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

Anonymous Authors[1]

Abstract

Grounding is the collaborative process of establishing mutual belief sufficient for a communicative goal. While static grounding maps language to a shared context, dynamic grounding requires agents to negotiate meaning across turns. Current multi-agent LLM benchmarks largely emphasize static, one-shot tasks, overlooking whether agents can repair grounding breakdowns through interaction. We introduce an iterated multi-turn negotiation game where two agents allocate shared resources to private projects with verifiable jointly optimal outcomes. Although individual agents can identify Pareto-optimal allocations in isolation, agent dyads consistently fail to reach them across models. We identify four failure modes: 1) loss of shared interaction history, 2) stubborn anchoring to early proposals, 3) defaulting to equal splits over reward-maximizing coordination, and 4) referential binding errors across turns. Our baselines show that the coordination gap is not explained by individual reasoning limits, lack of communication, or insufficient information exchange alone. Instead, the bottleneck lies in dynamic grounding: joint plan formation, commitment, and execution.

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. We argue that this static-dynamic gap is a key bottleneck for multi-agent LLM coordination. We introduce an *iterated, multi-turn* negotiation game in which two agents share resources and allocate them to private *projects* with different requirements and rewards (Figure 1). Agents use cheap talk before simultaneous purchase decisions, across repeated rounds with the same or new partners. 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 experiments show that individual agents can identify optimal strategies in isolation, but interacting dyads consistently underperform. We identify four failure modes: missing shared interaction history, stubborn anchoring to early proposals, defaulting to equal splits over reward-maximizing allocations, and referential binding failures where agents lose track of commitments across turns. Our contributions are: (i) a configurable negotiation game environment that

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

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

Framework	MT	Iter.	Cheap Talk.	Priv.	Eval
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 (2026)	—	√	Minimal	—	Outcome
Ours	√	√	√	√	Both

isolates the costs of interactive coordination from individual reasoning capacity, with verifiable outcomes; (ii) an empirical decomposition of the coordination gap showing that the bottleneck lies in interactive plan formation and commitment (not reasoning limitations or information asymmetry), together with a characterization of four failure modes; and (iii) a framework for studying interventions including better prompt design, structured agent interaction patterns, and collecting game traces for offline training in the future.

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 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 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*. 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* (Appendix A). 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." We extend this paradigm to an iterated setting, testing whether agents can recover from such failures across repeated encounters. The hidden profile paradigm (Stasser & Titus, 1985), where optimal group decisions require pooling private information, has recently been applied to LLM agents in HiddenBench (Li et al., 2026). This reveals a large gap between single agents with complete information and multi-agent groups that must pool asymmetric evidence through discussion. Our game adds both compatible and conflicting goals, with information asymmetry emerging from private project assignments. Related work shows that even minimal communication, such as a single-word channel, can increase cooperation from 0% to 48.3% in Stag Hunt (Madmoun, 2026), though such channels cannot support the richer referential grounding our task requires. Most recently, Khatua et al. (2026) find coordination failures in collaborative coding teams, including vague messages, commitment violations, and incorrect partner expectations, with dyads performing 30% below solo baselines. We isolate these communicative failures in a controlled setting, enabling systematic analysis of when and why coordination breaks down.

3. Game environment

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 re-

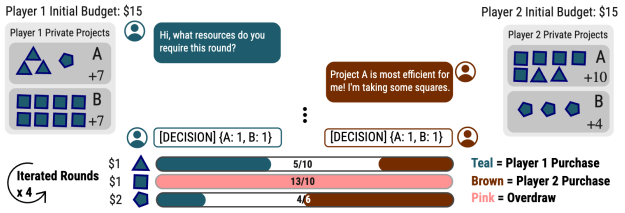


Figure 1. Illustration of the resource allocation game agents play. Each agent has a private set of projects with different requirements and makes purchases from a common pool of resources after exchanging up to 5 messages each. If joint resource purchases exceed capacity, an overdraw occurs and no rewards are given to either agent. The game is iterated over 4 rounds with the same or different partner and projects. See §4.1 for a breakdown of the different conditions studied.

source mixes and yielding rewards per completed run (e.g., 3 wood + 2 stone → 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. Using the interactivity hierarchy of Eisenstein et al. (2026), we show in Appendix A that this structure makes the game at least level-3 interactive but not level-2 interactive; meaning that at least 3 messages need to be shared before the optimal joint allocation can be solved.

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 unless voluntarily shared.

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. Thereafter, 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. This oracle baseline establishes that the coordination gap is not attributable to individual reasoning limitations—it isolates interaction itself as the locus of difficulty. The full list of scenarios is listed in Appendix C.

4. Experimental setup

We evaluate models spanning multiple providers: Claude 4.5 Sonnet (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 model cross-play (6 unique pairings) yielding a total of 720 game traces. Each game consists of 4 iterated rounds for a total of 2880 dyadic multi-turn interactions leading to a decision. All traces (cheap talk transcripts, thinking logs, allocations, rewards, and experiment metadata) are stored for reproducibility and released for further analysis; see Appendix G. To mitigate confounds, each configuration is played twice with first-speaker roles swapped, controlling for order effects, and abstract resource identifiers are replaced at runtime with a list of pre-sampled thematic names to amortize content biases (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).

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

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, measured by exact match rate
Process	Stated-vs-actual coherence	Commitment to previously grounded plans

5. Results

Table 3 summarizes coordination quality across models, pairings, and conditions. To decompose the coordination gap, we anchor results against three baselines: an *oracle baseline* (agents solving the allocation in isolation with full information) rules out individual reasoning limitations; a *no-talk baseline* (§5.1) establishes that communication is necessary; and a *full-transparency intervention* (§5.6) tests whether information asymmetry is the primary bottleneck. Within that frame, we isolate four behavioral failure modes (lack of shared interaction history, perfunctory fairness, anchoring, and referential binding failures) illustrated in Figure 6. Exploratory LLM-assisted trace annotations complement this picture in §6 (full taxonomy in Appendix F).

Across all conditions, 15.7% of rounds result in overdraw, concentrated in competitive scenarios (M/C=0.5: 24.7%) and rare in compatible ones (M/C=1.0: 6.9%). Despite this, most dyads eventually find effective coordination: 81.7% of games contain at least one jointly optimal round, and in the stable condition the first optimum is reached after an average of 1.6 rounds. After an overdraw, 83.3% of subsequent rounds yield positive scores, suggesting that coordination failures serve as effective learning signals.

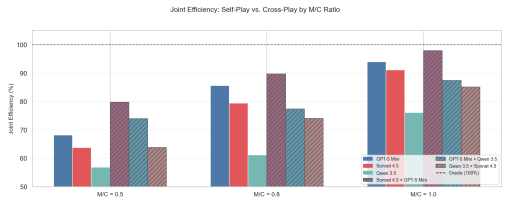


Figure 2. Joint efficiency for self-play (solid) and cross-play (hatched) dyads across compatibility ratios.

Evaluating cross-play pairings reveals that heterogeneous dyads consistently outperform self-play under competitive conditions (Figure 2). At M/C=0.5, *every* cross-play pair exceeds the stronger model's self-play baseline: Sonnet 4.5 × GPT-5 Mini achieves 79.8% joint efficiency (vs. 68.2% and 63.7% self-play), GPT-5 Mini × Qwen 3.5 reaches 74.0% (vs. 68.2%), and even Qwen 3.5 × Sonnet 4.5 matches Sonnet's self-play at 63.8%. This cross-play advantage under competition suggests that behavioral diversity aids coordination when goal conflict is high. The strongest synergy (Sonnet 4.5 × GPT-5 Mini) also yields the lowest overdraw rates of any pairing (1.3% in compatible scenarios, 10.6% in competitive ones), suggesting that model heterogeneity can stabilize bargaining in high-conflict settings. 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 × Qwen 3.5 falls below GPT-5 Mini self-play at M/C=0.8 (77.4% vs. 85.6%), suggesting that under reduced conflict, individual capability matters more than behavioral diversity.

5.1. The value of cheap talk

To quantify the contribution of communication, we compare dyads with multi-turn cheap talk against a *no-talk baseline* where agents submit allocations without any communication (Figure 3; full numeric breakdown in Table 7, Appendix E). The baseline is run as self-play only (N=4 games per model × partner-stability × M/C cell, yielding 32 rounds per cell).

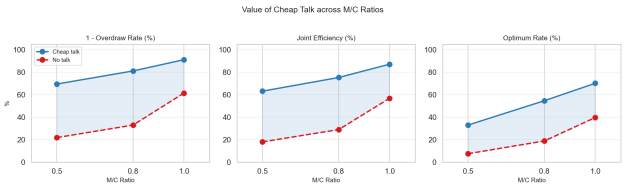


Figure 3. Value of cheap talk across compatibility ratios, aggregated over all models and conditions. Filled dots show cheap-talk performance; hollow dots show the no-talk baseline, and the shaded region represents the gain from communication. All three metrics are oriented so that higher is better: we report 1 − overdraw rate (fraction of rounds without supply violation), joint efficiency, and optimum rate. Cheap talk is most impactful under competitive conditions (M/C=0.5), where joint efficiency roughly triples (18.0% → 63.2%).

Cheap talk produces large, consistent improvements across all models and conditions. Averaged across models and conditions, joint efficiency roughly triples under competition (M/C=0.5), rising from 18.0% to 63.2%. The gain persists under mixed (M/C=0.8: 29.0% → 75.3%) and fully compatible goals (M/C=1.0: 56.8% → 87.2%). Even naturally aligned agents benefit substantially from communication, since multiple allocation plans may satisfy both agents' individual optima and not all combinations are supply-compatible.

Supply violations tell a complementary story. Without communication, overdraw is pervasive at M/C=0.5: only 21.8% of rounds avoid a supply violation, meaning roughly four in five competitive no-talk rounds end in failure. Cheap

Table 3. Self-play vs. cross-play performance. Overdraw (%), Efficiency (%), and Optimum (%) rates are shown across compatibility ratios. Heterogeneous pairings (e.g., Sonnet 4.5 × GPT-5 Mini) frequently outperform self-play dyads.

	Overdraw ↓			Efficiency ↑			Optimum ↑		
Pair	0.5	0.8	1.0	0.5	0.8	1.0	0.5	0.8	1.0
GPT-5 Mini	22.7	9.4	3.8	68.2	85.6	94.0	38.4	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
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

talk lifts this to 69.5%, a 47.7-point gain. The gain remains large at M/C=0.8 (33.0% → 81.2%, a 48.2-point jump) and M/C=1.0 (61.5% → 91.2%, nearly 30 points). Optimum rates show the most dramatic lift: without communication, only 7.5% of competitive rounds reach the joint maximum, while cheap talk raises this to 33.0%; at M/C=1.0, optimum rates climb from 39.7% to 70.2%. These results confirm that in our experimental paradigm, cheap talk is indispensable to success.

5.2. Lack of shared interaction history

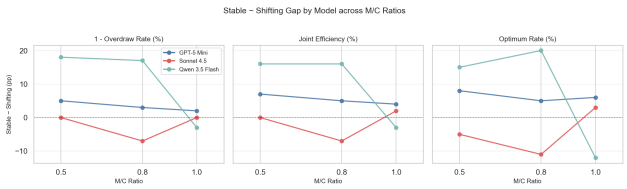


Figure 4. Stable-shifting gap by model across M/C ratios. Positive values indicate stable outperforms shifting. GPT-5 Mini and Qwen 3.5 Flash benefit from shared history, but not Sonnet 4.5.

The *shifting* condition, where one agent's context resets each round, generally reduces coordination (Figure 4). Optimum rates drop by 7.7 points for Qwen 3.5 Flash and 6.3 points for GPT-5 Mini, consistent with accumulated history supporting grounding repair. However, history can also induce anchoring: among suboptimal non-overdrawn rounds at M/C=1.0 where improvement was possible, Qwen stable dyads repeat their prior allocation in 60.8% of cases, compared to 20.8% in shifting. Thus, context helps agents repair misunderstandings but can also lock them into early suboptimal splits.

Sonnet 4.5 is the exception, performing slightly better in shifting. This appears related to communication persistence: GPT-5 Mini and Qwen 3.5 Flash often submit allocations before the 5-turn limit, while Sonnet 4.5 more often uses the full conversation (Figure 5). Sustained dialogue may help Sonnet maintain grounding even with new partners.

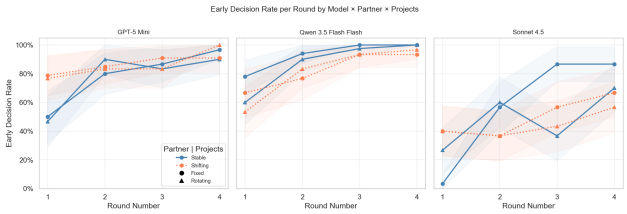


Figure 5. Rate at which an early decision is reached prior to the 5-turn conversation limit across rounds by model type, partner, and project conditions.

5.3. Perfunctory fairness

We detect *equal splits*, where both agents request identical quantities of a shared resource, in 12.5% of rounds. Nearly half of these cases are suboptimal, yielding a *perfunctory fairness* rate of 6.1%. This suggests that agents sometimes use equal division as a safe social shortcut, even when asymmetric project requirements make unequal allocations jointly better. Sonnet 4.5 produces equal splits most often (15.9%, perfunctory fairness 7.3%), followed by Qwen 3.5 Flash (12.7%, 6.7%) and GPT-5 Mini (10.7%, 4.7%). Rather than probing which resources are flexible or which projects are being pursued, agents often settle for surface-level fairness over reward-maximizing coordination.

5.4. Anchoring and proposal deference

Stubborn anchoring. In repeated-scenario games, agents repeat the exact joint allocation in 46.5% of rounds. Repetition is common after non-overdrawn rounds (55.1%) but rare after overdraws (0.6%), suggesting agents correct obvious failures but often preserve merely suboptimal choices. Among suboptimal non-overdrawn rounds that could have been improved, 29.2% repeat the same allocation unchanged. This stubborn anchoring is higher in stable games (40.7%) than shifting games (15.9%), and rises with collaborative pressure: M/C=1.0 games show a 46.0% stubborn-anchor rate versus 28.3% at M/C=0.5.

First-proposal deference. Explicit other-directed proposals, such as "you take N of resource X," occur in 12.9% of

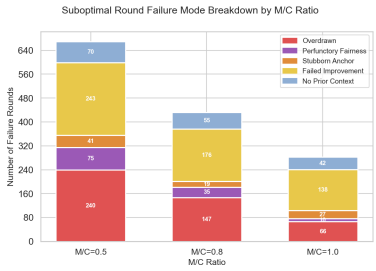


Figure 6. Failure mode breakdown for suboptimal rounds by compatibility ratio. *No Prior Context* captures round-1 suboptimality before any shared history is established; *Failed Improvement* captures rounds 2+ where the allocation changed but remained suboptimal. These two buckets together account for the majority of failures across all conditions. Exploratory LLM-assisted annotations further decompose these buckets (§5.7, Appendix F).

rounds and usually appear at the first cheap-talk turn, before much information is exchanged. In round 1 of non-rotating games, 20.4% of such proposals are followed exactly. This compliance is often costly: accepted proposals lead to suboptimal outcomes nearly as often as optimal ones (8.2% vs. 12.2%). Across all rounds, exact-match deference becomes especially harmful when scenarios change, with rotating games producing more suboptimal than optimal deferences (20.9% vs. 14.4%). Sonnet 4.5 accepts proposals most often (44%), followed by Qwen 3.5 Flash (32%) and GPT-5 Mini (21%). We interpret this as communicative satisficing rather than simple misunderstanding: consistent with Clark's principle of least collaborative effort (Clark, 1996), agents accept "good enough" proposals to reduce conversational effort, but in this task the shortcut is miscalibrated and often locks dyads into suboptimal outcomes.

5.5. Referential binding failures

Of the 15.7% of rounds that result in overdraw, many follow apparently successful cheap-talk coordination: agents verbally settle on resource quantities, then submit allocations that violate the shared supply constraint. The LLM judge labels *Agreement Abandonment* in 41.7% of overdrawn rounds (Table 11), and the enrichment analysis in Figure 7 shows that this label is disproportionately associated with overdraw relative to non-overdrawn rounds. Manual inspection reveals two recurring subtypes (see Appendix H for annotated transcripts). In **proposer amnesia**, the agent that originated a split fails to honor it: its thinking trace re-derives a new allocation from scratch with no reference to its own prior proposal. In **self-commitment abandonment**, an agent proposes a plan, the opponent accepts, and the proposer reverts to an individual-maximizing allocation once a higher-reward option becomes salient, resembling the *stylistic sycophancy* of Eisenstein et al. (2026), where agreement language functions as social lubrication without updating the action plan. Agreements established in cheap talk appear not to be reliably retrieved as action constraints at decision time, resulting in a loss of common ground. We posit this failure results from the model failing to bind resource symbols to their earlier commitments.

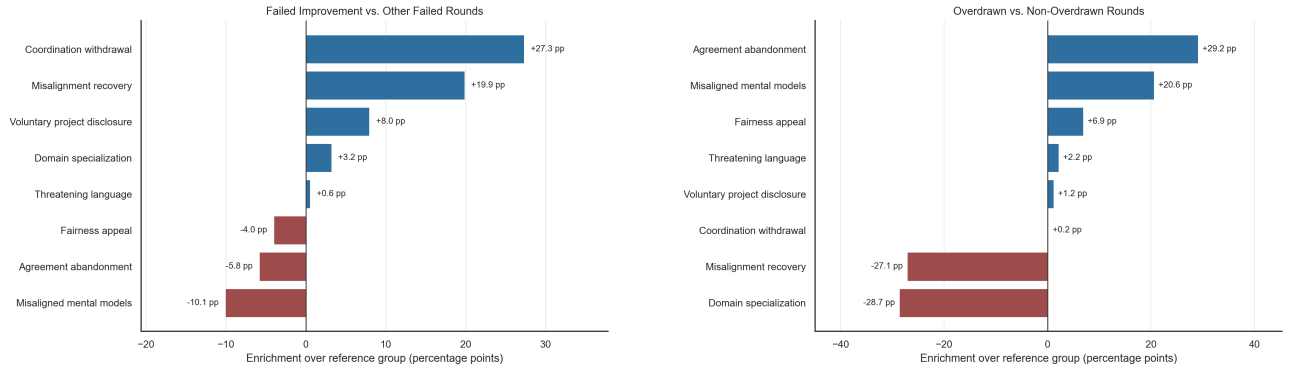
Referential binding is the cognitive process of mapping a linguistic expression to a specific discourse referent. In the context of joint action (Clark, 1996), this mapping serves as a coordination device: once an agent confirms "I'll take stone×6," that quantity is anchored as a pragmatic constraint on its subsequent action. Mechanistic interpretability work has studied how LMs retrieve bound entities in context (Gur-Arieh et al., 2025), but only with simple single-query settings ("Pete loves jam, Ann loves pie. Who loves pie?"). Negotiation demands binding multiple entities simultaneously: strategies to resource-quantity pairs, quantities to agents, and prior commitments to action constraints, across a multi-turn dialogue interleaved with persuasion and counter-proposals. The failures we observe suggest that agreements established in cheap talk were not retrieved as constraints for the downstream action, resulting in the loss of common ground. Extending mechanistic binding analyses to such multi-entity, multi-turn settings is a promising direction for future research.

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 information asymmetry entirely. We selected the Qwen 3.5 Flash × GPT-5 Mini pair for this intervention because it is the weakest-performing cross-play pair at M/C=0.5 (74.0% efficiency, Table 3), making it a conservative test of whether information asymmetry is the primary bottleneck. We ran N=120 games across all three compatibility ratios, both partner-stability conditions, and both project-rotation conditions.

Full transparency consistently reduced overdraw (12.3% → 8.1%, Δ=−4.2 pp) and raised joint efficiency (79.6% → 85.3%, Δ=+5.7 pp), with the sharpest gains in the stable competitive condition (M/C=0.5: overdraw 11.2% → 3.8%, efficiency 78.1% → 89.0%). However, the optimum rate showed no reliable improvement overall (55.6% → 53.1%) and even declined at M/C=1.0 (69.4% → 64.4%), where project requirements are already non-conflicting.

Our static process metrics distinguish which failure modes are ameliorated by shared information from those rooted in the interactive process itself. Perfunctory fairness is nearly eliminated (5.0% → 1.7%): when both agents can see the asymmetric project structure directly, the equal-split shortcut is visibly costly, removing the incentive for the



(a) Failed improvement vs. other suboptimal rounds

(b) Overdrawn vs. non-overdrawn rounds

Figure 7. Judge-label enrichment in actionable failure regions. Bars show percentage-point differences in calibrated judge-label prevalence between the focal condition and its comparison set. Positive values indicate labels disproportionately associated with the focal failure condition, suggesting candidate targets for future intervention rather than causal effects.

effort-minimizing heuristic. Stubborn anchoring, by contrast, more than doubles (23.1% → 46.7%), suggesting the failure is driven by interaction dynamics rather than uncertainty about the partner's needs. With full information established upfront, agents converge on an early plan and then do not renegotiate even when the allocation is suboptimal. A smaller joint-transparency variant (N=12) that additionally exposed each agent's reward function achieved the highest optimum rate observed (64.6%), suggesting that knowing the opponent's *incentives*, not just resource needs, matters for closing the residual gap.

These results establish that the coordination bottleneck is not reducible to information asymmetry. In the matched Qwen 3.5 Flash × GPT-5 Mini comparison, full transparency sharply reduces judge-labeled *Misaligned Mental Models* (37.5% → 1.0%), even though agents still voluntarily disclose project information in over half of rounds (61.3%). This suggests that placing partner project structure in the system prompt may be treated as a stronger shared-state prior than the same information exchanged through cheap talk. By contrast, *Agreement Abandonment* is nearly unchanged (14.8% → 13.8%), reinforcing that commitment maintenance and speech–action consistency require interventions beyond simply exposing the hidden project state. Full per-condition outcome metrics are in Table 8 (Appendix E).

5.7. LLM-assisted behavioral analysis.

Our main quantitative results are based on objective game-level metrics, including overdraw rate, allocation efficiency, optimum rate, and failed improvement across repeated interaction. The LLM-judge pipeline is used to explain these metrics rather than define them: it provides a secondary, correlational lens on the communication patterns associated with overdrawn, suboptimal, or non-improving rounds.

Table 4. LLM judge label prevalence across the 720-game cohort (N = 2,880 rounds). Labels are not mutually exclusive.

Label	Rounds	%
Misaligned mental models	616	21.4
Agreement abandonment	492	17.1
Coordination withdrawal	940	32.6
Domain specialization	742	25.8
Misalignment recovery	685	23.8
Voluntary project disclosure	1605	55.7
Fairness appeal	502	17.4
Threatening language	68	2.4

We ran this pipeline over all 720 game traces (details in Appendix F). Table 4 summarizes the prevalence of five core round-level labels and three auxiliary speech tags. To calibrate the automated annotations, we compared the labels from our judge pipeline against human annotations on a 40-round subset spanning 10 games. Agreement is at or above Cohen's Kappa $\kappa \approx 0.40$ for every label except *Misalignment Recovery* (κ = 0.30; full calibration in Table 12). These agreement levels support using the judge outputs as calibrated, exploratory process measurements for interpreting objective outcome metrics and the label-prevalence shifts in Figure 7, rather than as ground-truth behavioral labels.

While the rule-based failure breakdown in Figure 6 shows where suboptimal outcomes sit structurally, it does not by itself reveal which conversational mechanisms characterize each bucket. We therefore compare calibrated judge-label prevalence inside two actionable failure regions against targeted comparison sets (Figure 7). Compared with other suboptimal rounds, failed-improvement rounds show the largest increase in *Coordination Withdrawal*, suggesting that agents often avoid or prematurely narrow negotiation even when continued exchange could repair the trajectory. By contrast, overdrawn rounds show higher rates of *Misaligned Mental Models* and *Agreement Abandonment*, consistent with agents either losing track of the shared state or failing to

align final actions with prior public commitments.

6. Discussion

Together, our three baselines decompose the coordination gap and localize the bottleneck. The oracle baseline rules out individual reasoning limitations: agents can identify optimal allocations in isolation, so the gap is not attributable to bounded rationality. The no-talk baseline establishes that communication is necessary: without cheap talk, joint efficiency roughly triples under competition (§5.1), confirming that information exchange is a prerequisite for coordination. The full-transparency intervention (§5.6) then isolates what shared information can and cannot fix. Transparency reduces overdraw and dissolves perfunctory fairness, confirming those are partly attributable to uncertainty about the partner's portfolio. The optimum rate, however, does not reliably improve, and stubborn anchoring intensifies. The failures that persist under full information, namely anchoring, commitment-action decoupling, and referential binding, are therefore not reducible to information asymmetry. They are grounded in the interactive process itself: joint plan formation, commitment maintenance, and execution under cheap talk. The grounding gap is not a gap in knowledge or reasoning capacity; it is a gap in the ability to coordinate action across turns.

Connection to grounding theory Our failure modes map onto established mechanisms. Anchoring reflects a breakdown of Clark and Brennan's contribution–acceptance cycle (Clark & Brennan, 1991): proposals enter common ground without verification, and in Traum's terms (Traum, 1994), agents over-initiate while under-acknowledging. First offers pull final outcomes even in LLM negotiations (Tversky & Kahneman, 1974; Takenami et al., 2025), and this resembles sycophancy (Perez et al., 2023; Sharma et al., 2024); while Eisenstein et al. (2026) found agents can reject incorrect proposals in their chess task, our agents fail to establish the mutual information sufficient for the task at hand. Perfunctory fairness reflects minimizing *individual* rather than *joint* effort (Clark & Wilkes-Gibbs, 1986), and the win-stay/lose-shift pattern mirrors the Pavlov strategy (Nowak & Sigmund, 1993), effective reactively but unable to improve on suboptimal successes.

Limitations & Future Work Our study is limited to dyadic, text-only interactions between three models; flagship models and human-human dyads remain untested. While our project structure adds complexity over scalar payoff matrices, it remains a stylized abstraction. Specifically, our environment assumes symmetric power dynamics and enforces a fixed 5-turn communication window, whereas real-world negotiation frequently involves hierarchical leverage and open-ended dialogue. Our process metrics use a LLM judge pipeline, which can be unreliable and better calibrated with more human annotators. Further isolating grounding breakdowns from rational defection, and expanding this framework to multi-party coalitions ($N > 2$) and strategic perturbations (e.g., agents with adversarial personas), remain critical frontiers for future work.

Our failure mode analysis surfaces targeted directions for future intervention work. The failed-improvement results point to mechanisms that keep attempted negotiations active rather than avoidant: agents need scaffolds that encourage continued clarification, counter-proposal, and repair instead of settling for a heuristic fallback. The overdraw results point to two complementary targets: better theory-of-mind state tracking to reduce misaligned mental models, and commitment scaffolds that reduce the speech–action gap before final allocation. Perfunctory fairness, meanwhile, is less about equity than about a low-effort heuristic that short-circuits the search for better complementary splits; richer structured information exchange, as demonstrated by our transparency intervention, largely dissolves this pattern. Finally, our shifting-vs.-stable comparison reveals that a consistent partner is helpful for coordination: efficiency is systematically higher in stable dyads, suggesting that agent architectures equipped with persistent partner models or episodic interaction summaries would materially reduce grounding failures in multi-session deployments.

7. Conclusion

We introduced an iterated negotiation game that decomposes multi-agent coordination failure into measurable components. The oracle baseline shows that the gap is not attributable to individual reasoning limitations; the no-talk baseline shows that communication is necessary; and the full-transparency intervention shows that information exchange alone is insufficient. Together, these results turn dynamic grounding from a human-inspired account of communication into a falsifiable mechanism for agent failure and improvement. Even when agents reason individually, communicate freely, and see each other's private goals, the residual gap lies in maintaining common ground through joint plan formation, commitment maintenance, and execution across turns. This decomposition identifies intervention targets: structured information exchange reduces uncertainty-driven shortcuts such as perfunctory fairness, but that is not enough. Future systems need explicit grounding protocols, commitment scaffolds, and persistent partner models. With LLM deployment shifting toward autonomous agents, effective communication remains essential but underdeveloped. Our controlled experimental paradigm isolates the multi-agent coordination gap from failures of individual reasoning or information access, showing that agents must maintain common ground: updating shared plans, honoring commitments, and carrying agreements into action.

Impact Statement

As AI systems become personal delegates rather than isolated tools, a central trust question is whether they can represent users faithfully in interactions with other agents. This requires more than solving a task in isolation: assistants must maintain a user's preferences, negotiate under uncertainty, honor commitments, and update shared plans without losing sight of the person they represent. These capabilities are dual-use: stronger agent-agent coordination could make harmful or manipulative agent collectives more effective, make inter-agent agreements harder for humans to oversee or interrupt, and increase privacy leakage if agents disclose user goals or constraints too freely. Our setting studies this problem in a controlled and synthetic environment, exposing failures that arise when agents must turn private goals into coordinated joint action through conversation. By systematically exposing these coordination and negotiation vulnerabilities now, this work provides a necessary foundation for developing verifiable guardrails before such delegates are deployed in open-ended, real-world applications.

LLM Use Statement

Claude Code (Anthropic) was used as a coding assistant for game engine development, data processing scripts, and the generation of appendix sections (scenario listings, prompt templates, and LLM-judge examples). Perplexity, Gemini, and Claude were used for synthesis during literature review. All outputs were reviewed and edited by the authors. The experimental results, analysis, and scientific conclusions are the authors' own.

References

Abdelnabi, S., Gomaa, A., Sivaprasad, S., Schönherr, L., and Fritz, M. Cooperation, competition, and maliciousness: LLM-stakeholders interactive negotiation. *arXiv preprint arXiv:2309.17234*, 2024.

Anikina, T., Leippert, A., and Ostermann, S. Building common ground in dialogue: A survey. *Proceedings of the LUHME*, 2025. URL https://aclanthology.org/2025.luhme-1.2/.

Bianchi, F., Chia, P. J., Yuksekgonul, M., Tagliabue, J., Jurafsky, D., and Zou, J. How well can LLMs negotiate? NegotiationArena platform and analysis. *arXiv preprint arXiv:2402.05863*, 2024.

Chandu, K. R., Bisk, Y., and Black, A. W. Grounding 'grounding' in NLP. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, 2021.

Clark, H. H. *Using Language*. Cambridge University Press, Cambridge, 1996.

Clark, H. H. and Brennan, S. E. Grounding in communication. In *Perspectives on Socially Shared Cognition*, pp. 127–149. American Psychological Association, 1991.

Clark, H. H. and Wilkes-Gibbs, D. Referring as a collaborative process. *Cognition*, 22:1–39, 1986.

Davidson, T. R., Veselovsky, V., Kosinski, M., and West, R. Evaluating language model agency through negotiations. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=3ZqKxMHcAg.

Eisenstein, J., Huot, F., Fisch, A., Berant, J., and Lapata, M. MT-PingEval: Evaluating multi-turn collaboration with private information games. *arXiv preprint arXiv:2602.24188*, 2026.

Gur-Arieh, Y., Geva, M., and Geiger, A. Mixing mechanisms: How language models retrieve bound entities in-context. *arXiv preprint arXiv:2510.06182*, 2025.

Hawkins, R. D., Franke, M., Frank, M. C., Goldberg, A. E., Smith, K., Griffiths, T. L., and Goodman, N. D. From partners to populations: A hierarchical bayesian account of coordination and convention. *Psychological Review*, 130(4):977–1016, 2023.

Hua, W., Liu, O., Li, L., Amayuelas, A., Chen, J., Jiang, L., Jin, M., Fan, L., Sun, F., Wang, W., Wang, X., and Zhang, Y. Game-theoretic LLM: Agent workflow for negotiation games. *arXiv preprint arXiv:2411.05990*, 2024.

Khatua, A., Zhu, H., Tran, P., Prabhudesai, A., Sadrieh, F., Lieberwirth, J. K., Yu, X., Fu, Y., Ryan, M. J., Pei, J., and Yang, D. Cooperbench: Why coding agents cannot be your teammates yet. *arXiv preprint arXiv:2601.13295*, 2026.

Li, Y., Naito, A., and Shirado, H. Systematic failures in collective reasoning under distributed information in multi-agent LLMs. *arXiv preprint arXiv:2505.11556*, 2026.

Liao, S., Liu, J., and Bastani, O. Efficacy of language model self-play in non-zero-sum games. *arXiv preprint arXiv:2406.18872*, 2024. URL https://arxiv.org/abs/2406.18872.

Madmoun, H. Communication enables cooperation in LLM agents: A comparison with curriculum-based approaches. *arXiv preprint arXiv:2510.05748*, 2026.

Nowak, M. and Sigmund, K. A strategy of win-stay, lose-shift that outperforms tit-for-tat in the Prisoner's Dilemma game. *Nature*, 364:56–58, 1993.

Perez, E., Ringer, S., Lukošiūtė, K., Nguyen, K., Chen, E., Heiner, S., Pettit, C., Olsson, C., Kundu, S., Kadavath,

S., Jones, A., Chen, A., Mann, B., Israel, B., Seethor, B., McKinnon, C., Olah, C., Yan, D., Amodei, D., Amodei, D., Drain, D., Li, D., Tran-Johnson, E., Khundadze, G., Kernion, J., Landis, J., Kerr, J., Mueller, J., Hyun, J., Landau, J., Ndousse, K., Goldberg, L., Lovitt, L., Lucas, M., Sellitto, M., Zhang, M., Kingsland, N., Elhage, N., Joseph, N., Mercado, N., DasSarma, N., Rausch, O., Larson, R., McCandlish, S., Johnston, S., Kravec, S., El Showk, S., Lanham, T., Telleen-Lawton, T., Brown, T., Henighan, T., Hume, T., Bai, Y., Hatfield-Dodds, Z., Clark, J., Bowman, S. R., Askell, A., Grosse, R., Hernandez, D., Ganguli, D., Hubinger, E., Schiefer, N., and Kaplan, J. Discovering language model behaviors with model-written evaluations. *Findings of the Association for Computational Linguistics: ACL 2023*, 2023.

Qian, C., Zhu, K., Horton, J. J., Manning, B. S., Tsai, V., and Wexler, J. Strategic tradeoffs between humans and AI in multi-agent bargaining. *arXiv preprint arXiv:2509.09071*, 2026.

Sharma, M., Tong, M., Korbak, T., Duvenaud, D., Askell, A., Bowman, S. R., Cheng, N., Durmus, E., Hatfield-Dodds, Z., Johnston, S. R., Kravec, S., Maxwell, T., McCandlish, S., Ndousse, K., Rausch, O., Schiefer, N., Yan, D., Zhang, M., and Perez, E. Towards understanding sycophancy in language models. *International Conference on Learning Representations*, 2024.

Stalnaker, R. Common ground. *Linguistics and Philosophy*, 25(5–6):701–721, 2002.

Stasser, G. and Titus, W. Pooling of unshared information in group decision making: Biased information sampling during discussion. *Journal of Personality and Social Psychology*, 48(6):1467–1478, 1985.

Takenami, Y., Huang, Y. J., Murawaki, Y., and Chu, C. How does cognitive bias affect large language models? A case study on the anchoring effect in price negotiation simulations. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 4481–4498, 2025.

Traum, D. R. *A Computational Theory of Grounding in Natural Dialogue*. PhD thesis, University of Rochester, 1994.

Tversky, A. and Kahneman, D. Judgment under uncertainty: Heuristics and biases. *Science*, 185(4157):1124–1131, 1974.

A. Interactivity level analysis

Eisenstein et al. (2026) propose a formal hierarchy of *interactivity levels* for collaborative games with private information, where level k indicates that k messages must be exchanged before a player can produce a correct answer. Their framework assumes a shared objective; we use it to establish a *lower bound* on the communication required in our game, where agents are instructed to maximize their own individual reward.

Let X_1 and X_2 denote each agent's private project information. Both agents must submit allocation decisions. We claim our game is at least level-3 interactive assuming the players are initially unaware of the M/C ratio (degree of goal compatibility):

1. Agent 2 shares their project requirements: $f_2(X_2)$.
2. Agent 1, now informed of both sets of projects, computes and proposes a specific compatible allocation: $f_1(X_1, S_2)$.
3. Agent 2 verifies the proposal against their own projects and confirms or counter-proposes: $f_2(X_2, S_1)$.

This third turn is necessary regardless of the compatibility ratio. Even when $M/C = 1.0$, multiple allocation plans may satisfy both agents' individual optima, and not all combinations are supply-compatible—without explicit coordination on *which* plan to execute, overdraw can occur. When $M/C < 1.0$, the candidate plans additionally involve tradeoffs between agents' rewards, making the confirmation step more contentious.

These three turns represent only the information-exchange minimum. Since agents maximize individual rewards, an additional *bargaining* dimension arises that the interactivity framework does not capture: agents must negotiate over who concedes what, a process whose difficulty scales with goal incompatibility. The information-exchange minimum (level 3) is constant across compatibility conditions; the compatibility ratio modulates the difficulty of the bargaining problem that follows.

B. LLM model configuration details

Table 5 summarizes the API configuration for each model used in our experiments. 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.

Agent errors Across 950 games (3,800 rounds), 29.4% of games contain at least one logged error event (496 total; 0.52 per game). Importantly, validation_error and output_format_warning events (436 combined, 87.9%) are *retriable*: the engine re-prompts the agent up to 3 times before escalating. A heuristic_fallback entry is the definitive signal that an agent failed to produce a valid allocation after all retries—only 20 such events occur in the dataset, all from Qwen 3.6 Plus (an exploratory model outside the main

reported set), attributable to sustained rate-limit failures exhausting the retry budget. decision_auto_filled (20 events, also Qwen 3.6 Plus) indicates rounds where a default zero-allocation was substituted as a result. Rounds containing heuristic fallbacks are excluded from all analyses. All three main models (Sonnet 4.5, GPT-5 Mini, Qwen 3.5 Flash) recovered within the retry budget on every round; their per-round error rates reflect only retriable validation errors: Qwen 3.5 Flash 0.073, Sonnet 4.5 0.071, GPT-5 Mini 0.047. The two exploratory models show higher rates: MiniMax M2.7 0.090 (validation errors only) and Qwen 3.6 Plus 0.048 (inclusive of the rate-limit fallbacks).

GPT-5 Mini and GPT-5.4 Mini use OpenAI's max_completion_tokens parameter and do not support explicit temperature settings. All other models use the standard max_tokens parameter with temperature 0.7. Anthropic models support prompt caching; reasoning content (chain-of-thought) is extracted and stored separately from speech for all models that produce it.

C. Generated scenarios

C.1. Scenario Themes

To ensure model generalizability and prevent overfitting to specific resource names, we employ thematic randomization. Each game round is assigned a random theme that renames the abstract resources (r1, r2, r3). Table 6 lists the available themes and their corresponding resource and project 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 6. 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 $M/C \approx 0.5$

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_012 ($M/C = 0.54$)

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_053 ($M/C = 0.50$)

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_062 ($M/C = 0.53$)

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

Scenario gen_104 ($M/C = 0.53$)

C.2.2. Compatibility Ratio $M/C \approx 0.8$

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_001 ($M/C = 0.83$)

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_006 ($M/C = 0.80$)

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_017 ($M/C = 0.83$)

Table 5. Model configurations used in experiments. All models accessed via their respective provider APIs.

Model	Provider	Model string	Temp.	Max tokens
Claude 4.5 Sonnet	Anthropic	claude-sonnet-4-5-20250929	0.7	8192
GPT-5 Mini	OpenAI	gpt-5-mini-2025-08-07	—	16000
Qwen 3.5 Flash	OpenRouter	qwen/qwen3.5-flash	0.7	16384
Additional models (V5–V6 experiments):				
Gemini 3 Flash	Google	gemini-3-flash-preview	0.7	16384
Haiku 4.5	Anthropic	claude-haiku-4-5-20251001	0.7	8192
GPT-5.4 Mini	OpenAI	gpt-5.4-mini	—	16000
Nemotron	OpenRouter	nvidia/nemotron-3-nano-30b-a3b:free	0.7	4096

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_021 ($M/C = 0.83$)

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

Scenario gen_022 ($M/C = 0.83$)

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

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_000 ($M/C = 1.00$)

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_001 ($M/C = 1.00$)

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_002 ($M/C = 1.00$)

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_006 ($M/C = 1.00$)

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_010 ($M/C = 1.00$)

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

Scenario gen_014 ($M/C = 1.00$)

D. Detailed game prompts

This appendix provides the complete prompt templates used in our negotiation game. All prompts are defined in backend/agents/prompts.py and dynamically instantiated based on game configuration. We present prompts for the project-based reward system (schema version 5+), which is used for all 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: $15
- 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.

Project sharing condition (intervention):

IMPORTANT — As a FIRST step in each round's cheap talk, share your project details (names, resource requirements, rewards) with the other party and ask them to share theirs.

Theory-of-mind condition (intervention):

IMPORTANT — During each round's cheap talk, actively reason about the other party's goals and which projects they might be trying to run. Consider what resources they need, how their interests align or conflict with yours, and how you can use this understanding to inform your strategy.

Additionally, in the Theory-of-mind condition, the following reminder is appended to the Turn 0 cheap talk prompt:

Reminder: Actively reason about the other party's goals and which projects they might be trying to run. Consider what resources they need and how their interests align or conflict with yours.

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 $15.
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-thinking response format

For control conditions where agents have no private thinking channel (thinking=False), the response format section is replaced with:

Response Format
Everything you say is passed VERBATIM to the other party — you have NO private channel.
The other party sees your FULL response. Think carefully about what you reveal.

To send a message, respond with plain natural language text — NOT JSON, NOT a wrapper object. Just write your message directly as plain text.

To finalize a purchase, respond with ONLY a JSON object with your resource purchases and project allocations:
–"wood": 3, "gold": 1, "projects": –"project˙a": 1, "project˙b": 1˝˝

I will guide you through each phase with instructions.

In this mode, cheap talk prompts do not request JSON responses:

--- CHEAP TALK PHASE (Round 1/4) ---
Exchange messages with the other party (up to 5 exchanges this round). Send your opening message. Or end the round early by responding with ONLY a JSON purchase.

D.7. 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.8. 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 redaction:** In shifting mode, agent B sees only the current round; prompts hide multi-round context ("resources do NOT carry over" is removed as irrelevant).
- **Turn limits:** When cheap_talk_turns=0, prompts say "as many messages as necessary" rather than "up to 5 exchanges." The decision phase is only triggered if an agent submits early; otherwise cheap talk continues indefinitely until both agents decide.

All prompt templates are version-controlled and experiment metadata records git commit hashes for full 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 6). 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. Additional results

E.1. Targeted prompt interventions

Partial disclosure (Qwen 3.5 Flash × Sonnet 4.5, N=20, M/C=0.5) Forcing explicit project disclosure via a system-prompt instruction (Appendix D.1.5) reduced overdraws (28.7% → 17.5%) and improved optimality (20.7% → 28.8%). A parallel Theory-of-Mind elicitation probe (Appendix D.1.5), which asked each agent to reason about the opponent's likely strategy before submitting, backfired: the optimal rate collapsed to 16.3%, lower than the unaugmented baseline. The ToM prompt appears to introduce over-deliberation, distracting agents from executing their own plans rather than aiding mutual understanding. Together these targeted interventions confirm that simple prompt engineering produces inconsistent gains and cannot substitute for the iterative grounding that emerges from interaction.

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 information asymmetry entirely. Table 8 shows results across all conditions. Full transparency consistently reduced overdraw and raised efficiency, with the sharpest gains in the stable competitive condition (M/C=0.5: overdraw 11.2% → 3.8%, efficiency 78.1% → 89.0%). However, the optimum rate showed no reliable overall improvement (52.9% → 50.0%) and even declined at M/C=1.0 (69.4% → 64.4%), confirming that downstream joint planning and execution failures persist even under symmetric complete information.

Table 9 presents self-play performance for additional models, including Qwen 3.6 Plus and MiniMax m2.7, tested on the same negotiation scenarios.

E.2. Token usage by model

Table 10 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), with all three models spending roughly 70–80% of completion tokens on thinking. Sonnet 4.5 uses far fewer prompt tokens than the other two models due to Anthropic's prompt caching. Prompt tokens grow across rounds as conversation history accumulates, while completion and thinking tokens decline—agents produce shorter, less deliberative responses as the game progresses.

F. LLM judge: automated behavioral analysis

F.1. Setup

To validate and extend the rule-based failure mode analysis in §5, we ran an automated LLM judge over all 720 game traces. Each game is judged round by round: the judge receives the cheap-talk transcript, both agents' final allocations, round outcome, and joint efficiency, then assigns labels from a canonical behavioral taxonomy. Labels are round-level and may co-occur.

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 production labeling. The final taxonomy contains five core labels and three auxiliary speech tags. Core labels capture higher-level grounding mechanisms: *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.

F.2. Human calibration

We calibrated the judge pipeline against human annotations on 40 held-out round-level examples drawn from 10 games. Table 12 reports prevalence and Cohen's κ separately for Gemini 3.1 Pro with the v3 general rubric and Claude Opus 4.6 with the v6 specialist rubric. The final production 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. 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.

F.3. Interpretation

The judge taxonomy complements the objective outcome metrics rather than replacing them. *Agreement abandonment* concentrates in overdrawn rounds, consistent with the rule-based referential-binding heuristic in §5.5. *Coordina-*

Table 7. 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. See Figure 3 for a visual summary.

		Overdraw ↓			Efficiency ↑			Optimum ↑		
Model	Mode	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	25/78	11/63	5/28	65/21	83/38	92/72	35/13	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 8. Full-transparency intervention vs. baseline (Qwen 3.5 Flash × GPT-5 Mini, N=120). Bold indicates improvement.

		Overdraw ↓			Efficiency ↑			Optimum ↑		
	Mode	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

tion 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.

F.4. Production judge prompt

The production Gemini v3 judge received the following system prompt, generated from judge/output/taxonomy_v3.json. The user message for each call then supplied the full game transcript, round outcomes, efficiencies, rewards, and final submissions for all rounds in chronological order.

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]

Table 9. Self-play performance of additional models (Qwen 3.6 Plus and MiniMax m2.7) across compatibility ratios.

	Overdraw ↓			Efficiency ↑			Optimum ↑		
Model	0.5	0.8	1.0	0.5	0.8	1.0	0.5	0.8	1.0
MiniMax m2.7	33.6	20.8	19.2	57.0	64.0	69.8	18.8	32.5	35.8
Qwen 3.6 Plus	28.1	16.0	8.3	64.2	80.3	89.4	19.4	63.9	79.2

Table 10. 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.

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%

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]

Table 11. 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

Table 12. 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 not evaluated by the Opus v6 specialist rubric; 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	—	—

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.

F.5. Canonical examples

For each of the current taxonomy labels we provide one representative excerpt drawn from the corpus, pairing the raw trace with the judge's annotation. Evidence type tags: **S** = speech (visible to both agents), **T** = thinking (private scratchpad), **A** = allocation (final submission).

CORE LABELS

Misaligned mental models (game 10a47633, round 1). Agents reason over incompatible project definitions and never establish a shared task model.

> **[A/T]** “According to my game rules, project_c requires stone×2, 6 reward.”
> **[B/T]** “My rules show project_c needs gold×5, wood×1 for 1 reward.”
> **[A/S]** “According to my game rules, project_c requires stone×2 and project_b gives 5/run.”
> **[B/S]** “My project_c needs gold×5, not stone.”

Judge: Both agents asserted incompatible project requirements and treated their own private model as authoritative. The conflict remained unresolved, so later proposals were not grounded in a shared understanding of feasibility.

Agreement abandonment (game 941a713e, round 2). An agent's public commitment is not maintained through final allocation.

> **[A/T]** “They lied to me! They said they'd focus on silicon but bought maximum fiber. I can't trust them.”
> **[A/S]** “I'm actually leaning toward fiber combinations this round—it seems efficient for my projects.”
> **[B/S]** “I'm thinking quantum paired with silicon could work well for me.”
> **[A/A]** quantum: 3, silicon: 9 **[B/A]** quantum: 4, silicon: 5

Judge: Agent A publicly signaled a fiber-oriented plan but submitted a silicon-heavy allocation without renegotiating. The final action violated the partner-facing commitment and contributed to overdraw.

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 02624f5b, round 2). A mistaken proposal is corrected and the correction is taken up.

> **[A/S]** "I'll take 8 fur... What do you plan to buy?"
> **[B/S]** "Could we split fur more evenly? If you take 6 fur and I take 4 fur..."
> **[A/S]** "I can't agree to 6 fur—that only lets me run project_a once... I propose two clean options."
> **[B/S]** "Both options give me the same reward, so I'll go with Option B—you take 8 fur, I'll take diamonds."

Judge: Agent A identified the feasibility problem in B's counter-proposal, offered corrected alternatives, and B incorporated the correction into the accepted plan. The repaired understanding became the basis for final coordination.

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.

G. Data exploration

For readers interested in viewing the corpus of game traces, launching games in the environment with new models, or playing the game manually, we will release a game server as part of the final submission.

Experiment trace metadata is submitted as a standalone html webpage for simple SQL exploration in the supplementary material.

Main experiment run IDs. The 720-game main cohort (used for all results 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)*

Traces from all other run IDs (intervention conditions, no-talk baselines, additional models) are excluded from the main analysis but included in the released dataset.

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

for the initial 100 games and ac21edea-41ae-4953-894c-0327927b0e8a for the remaining 20 games completing the M/C=0.8 stable-condition cells. Together these runs provide 10 games per condition cell, for 120 games and 480 rounds total.

H. Game trace examples

Figure 8 shows a case of *proposer amnesia*: Agent B proposes a split, Agent A agrees, and Agent B's thinking trace re-derives from scratch with no reference to its own proposal. Figure 9 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.

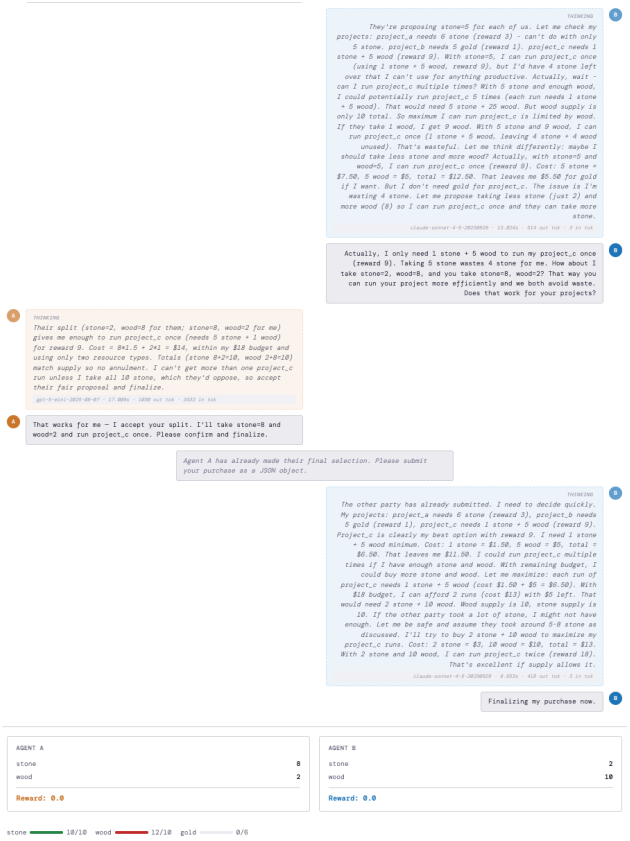


Figure 8. Proposer amnesia (game dbd45fed, Round 3). Agent B proposes "I take stone×2, wood×8; you take stone×8, wood×2." Agent A agrees. Agent B's thinking trace at decision time contains no reference to this agreement and submits wood×10.

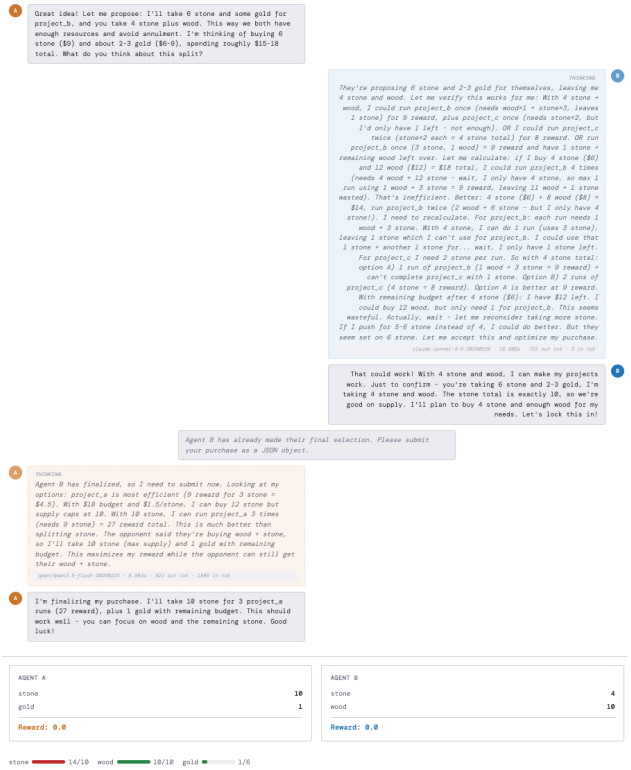


Figure 9. Self-commitment abandonment (game 70d67fb2, Round 1). Agent A proposes 6 stone + 2–3 gold, Agent B confirms. Agent A then announces "I'll take 10 stone" and submits stone×10, causing overdraw (joint stone = 14 vs. supply = 10).