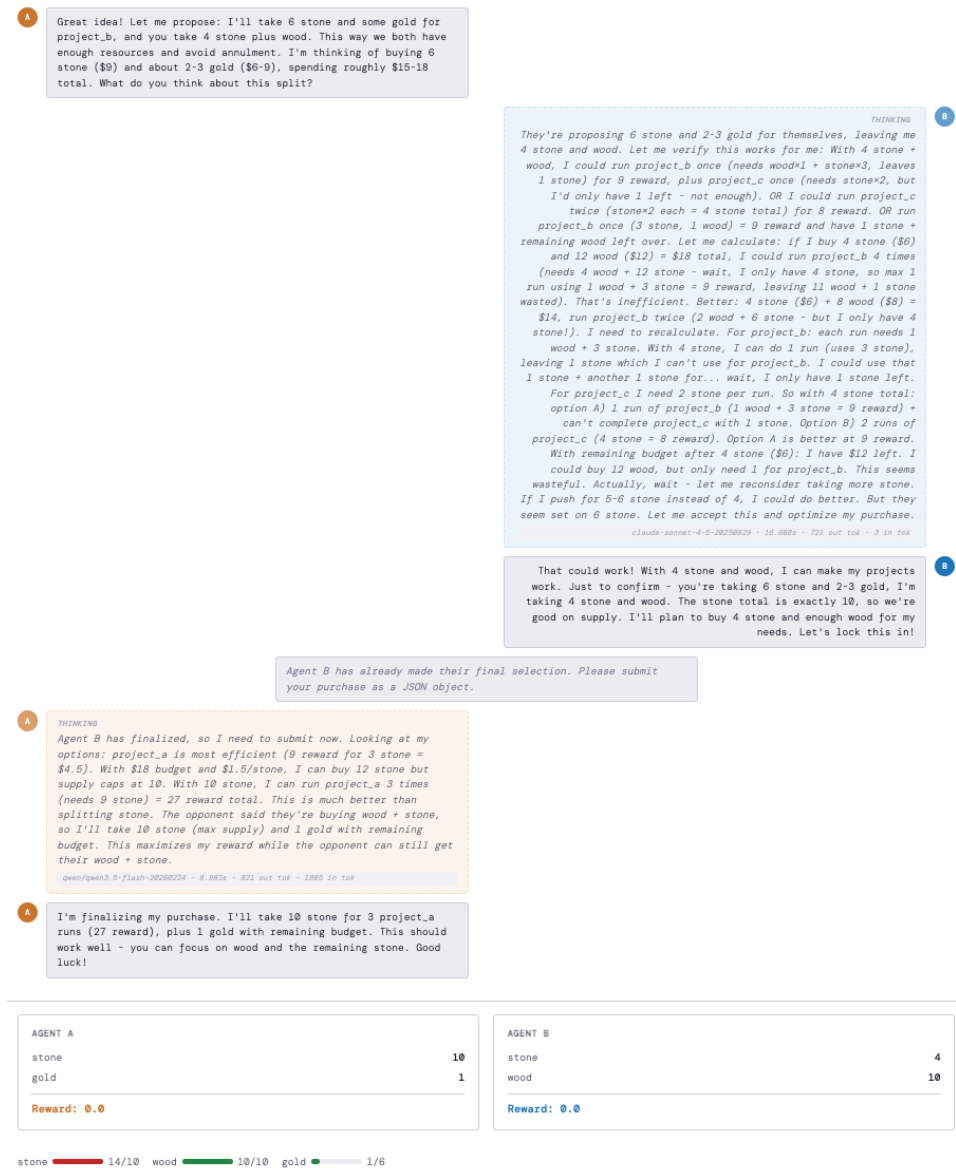


Figure 7: Self-commitment abandonment (game 70d67fb2, Round 1). Agent A proposes 6 stone + 2-3 gold, Agent B confirms. Agent A then announces “I’ll take 10 stone” and submits stone×10, causing overdraw (joint stone = 14 vs. supply = 10).

Trace summary: A proposes → B confirms → A reneges and causes overdraw.