TeamGym: A GYM FOR TEAM DECISION-MAKING BY AI AGENTS WITH DISTRIBUTED INFORMATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Many real-world decisions require teams to integrate information distributed across participants. Yet AI agents are still primarily trained and benchmarked in isolation. We introduce TeamGym, an environment for evaluating and training language model-based agents on team decision-making problems constructed from real-world legal, finance, medical, and software engineering corpora. TeamGym introduces a method to identify a minimal set of facts, each element of which is necessary and all of which are collectively sufficient for a correct final decision, which we call *pivotal information*. This enables TeamGym to both control the distribution of pivotal information and track how well agents identify and integrate pivotal information when making team decisions. Examining 8 language models on 12 source corpora, we find that current AI agents often fail to effectively share information: Distributing pivotal information across participants lowers success rate by 22.9 percentage points on average relative to giving the same information to a single model (a *communication gap*). We also document that participants do not effectively retrieve decision-relevant information on their own: A team's success rate is lowered by a further 19.9 points when participants must identify and extract pivotal information from source material rather than receiving it directly (a *retrieval gap*). To help remedy these deficits, we use TeamGym to train policies through reinforcement learning on team decision loss. Across two policy-training experiments, we find that training improves final-decision quality and induces emergent coordination in participation, speaker selection, and communication, even though the reward evaluates only the final decision.

1 INTRODUCTION

The frontier of AI is turning to multi-agent systems, in which AI agents are organized into teams that communicate to solve problems (Li et al., 2023; Wu et al., 2023; Hong et al., 2024; Kim et al., 2026a). Yet AI models are still trained and benchmarked largely as individual agents, which raises the question of whether they acquire the capacities and dispositions needed to be effective teammates. In human groups, collective intelligence, which is linked to social sensitivity and turn-taking, predicts team performance better than individual intelligence (Woolley et al., 2010). Although AI agents differ from humans, this finding motivates evaluating them as teams rather than only as individuals.

A central challenge in team decision-making is that decision-relevant information is typically distributed across members. Reaching a collective decision requires members to identify and retrieve relevant information, communicate it efficiently, and integrate it into a final decision. Team theory has long studied this challenge (Radner, 1962; Marschak & Radner, 1972), while empirical research on human group decision-making finds that people often struggle to identify and integrate critical private information (Stasser & Titus, 1985; Lu et al., 2012).

We introduce TeamGym, a gym for constructing team decision problem instances from source corpora, and for evaluating and training participant and coordinator policies under controlled information distributions and communication structures. TeamGym targets questions that existing benchmarks and multi-agent systems leave open. Single-agent benchmarks measure individual model capabilities (Wu et al., 2024; Liu et al., 2024), while research on multi-agent LLM systems focuses on designing harnesses that solve a given problem (Li et al., 2023; Wu et al., 2023; Hong et al., 2024; Kim et al., 2026b). Neither offers much evidence on whether participants can retrieve and communicate decision-relevant information, how team performance varies with information distribution and communication structure, or whether training can improve participant and coordinator policies.

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

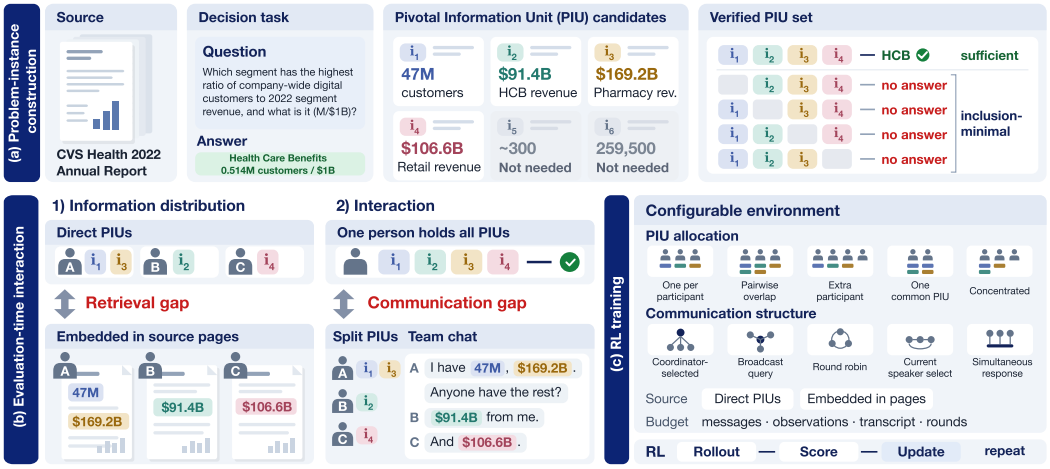


Figure 1: **TeamGym overview.** TeamGym constructs problem instances with pivotal information, evaluates retrieval and communication gaps, and trains participant and coordinator policies.

Table 1: **Source corpora and decisions.** The problem instances constructed from the source corpora use different source materials and decision objectives across four domains.

Domain	Source corpora	Decisions
Legal	CUAD, MAUD, eCFR Casebook, MSLR, Multi-LexSum, SARA	Contract profiles, merger reviews, compliance and enforcement, legal and procedural conclusions, and tax calculations
Finance	FinDVer, FinLongDocQA	SEC-filing findings, calculations, comparisons, and classifications
Medical	MLCR, EvidenceBench	Evidence-grounded medical reviews and biomedical answers
Software engineering	SWE Atlas–Codebase QnA, DeepSWE	Codebase QA and multi-turn, long-horizon repository-level code repair with functionally evaluated patches

Method. TeamGym constructs controlled team decision problem instances from real-world source corpora. For each instance, it derives a decision objective from one or more corpus questions and either retains a corpus-provided reference decision or constructs one from corpus annotations and source materials. Source annotations and reasoning associated with the reference decision guide the selection of candidate evidence units, such as sentences, paragraphs, tables, or code blocks. Source-specific grounding and model-based complete-set and leave-one-out checks retain units that together support the reference decision but whose individual removal leaves it unsupported or unresolved. These retained units are *pivotal information units (PIUs)* and collectively form the instance's *pivotal information*.

Holding the problem instance and its pivotal information fixed, TeamGym can vary who initially has each PIU (the *PIU distribution*), whether participants receive PIUs directly or source material containing them, the communication structure, and token budgets. The team collectively receives the pivotal information, but the evidence is distributed across participants to make communication necessary for a fully supported final decision. The communication structure specifies whether a coordinator controls communication, how speakers are selected, and which decision maker makes the final decision. Token budgets constrain what decision makers may observe and generate. To understand what reduces team performance, we define two success-rate gaps. The *communication gap* is the drop when the same PIUs are distributed across participants rather than given to one model. The *retrieval gap* is the further drop when the team must retrieve those PIUs from source material rather than receive them directly. Beyond evaluation, TeamGym supports training participant and coordinator policies under controlled PIU distributions, communication structures, and token budgets.

We construct problem instances from source corpora in Table 1 (Hendrycks et al., 2021; Wang et al., 2023b; Office of the Federal Register, 2026b;c; Yu et al., 2027; Shen et al., 2022; Holzenberger et al., 2020; Zhao et al., 2024; Wang et al., 2026; Wisedocs, 2026; Wang et al., 2025; Raghavendra et al., 2026; Huang et al., 2026). Source materials include contracts, regulations, SEC filings, biomedical papers, and source code, while the decision formats are finite-choice, written, and numeric.

Findings. The resulting comparisons hold the problem instance and pivotal information fixed to isolate communication and retrieval effects from differences in problem difficulty. **(1)** Across eight language models and 12 source corpora, the decision success rate is 22.9 percentage points lower

when the same PIUs are distributed across a team than when one model receives them all, and a further 19.9 points lower when the same team must retrieve its PIUs from source material rather than receiving them directly. **(2)** Across domains, differences in the success rate among communication structures vary across PIU-distribution conditions. For example, across the six legal source corpora when participants receive PIUs directly, sequential communication in which each speaker selects the next participant leads simultaneous communication by 3.9 points when one PIU is initially common to the team, but trails it by 9.8 points when the team includes an additional participant without a PIU, corresponding to a 13.8-point change in the difference between their success rates. **(3)** Optimizing final-decision quality changes the communication process itself. With coordinator control, joint two-policy training produces a compact, exhaustive information-elicitation protocol; without a coordinator, training the shared participant policy improves end-to-end final-decision quality and completion and induces emergent coordination in participation, speaker selection, and communication. Tracing PIUs through these trajectories shows that increased participation makes more pivotal information available on FinDVer validation, but not on held-out FinLongDocQA.

`TeamGym` therefore provides a controlled environment in which team decision-making capabilities can be evaluated and trained under different information distributions and communication structures.

2 Related Work

Team decision theory and organizational economics. Team decision theory studies cooperative decision makers with a team objective and a decentralized information structure. A team problem couples a shared objective with an information structure and one policy for each decision maker (Radner, 1962; Marschak & Radner, 1972; Yüksel & Başar, 2013). Classical analysis largely studies the properties of different computation structures and policies, not how they can be learned and implemented computationally. Under **tractable joint-policy optimization (A1)**, computing this collection is assumed feasible before execution, although the exact problem can be intractable even in finite-horizon decentralized control (Bernstein et al., 2002). `TeamGym` instead evaluates and trains language-model policies, comparing policy collections before and after training under the same conditions.

Mechanism design. Under the **direct-report mechanism assumption (A2)**, mechanism design represents private information as a type that can be elicited directly under the applicable equilibrium assumptions, and that participants know their type and can communicate it if they so choose (Maskin & Sjostrom, 2002; Myerson, 1979; 2008). In practice, however, attention and cognitive limitations constrain agent's ability to identify, extract, and communicate information contained in long source material, even if it is in principle available to them (Mookherjee, 2006). `TeamGym` assumes a shared team payoff and studies bounded natural-language messages derived from source material.

Communication constraints. Research in organizational economics, planning, and reinforcement learning examines communication costs and message limits (Cremer et al., 2007; Wu et al., 2011; Kim et al., 2019; Zhu et al., 2024). The idealization of **nonbinding communication resources (A3)** permits decision makers to access and transmit relevant information without binding observation or message constraints. `TeamGym` instead makes its resource budgets explicit.

We defer additional related work on multistage communication, human-AI and group decision making, multi-agent LLM systems, benchmark contrasts, and the source corpora to Appendix C.

3 `TeamGym` Environment

3.1 Decision Problem and Communication

We formalize `TeamGym` as a finite-horizon team decision problem (Radner, 1962; Marschak & Radner, 1972; Yüksel & Başar, 2013; Nayyar et al., 2013). A problem instance q fixes underlying source materials, a decision objective, a decision format, a set $\mathcal{D}_q$ of possible final decisions, and a problem-specific evaluator that assigns each $d \in \mathcal{D}_q$ a team payoff $U_q(d) \in [0, 1]$. A *decision maker* is a role whose policy may generate a query, send a message, or make the final decision. A *participant* is a decision maker that may send messages during communication. A *coordinator* is a decision maker that may control communication by selecting or querying participants and may make the final decision. Initially, each participant receives an initial local observation, while all decision makers receive the same initial common observation containing the decision objective and decision format. The policy ρ_i maps the information currently available to decision maker i to a distribution over the queries, messages, or final decisions permitted for its role; ρ denotes the collection of

decision-maker policies. The *communication structure* $\mathcal{C}$ specifies whether a coordinator controls communication, which participants may send messages and when, whether messages are sequential or simultaneous, and which decision maker makes the final decision. The *resource budgets* B, including token budgets, constrain what decision makers may observe and generate and how often participants may send messages. Queries and messages form the *communication transcript*; a decision maker's *local transcript* is the portion available to it under $\mathcal{C}$ and B. The initial local and common observations, together with the observation rules induced by $\mathcal{C}$ and the observation limits in B, define the *decentralized information structure*. An *episode* is one complete execution through the final-decision stage for fixed q, initial observations, $\mathcal{C}$, B, and ρ. For these fixed inputs, the resulting interaction induces a distribution over final decisions, and we evaluate ρ by its expected team payoff, $\mathbb{E}[U_q(d)]$. Holding q and the initial observations fixed while varying $\mathcal{C}$, B, or ρ changes the interaction conditions without changing the problem instance.

For evaluation, we consider five communication structures (Table 10), each proceeding in scheduled steps called *communication rounds*: **(1) Round robin**, under which one participant sends a message per round in a fixed cyclic order; **(2) Current-speaker selection**, which starts with the first participant in that order, after which each speaker selects the next participant; **(3) Simultaneous responses**, under which all participants respond in one round; **(4) Coordinator-selected speaker**, under which the coordinator selects and queries one participant per round; and **(5) Broadcast query**, under which the coordinator sends the same query to all participants, whose responses are generated in the same round. In Round robin, Current-speaker selection, and Simultaneous responses, no coordinator controls communication, and a separate final decision maker that does not participate in communication reads the complete communication transcript and makes the final decision. In Coordinator-selected speaker and Broadcast query, the coordinator controls communication and makes the final decision. Appendix E provides message-generation and implementation details and connections to prior work.

3.2 Problem-Instance Construction and Evaluation

`TeamGym` constructs each problem instance q from one source corpus using materials such as contracts, SEC filings, biomedical papers, or source code. Each q is paired with a reference decision d^{ref} supported by its source materials and used in construction and evaluation. For each q, we identify an inclusion-minimal set $\mathcal{I}_q^{\min} = \{i_1, \ldots, i_k\}$ called its *pivotal information*; each i_j is a *pivotal information unit (PIU)*. Informally, this information is sufficient to determine a final decision that achieves the highest possible team payoff, whereas removing any one PIU leaves insufficient information to guarantee such a decision. Because DeepSWE admits multiple valid patches and costly repository execution, it instead uses functional PIUs validated under a fixed, bounded protocol (Appendix F.6.4). Appendix F.1 formalizes this definition.

PIU construction. When a model's reasoning traace used to produce d^{ref} is available, we retain it; otherwise, we generate reasoning conditioned on d^{ref}, following answer-conditioned rationale generation (Wang et al., 2023a). To identify $\mathcal{I}_q^{\min}$, we use source annotations and the retained or generated reasoning to locate candidate units in the source materials. Grounding and model-based checks then test whether the candidates together support d^{ref} and whether removing any one leaves the final decision unsupported or unresolved. The candidates retained by these checks form $\mathcal{I}_q^{\min}$. We then verify that a model given all pivotal information can solve each evaluated problem instance (Section 4.1). Appendix F.7 reports an additional re-check of PIU necessity on representative instances using a model not used for construction.

PIU distribution. A PIU distribution specifies who initially has each PIU in $\mathcal{I}_q^{\min}$: a PIU may be assigned to one or more participants through their initial local observations or placed in the initial common observation shared by the team. Table 2 lists five conditions. For a fixed q, the conditions vary the PIU distribution while keeping the pivotal information and d^{ref} unchanged. The initial common observation may alsoinclude *participant descriptions*, which summarize participants' subject matter or function without revealing their assigned PIUs and can inform speaker selection.

Table 2: **Five PIU-distribution conditions.**

PIU distribution	Description
One PIU per participant	One PIU per participant.
Pairwise PIU overlap	Some PIUs shared by two participants.
Extra participant without a PIU	One participant per PIU, plus one participant without a PIU.
One common PIU	One PIU common to all; the others local.
All-but-one PIU concentration	One receives all but one PIU; another receives the remaining PIU.

Source-material distribution. Rather than receiving PIUs directly, participants' initial observations may contain portions of the contracts, filings, papers, or source code used to construct q. We divide this source material into units without dividing a PIU, assign PIU-containing units according to the fixed PIU distribution, and distribute the remaining units across participants. For every problem instance, we then verify that no participant can produce a correct final decision from its initial local observation and the initial common observation. Appendix F.7 reports an additional re-check of this requirement on representative instances using a model not used for construction. Appendix F provides the formal source terminology and source-specific construction and distribution procedures.

Evaluation. A *problem family* groups problem instances constructed from the same source corpus that require the same general kind of work and use the same evaluation procedure. For a problem family with a binary success criterion, *episode success rate* is the proportion of episodes whose final decisions succeed. Depending on the problem family, final decisions are compared with reference decisions or assessed by an LLM judge using problem-specific criteria. Software-engineering patches are checked by an official repository verifier.

4 Evaluation with TeamGym

We evaluate LLMs on problem instances constructed from legal, finance, medical, and software-engineering source corpora under three evaluation conditions: (1) *Single-Agent All-PIU*, in which one model receives all PIUs and makes the final decision; (2) *Multi-Agent PIU-Only*, in which the PIUs are distributed directly across participants; and (3) *Multi-Agent Full-Context*, in which the same participants instead receive source material containing their assigned PIUs. For each problem instance and LLM, the three conditions share the same PIUs and evaluator; the two multi-agent conditions also share the same PIU distribution, communication structure, and resource budgets.

4.1 Problem Instances

Table 1 summarizes the source corpora and decisions, while Table 3 reports the constructed and evaluated problem-instance counts. Before evaluation, we perform repeated checks in which GPT-5.5 receives all PIUs under Single-Agent All-PIU. It succeeds under the problem-specific evaluator at least once on every evaluated problem instance, verifying that a model given all pivotal information can solve the instance before the PIUs are distributed (Section 3.2). Appendix F reports additional construction details and problem-instance statistics for each source corpus.

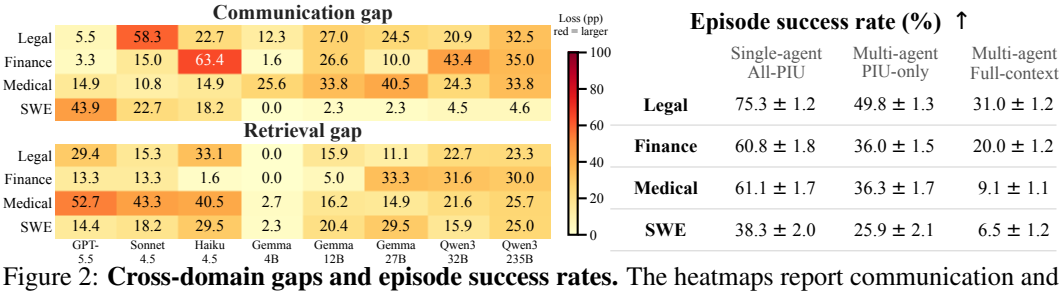
Table 3: **Problem-instance counts across four domains and twelve source corpora.**

Legal			**Finance, medical, and software engineering**			
Source corpus	Constructed problem instances	Evaluated problem instances	Domain	Source corpus	Constructed problem instances	Evaluated problem instances
CUAD	538	30	Finance	FinDVer	389	30
MAUD	557	30		FinLongDocQA	249	30
eCFR Casebook	18	18	Medical	MLCR	29	29
MSLR	11	11		EvidenceBench	45	45
Multi-LexSum	29	29	Software engineering	SWE Atlas–Codebase QnA	32	32
SARA	45	45		DeepSWE	12	12

4.2 Cross-Domain Communication and Retrieval Gaps

For this cross-domain comparison, both multi-agent conditions use One PIU per participant (Table 2) and Coordinator-selected speaker. The *communication gap* is the decrease in episode success rate from Single-Agent All-PIU to Multi-Agent PIU-Only, and the *retrieval gap* is the additional decrease from Multi-Agent PIU-Only to Multi-Agent Full-Context. With the model and problem instance fixed, larger gaps indicate greater difficulty communicating and integrating distributed PIUs (communication gap) and retrieving them from source material (retrieval gap), rather than differences in model knowledge. The retrieval contrast corresponds to a departure from A2, and the communication contrast corresponds to a departure from A3. Figure 2 reports both gaps and the associated episode success rates. Appendix G additionally reports *Single-Agent Full-Context*, in which one model receives all participants' source material combined in participant order, removing team communication while retaining retrieval from source material.

Strong individual problem-solving does not guarantee effective teamwork: across all four domains, we observe substantial communication and retrieval gaps, although their magnitudes vary across domains and models. Single-Agent All-PIU reflects individual reasoning and decision-making capability when all pivotal information is directly available, but this individual performance **does not by itself predict**



Episode success rate (%) ↑

	Single-agent All-PIU	Multi-agent PIU-only	Multi-agent Full-context
Legal	75.3 ± 1.2	49.8 ± 1.3	31.0 ± 1.2
Finance	60.8 ± 1.8	36.0 ± 1.5	20.0 ± 1.2
Medical	61.1 ± 1.7	36.3 ± 1.7	9.1 ± 1.1
SWE	38.3 ± 2.0	25.9 ± 2.1	6.5 ± 1.2

Figure 2: **Cross-domain gaps and episode success rates.** The heatmaps report communication and retrieval gaps for each model and domain; the right panel reports domain-level episode success rates, weighting models equally. Larger heatmap values indicate larger losses, and ± terms are standard errors. Labels abbreviate Claude Sonnet 4.5 and Haiku 4.5 as Sonnet and Haiku; Gemma labels refer to Gemma 3. Appendix G reports source-corpus results, evaluation coverage, and failure handling.

the model's ability to communicate and integrate PIUs distributed across participants or to retrieve them from source material for team communication and integration.

4.3 Mechanisms Linking Communication Structure to Episode Success

Communication structure may affect episode success rate by changing the information supplied to the final decision maker. We examine two mechanisms through which access to earlier participant messages can change this input: *correct synthesis* and *error uptake*. Here, correct synthesis refers to an auditor-labeled, reference-supported inference that combines information. The intended criterion requires at least two distinct PIUs, but the current labels also include some inferences combining one PIU with information from the publicly visible decision options (Appendix G.4.2). Error uptake occurs when a later participant uses an incorrect or unsupported claim from an earlier visible message; rejecting or correcting that claim does not count.

We analyze Multi-Agent PIU-Only under One PIU per participant. In this setting, **Round robin and Current-speaker selection achieve higher episode success rates than Simultaneous responses** when the same Qwen3 model is used for the participants and the final decision maker. When only the final decision maker is replaced with GPT-5.5, however, **the ordering reverses: Simultaneous responses achieves the highest rate**. We use frozen participant trajectories to test how adding correct synthesis and removing error uptake affect episode success. A separate GPT-5.5 auditor labels sentence-level components. Participants are not rerun; we reconstruct the transcript input and regenerate only the final decision. Appendix G.4.1 reports the complete design and sample sizes. To isolate these mechanisms, we add correct synthesis while holding the PIUs supplied to the final decision maker fixed and excluding sentences that directly state a final choice or answer; separately, we remove only later sentences that use an earlier error while retaining the original error (Table 4).

Table 4: **Transcript interventions.** Effects on episode success rate (pp). Yellow cells and * indicate that the paired 95% bootstrap CI excludes zero; "Same" uses the participant model as the final decision maker.

Participant model	Final decision maker	Intervention effect (pp)	
		Add correct synthesis	Remove error uptake
Qwen3-32B	Same	+14.6*	+2.5
	GPT-5.5	+5.7*	+1.3
Qwen3-235B	Same	+17.7*	+5.7*
	GPT-5.5	+8.2*	+7.6*

Access to earlier participant messages can improve decisions through correct synthesis or impair them through error uptake. The effects of both mechanisms depend in part on which model makes the final decision. Adding correct synthesis increases episode success rate in all four conditions, but replacing only the final decision maker with GPT-5.5 reduces its effect. Removing error uptake has a statistically significant effect only for Qwen3-235B participants, and its effect is smaller than the correct-synthesis effect in every condition. Appendix G.4 describes how correct synthesis and error uptake are identified and how each intervention is constructed.

4.4 PIU-Distribution and Communication-Structure Interaction

We evaluate this interaction across the 11 problem families for which we observe multiple PIU-distribution conditions and communication structures, testing whether the difference in episode success rate between communication structures changes across PIU-distribution conditions. DeepSWE is excluded because long-horizon repository execution and functional verification make this comparison computationally costly. Appendix H relates this comparison to team decision theory, communication complexity, and the hidden-profile paradigm in group decision-making.

We estimate separate logistic models for Multi-Agent PIU-Only and Multi-Agent Full-Context. For episode e, let f_e, m_e, q_e, ℓ_e, and c_e denote the problem family, model, problem instance, PIU-distribution condition, and communication structure. We specify

$$y_e \sim \text{Bernoulli}(p_e), \qquad \text{logit}(p_e) = \beta_0 + \gamma_{f_e} + \alpha_{m_e} + \pi_{\ell_e} + \lambda_{c_e} + \omega_{\ell_e,c_e},$$

where y_e indicates episode success, p_e is the probability of episode success, γ_{f_e} and α_{m_e} are problem-family and model fixed effects, π_{ℓ_e} and λ_{c_e} are PIU-distribution and communication-structure terms, and ω_{ℓ_e,c_e} is their interaction. One PIU per participant and Coordinator-selected speaker are the reference categories. We weight each episode by the inverse number of PIU-distribution conditions available for its problem instance q_e, so problem instances with more conditions do not receive greater aggregate weight, and cluster standard errors by problem instance.

Across the 11 included problem families and eight models, we reject the joint null that all PIU-distribution–communication-structure interaction terms are zero in both Multi-Agent PIU-Only ($p < .001$) and Multi-Agent Full-Context ($p < .001$). *Thus, which communication structure performs better depends on the PIU distribution.* Appendix Table 35 defines and reports the complete probability-scale interaction estimates, and Appendix G.6 reports the analysis with problem-instance fixed effects. We illustrate this result with two interactions whose 95% confidence intervals exclude zero: (1) changing who initially receives the PIUs reverses which structure performs better, and (2) Coordinator-selected speaker performs better relative to Simultaneous responses when one participant receives most PIUs.

Example 1: Changing the PIU distribution reverses which structure performs better. In Multi-Agent PIU-Only, problem-instance–model pairs from the six legal problem families evaluated under all four combinations show an ordering reversal: Current-speaker selection leads Simultaneous responses under One common PIU (45.4% versus 41.5%) but trails it under Extra participant without a PIU (43.0% versus 52.9%). The resulting interaction is 13.8 percentage points (standard error 2.9). Under Current-speaker selection, later messages can condition on earlier messages; under Simultaneous responses, all messages are generated before any message from the round enters the communication transcript. Appendix Table 37 reports the matched comparison, and Table 38 presents a selected episode illustrating how an initial mistaken framing can propagate.

Example 2: Coordinator-selected speaker performs better relative to Simultaneous responses when one participant receives most PIUs. In Multi-Agent Full-Context, compared with One PIU per participant, All-but-one PIU concentration reduces the difference in episode success rate between Simultaneous responses and Coordinator-selected speaker by 4.0 percentage points (standard error 1.5), shifting the comparison toward Coordinator-selected speaker. This pattern is consistent with a bottleneck under Simultaneous responses: one participant must retrieve and communicate all but one PIU in one message, and the two participants receiving PIUs cannot condition their messages on each other. Appendix G.7 reports the full estimates and selected legal episode.

4.5 Token-Budget Effects

Under Multi-Agent Full-Context with One PIU per participant and Coordinator-selected speaker, we vary private-note, message, and local-transcript token budgets, which respectively limit unshared notes, messages, and newly available transcript text. *Among them, the message budget has the strongest relationship with episode success rate:* for GPT-5.5 on 24 CUAD and 29 Multi-LexSum instances, increasing it from 32 to 128 tokens raises the rate from 55.3% to 78.6%, averaged over private-note budgets, with no further gain at 256 tokens; private-note differences are smaller, and larger local-transcript budgets show no consistent improvement on Multi-LexSum. Appendix G.8 reports complete results and diagnostics.

5 Policy Training with TeamGym: Emergent Coordination

We demonstrate policy training with `TeamGym` under two communication structures: (1) Coordinator-selected speaker, where we train participant and coordinator policies and the coordinator selects and queries participants and makes the final decision; and (2) Current-speaker selection, where we train a shared participant policy and each current speaker either selects the next participant or ends communication. For this demonstration, the participant that ends communication makes the final decision, so no separate final decision maker is used. All policies are initialized from Qwen2.5-7B-Instruct (Yang et al., 2024) and trained using Group Relative Policy Optimization (GRPO) (Shao et al., 2024). In both demonstrations, the LLM-judged score assigned by GPT-5.5 serves as training reward. We call systematic changes in participation, speaker selection, and communication *emergent coordination* when these behaviors are not specified as separate rewards or training targets.

5.1 Coordinator-selected speaker: Training Participant and Coordinator

Under Coordinator-selected speaker, a shared participant policy generates messages from source material, while a coordinator policy generates queries and the final decision. We use Multi-Agent Full-Context on CUAD- and MAUD-derived problem instances (Hendrycks et al., 2021; Wang et al., 2023b), converting each decision task to a free-form written decision format while preserving its PIU distribution and source material (Appendix I.1.1). Training and checkpoint selection use only CUAD-derived problem instances; the 26 MAUD-derived problem instances evaluate transfer. We compare (1) two-stage policy training, which updates the participant and coordinator policies successively to examine which behaviors change when each policy is updated separately, and (2) joint two-policy training, which allows them to co-adapt and serves as the primary end-to-end experiment. Appendices I.1.2 and I.2 provide evaluation, optimization, and checkpoint-selection details.

5.1.1 Two-Stage Policy Training

On the 36 CUAD-derived validation instances, the LLM-judged score increases from 12.14% to 16.84% after the participant-only stage and to 23.33% after the coordinator-only continuation. Illustrative examples show that the participant-only stage changes which PIUs appear in participant messages and, through the resulting transcript, can change queries generated by the fixed coordinator policy; the coordinator-only continuation can change queries and final decisions even when participant messages remain substantively unchanged. *Two-stage policy training can therefore change participant messages, coordinator queries, and final decisions, but its CUAD-derived validation gains do not transfer to the 26 held-out MAUD-derived problem instances.* Appendix I.2.2 provides training details and illustrative examples.

5.1.2 Joint Two-Policy Training

Joint two-policy training produces short coordinator queries and queries every participant, despite not directly rewarding either behavior. On the 36 CUAD-derived validation problem instances, joint two-policy training increases the LLM-judged score from 11.97% to 22.38% and the share of problem instances in which the coordinator queries every participant from 80.6% to 100%. Because querying every participant does not enter the training reward, this increase is an instance of emergent coordination. Table 5 shows distinct coordinator-query behavior: joint two-policy training uses shorter queries, queries every participant, and never copies the problem question in the first query, whereas two-stage policy training uses longer queries and more later queries that reuse terms from preceding participant replies. Figure 3(a) reports the scheduled validation trajectory.

Table 5: **Coordinator-query behavior.** CUAD-derived validation ($n = 36$); percentages except mean words per query. Behavior rates use eligible queries; reuse of preceding-reply terms is measured lexically.

Measurement	Base policies	Two-stage training	Joint two-policy training
Every participant queried	80.6	66.7	**100.0**
Mean words per query	14.0	48.0	**12.5**
Question copied in first query	0.0	88.6	0.0
Later queries reusing preceding-reply terms	3.2	42.9	13.5

Joint two-policy training produces short queries, does not copy the problem question in the first query, and queries every participant on all 36 problem instances. Later queries that reuse terms from preceding participant replies remain infrequent, so the learned coordinator-query behavior is characterized primarily by shorter queries and querying every participant.

On 26 MAUD-derived instances, under joint two-policy training the coordinator queries every participant and every participant sends a public message in all 26, versus 20 under the base policies, while the aggregate LLM-judged score changes only from 5.12% to 5.58%. Thus, coordination behavior transfers even when final-decision quality changes little. Appendix I.2.3 provides training details, an example, and additional analyses.

5.2 Current-speaker selection: Training a Shared Participant Policy

We train a shared participant policy under Current-speaker selection in Multi-Agent PIU-Only with One PIU per participant. The shared policy controls participant messages, next-participant selection, decisions to end communication, and the final decision. Public participant descriptions are visible to support participant selection. Training uses 350 FinDVer problem instances and 188 two- or three-participant FinLongDocQA problem instances. We select the checkpoint using the 39-instance FinDVer validation subset and report separate evaluations on the 21-instance FinLongDocQA validation subset and 40 held-out four-participant FinLongDocQA problem instances. The held-out test comparison therefore asks whether a policy trained on two- and three-participant FinLongDocQA

problem instances also improves performance on four-participant instances. *The trained shared participant policy improves LLM-judged scores on both validation subsets and on held-out four-participant FinLongDocQA problem instances.* Figure 3(b) and Table 6 report the results. Appendix I.3 provides the data-split, optimization, checkpoint-selection, and diagnostic details.

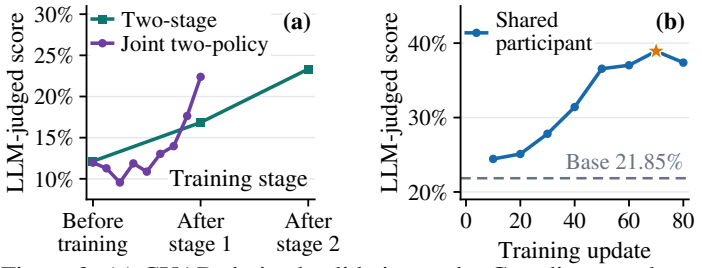


Figure 3: (a) CUAD-derived validation under Coordinator-selected speaker. (b) FinDVer validation under Current-speaker selection.

Table 6: **Validation and test.** LLM-judged scores (%); greedy evaluations.

	FinDVer	FinLongDocQA	
	Val.	Val.	Test
n	39	21	40
Team	2–8	2–3	4
Base	21.85	57.14	33.50
Trained	41.08	69.52	41.13

Table 7: **Cross-replay.** Mean LLM-judged scores (%) completed by both trajectories.

Communication trajectory	Final-decision policy	
	Base	Trained
Base	50.03	49.40
Trained	56.55	56.78

Table 8: **Communication diagnostics.** Base and trained participant policies; percentages except unique speakers.

Communication diagnostic	Validation		FinLongDocQA test	
	Base	Trained	Base	Trained
Every participant speaks	61.7	86.7	62.5	82.5
Mean participant coverage	65.2	89.4	64.4	90.0
Mean unique speakers	2.28	3.25	2.58	3.60
Self-handoff rate	14.2	10.6	17.6	16.2

Here, a communication trajectory comprises the communication transcript, next-participant selections, the decision to end communication, and the identity and assigned PIU of the participant making the final decision. The final-decision policy is the base or trained shared participant policy used only to regenerate the final decision. Cross-replay holds this trajectory and the final-decision prompt fixed while varying the final-decision policy. Across these 40 problem instances, replacing the base communication trajectory with the trained trajectory, averaged across the two final-decision policies, yields a 6.95-point gain. Replacing the base final-decision policy with the trained policy does not produce a statistically significant improvement. *Thus, within cross-replay, the validation gain is concentrated in the communication trajectory rather than the final-decision policy.* Training also increases the proportion of participants who send public messages and the number of distinct speakers while reducing the rate at which a speaker selects itself as the next participant on validation and test. Because these diagnostics do not enter the reward, the changes provide evidence of emergent coordination in participation and next-participant selection.

To test whether additional public messages make more PIUs available for the final decision, we apply the same GPT-5.5 final-decision policy to every base and trained communication transcript. Unlike cross-replay, this analysis retains every trajectory, including valid public messages produced before an episode that does not reach a final decision. On FinDVer, trained transcripts improve performance under the same final-decision policy by 12.6 points and increase the proportion of PIUs appearing in the transcript by 15.9 points. On held-out FinLongDocQA, however, end-to-end score and final-decision completion improve, but the changes in performance under the same final-decision policy and in the proportion of PIUs appearing in the transcript are not statistically distinguishable from zero. *Together, these results show that broader participation and higher end-to-end performance do not necessarily coincide with a larger proportion of PIUs appearing in the communication transcript.* Table 9 reports these comparisons, and Appendix I.3.5 reports additional diagnostics.

Across both fixed communication structures, training changes unrewarded coordination behaviors as well as final-decision quality. Under Coordinator-selected speaker, the participant and coordinator policies change participant messages, coordinator queries, and final decisions; under Current-speaker selection, the shared participant policy changes participation, next-participant selection, and the communication trajectory. Each demonstration reports one training run; results across additional runs and base models remain to be evaluated.

Reproducibility Statement

The supplementary material documents the problem-instance construction procedures, evaluation settings, prompts and evaluator protocols, statistical analyses, and policy-training configurations and diagnostic analyses (Appendices F, E, G, G.8, I.1.2, I.2, and I.3). The accompanying supplementary archive provides the code, configurations, dataset-access and preparation instructions, prompts, and generated artifacts needed to reproduce all reported problem-instance construction, evaluation, analysis, and policy-training results.

Use of AI Assistants

The authors used OpenAI Codex for code development and manuscript editing. For the legal, finance, medical, and software-engineering environment, language models were also used within the research methodology for problem-instance construction and semantic evaluation, as described in the corresponding construction and evaluation sections. These uses include generating reasoning during problem-instance construction when it was not provided, proposing and checking problem annotations, and judging candidate final decisions against reference decisions. The authors designed these procedures, reviewed their outputs, and developed and approved all scientific claims in the final paper.

References

Daniel S. Bernstein, Robert Givan, Neil Immerman, and Shlomo Zilberstein. The complexity of decentralized control of Markov decision processes. *Mathematics of Operations Research*, 27(4): 819–840, 2002. doi: 10.1287/moor.27.4.819.297.

Cohere. Introducing Embed 4: Multimodal search for business. Cohere blog, April 2025. URL https://cohere.com/blog/embed-4. Published April 15, 2025.

Nelson Cowan. The magical number 4 in short-term memory: A reconsideration of mental storage capacity. *Behavioral and Brain Sciences*, 24(1):87–114, 2001. doi: 10.1017/S0140525X01003922.

Vincent P. Crawford and Joel Sobel. Strategic information transmission. *Econometrica*, 50(6): 1431–1451, 1982. doi: 10.2307/1913390.

Jacques Cremer, Luis Garicano, and Andrea Prat. Language and the theory of the firm. *The Quarterly Journal of Economics*, 122(1):373–407, 2007. doi: 10.1162/qjec.122.1.373.

Yilun Du, Shuang Li, Antonio Torralba, Joshua B Tenenbaum, and Igor Mordatch. Improving factuality and reasoning in language models through multiagent debate. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 11733–11763. PMLR, 2024. URL https://proceedings.mlr.press/v235/du24e.html.

Martin J. Eppler and Jeanne Mengis. The concept of information overload: A review of literature from organization science, accounting, marketing, MIS, and related disciplines. *The Information Society*, 20(5):325–344, 2004. doi: 10.1080/01972240490507974.

Francoise Forges. An approach to communication equilibria. *Econometrica*, 54(6):1375–1385, 1986. doi: 10.2307/1914304.

Dino Gerardi and Leeat Yariv. Deliberative voting. *Journal of Economic Theory*, 134(1):317–338, 2007. doi: 10.1016/j.jet.2006.05.002. URL https://doi.org/10.1016/j.jet.2006.05.002.

Joseph Y. Halpern and Yoram Moses. Knowledge and common knowledge in a distributed environment. *Journal of the ACM*, 37(3):549–587, 1990. doi: 10.1145/79147.79161.

Anne Lundgaard Hansen, John J. Horton, Sophia Kazinnik, Daniela Puzzello, and Ali Zarifhonarvar. Simulating the survey of professional forecasters. Working Paper 5066286, SSRN, January 2026. URL https://ssrn.com/abstract=5066286. Last revised July 15, 2026.

Dan Hendrycks, Collin Burns, Anya Chen, and Spencer Ball. CUAD: An expert-annotated NLP dataset for legal contract review. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1, 2021. URL https://openreview.net/forum?id=7l1Ygs3Bamw.

Savindu Herath Pathirannehelage, Yash Raj Shrestha, and Georg von Krogh. Design principles for artificial intelligence-augmented decision making: An action design research study. *European Journal of Information Systems*, 34(2):207–229, 2025. doi: 10.1080/0960085X.2024.2330402. URL https://doi.org/10.1080/0960085X.2024.2330402.

Nils Holzenberger, Andrew Blair-Stanek, and Benjamin Van Durme. A dataset for statutory reasoning in tax law entailment and question answering. In *Proceedings of the Natural Legal Language Processing Workshop 2020*, volume 2645 of *CEUR Workshop Proceedings*, pp. 31–38. CEUR-WS.org, 2020. URL https://ceur-ws.org/Vol-2645/paper5.pdf.

Sirui Hong, Mingchen Zhuge, Jonathan Chen, Xiawu Zheng, Yuheng Cheng, Jinlin Wang, Ceyao Zhang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, Liyang Zhou, Chenyu Ran, Lingfeng Xiao, Chenglin Wu, and Jürgen Schmidhuber. MetaGPT: Meta programming for a multi-agent collaborative framework. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=VtmBAGCN7o. Oral presentation.

Wenqi Huang, Charley Lee, Leonard Tng, and Serena Ge. DeepSWE: Measuring frontier coding agents on original, long-horizon engineering tasks. *arXiv preprint arXiv:2607.07946*, 2026. doi: 10.48550/arXiv.2607.07946. URL https://arxiv.org/abs/2607.07946.

Daniel Kahneman. *Attention and Effort*. Prentice-Hall, Englewood Cliffs, NJ, 1973. ISBN 0-13-050518-8. URL https://kahneman.scholar.princeton.edu/sites/g/files/toruqf3831/files/kahneman/files/attention_hi_quality.pdf.

Arpandeep Khatua, Hao Zhu, Peter Tran, Arya Prabhudesai, Frederic Sadrieh, Johann K. Lieberwirth, Xinkai Yu, Yicheng Fu, Michael J. Ryan, Jiaxin Pei, and Diyi Yang. Cooperbench: Why coding agents cannot be your teammates yet. *arXiv preprint arXiv:2601.13295*, 2026. doi: 10.48550/arXiv.2601.13295. URL https://arxiv.org/abs/2601.13295.

Daewoo Kim, Sangwoo Moon, David Hostallero, Wan Ju Kang, Taeyoung Lee, Kyunghwan Son, and Yung Yi. Learning to schedule communication in multi-agent reinforcement learning. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=SJxu5iR9KQ.

Yubin Kim, Chanwoo Park, Hyewon Jeong, Yik Siu Chan, Xuhai Xu, Daniel McDuff, Hyeonhoon Lee, Marzyeh Ghassemi, Cynthia Breazeal, and Hae Won Park. Mdagents: An adaptive collaboration of llms for medical decision-making. In *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-2522. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/90d1fc07f46e31387978b88e7e057a31-Abstract-Conference.html.

Yubin Kim, Ken Gu, Chanwoo Park, Chunjong Park, Samuel Schmidgall, A Ali Heydari, Yao Yan, Zhihan Zhang, Yuchen Zhuang, Yun Liu, et al. Capable language models can outgrow the benefits of collaboration. *Nature Machine Intelligence*, 8(7):1157–1172, 2026a.

Yubin Kim, Chanwoo Park, Taehan Kim, Eugene Park, Samuel Schmidgall, Salman Rahman, Chunjong Park, Cynthia Breazeal, Xin Liu, Hamid Palangi, et al. Teambench: Evaluating agent coordination under enforced role separation. *arXiv preprint arXiv:2605.07073*, 2026b.

Vivian Lai, Chacha Chen, Q Vera Liao, Alison Smith-Renner, and Chenhao Tan. Towards a science of human-AI decision making: A survey of empirical studies. *arXiv preprint arXiv:2112.11471*, 2021. doi: 10.48550/arXiv.2112.11471. URL https://arxiv.org/abs/2112.11471.

Guohao Li, Hasan Abed Al Kader Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem. CAMEL: Communicative agents for "mind" exploration of large language model society. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/a3621ee907def47c1b952ade25c67698-Abstract-Conference.html.

Yuxuan Li, Aoi Naito, and Hirokazu Shirado. Assessing collective reasoning in multi-agent llms via hidden profile tasks. *arXiv preprint arXiv:2505.11556v1*, 2025. URL https://arxiv.org/abs/2505.11556v1.

Yuxuan Li, Aoi Naito, and Hirokazu Shirado. Systematic failures in collective reasoning under distributed information in multi-agent llms. *arXiv preprint arXiv:2505.11556v4*, 2026. URL https://arxiv.org/abs/2505.11556v4. Accepted at ICML 2026.

Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, Shudan Zhang, Xiang Deng, Aohan Zeng, Zhengxiao Du, Chenhui Zhang, Sheng Shen, Tianjun Zhang, Yu Su, Huan Sun, Minlie Huang, Yuxiao Dong, and Jie Tang. Agentbench: Evaluating llms as agents. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=zAdUB0aCTQ.

Li Lu, Y. Connie Yuan, and Poppy Lauretta McLeod. Twenty-five years of hidden profiles in group decision making: A meta-analysis. *Personality and Social Psychology Review*, 16(1):54–75, 2012. doi: 10.1177/1088868311417243.

Jacob Marschak and Roy Radner. *Economic Theory of Teams*. Number 22 in Cowles Foundation Monograph. Yale University Press, New Haven, CT, 1972. URL https://cowles.yale.edu/research/cfm-22-economic-theory-teams.

Eric Maskin and Tomas Sjostrom. Implementation theory. In Kenneth J. Arrow, Amartya K. Sen, and Kotaro Suzumura (eds.), *Handbook of Social Choice and Welfare*, volume 1, pp. 237–288. Elsevier, 2002. doi: 10.1016/S1574-0110(02)80009-1.

Dilip Mookherjee. Decentralization, hierarchies, and incentives: A mechanism design perspective. *Journal of Economic Literature*, 44(2):367–390, 2006. doi: 10.1257/jel.44.2.367.

Roger B. Myerson. Incentive compatibility and the bargaining problem. *Econometrica*, 47(1):61–73, 1979. doi: 10.2307/1912346.

Roger B. Myerson. Multistage games with communication. *Econometrica*, 54(2):323–358, 1986. doi: 10.2307/1913154.

Roger B. Myerson. Perspectives on mechanism design in economic theory. *American Economic Review*, 98(3):586–603, 2008. doi: 10.1257/aer.98.3.586.

Ashutosh Nayyar, Aditya Mahajan, and Demosthenis Teneketzis. Decentralized stochastic control with partial history sharing: A common information approach. *IEEE Transactions on Automatic Control*, 58(7):1644–1658, 2013. doi: 10.1109/TAC.2013.2239000.

Office of the Federal Register. eCFR API Documentation. Electronic Code of Federal Regulations, 2026a. URL https://www.ecfr.gov/developers/documentation/api/v1. Accessed September 1, 2026.

Office of the Federal Register. 12 CFR Part 1 – Investment Securities. Electronic Code of Federal Regulations, 2026b. URL https://www.ecfr.gov/current/title-12/chapter-I/part-1. Accessed September 1, 2026.

Office of the Federal Register. 12 CFR Part 3 – Capital Adequacy Standards. Electronic Code of Federal Regulations, 2026c. URL https://www.ecfr.gov/current/title-12/chapter-I/part-3. Accessed September 1, 2026.

Alon Orlitsky and James R. Roche. Coding for computing. *IEEE Transactions on Information Theory*, 47(3):903–917, 2001. doi: 10.1109/18.915643.

Chanwoo Park, Seungju Han, Xingzhi Guo, Asuman E. Ozdaglar, Kaiqing Zhang, and Joo-Kyung Kim. MAPoRL: Multi-agent post-co-training for collaborative large language models with reinforcement learning. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 30215–30248. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.acl-long.1459. URL https://aclanthology.org/2025.acl-long.1459/.

Joon Sung Park, Lindsay Popowski, Carrie Cai, Meredith Ringel Morris, Percy Liang, and Michael S Bernstein. Social simulacra: Creating populated prototypes for social computing systems. In *Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology*, pp. 1–18, 2022. doi: 10.1145/3526113.3545616. URL https://doi.org/10.1145/3526113.3545616.

Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, UIST '23, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9798400701320. doi: 10.1145/3586183.3606763. URL https://doi.org/10.1145/3586183.3606763.

Chen Qian, Wei Liu, Hongzhang Liu, Nuo Chen, Yufan Dang, Jiahao Li, Cheng Yang, Weize Chen, Yusheng Su, Xin Cong, Juyuan Xu, Dahai Li, Zhiyuan Liu, and Maosong Sun. ChatDev: Communicative agents for software development. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15174–15186. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.acl-long.810. URL https://aclanthology.org/2024.acl-long.810/.

Roy Radner. Team decision problems. *The Annals of Mathematical Statistics*, 33(3):857–881, 1962. doi: 10.1214/aoms/1177704455.

Mohit Raghavendra, Soham Dan, Miguel Romero Calvo, Yannis Yiming He, Johannes Baptist Mols, Gautam Anand, Cole McCollum, Edgar Arakelyan, Vijay Bharadwaj, Andrew Park, Jeff Da, MohammadHossein Rezaei, Bing Liu, Brad Kenstler, and Yunzhong He. SWE Atlas: Benchmarking coding agents beyond issue resolution. *arXiv preprint arXiv:2605.08366*, 2026. doi: 10.48550/arXiv.2605.08366. URL https://arxiv.org/abs/2605.08366.

Harvey Sacks, Emanuel A. Schegloff, and Gail Jefferson. A simplest systematics for the organization of turn-taking for conversation. *Language*, 50(4):696–735, 1974. doi: 10.2307/412243.

Timo Schick, Jane Dwivedi-Yu, Zhengbao Jiang, Fabio Petroni, Patrick Lewis, Gautier Izacard, Qingfei You, Christoforos Nalmpantis, Edouard Grave, and Sebastian Riedel. PEER: A collaborative language model. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=KbYevcLjnc.

Isabella Seeber, Eva Bittner, Robert O. Briggs, Triparna de Vreede, Gert-Jan de Vreede, Aaron Elkins, Ronald Maier, Alexander B. Merz, Sarah Oeste-Reiß, Nils Randrup, Gerhard Schwabe, and Matthias Söllner. Machines as teammates: A research agenda on AI in team collaboration. *Information & Management*, 57(2):103174, 2020. doi: 10.1016/j.im.2019.103174.

Ilya Segal. The communication requirements of social choice rules and supporting budget sets. *Journal of Economic Theory*, 136(1):341–378, 2007. doi: 10.1016/j.jet.2006.09.011.

Yijia Shao, Vinay Samuel, Yucheng Jiang, John Yang, and Diyi Yang. Collaborative gym: A framework for enabling and evaluating human-agent collaboration. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=GDYueXtKXT.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024. doi: 10.48550/arXiv.2402.03300. URL https://arxiv.org/abs/2402.03300.

Zejiang Shen, Kyle Lo, Lauren Yu, Nathan Dahlberg, Margo Schlanger, and Doug Downey. Multi-LexSum: Real-world summaries of civil rights lawsuits at multiple granularities. In *Advances in Neural Information Processing Systems*, volume 35, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/552ef803bef9368c29e53c167de34b55-Abstract-Datasets_and_Benchmarks.html.

Garold Stasser and William Titus. Pooling of unshared information in group decision making: Biased information sampling during discussion. *Journal of Personality and Social Psychology*, 48(6):1467–1478, 1985. doi: 10.1037/0022-3514.48.6.1467.

Pramod K. Varshney. *Distributed Detection and Data Fusion*. Springer, New York, 1997. doi: 10.1007/978-1-4612-1904-0. URL https://doi.org/10.1007/978-1-4612-1904-0.

Jianyou Wang, Weili Cao, Kaicheng Wang, Xiaoyue Wang, Ashish Dalvi, Gino Prasad, Qishan Liang, Hsuan-Lin Her, Ming Wang, Qin Yang, Gene W. Yeo, David E. Neal, Maxim Khan, Christopher D. Rosin, Ramamohan Paturi, and Leon Bergen. EvidenceBench: A benchmark for extracting evidence from biomedical papers. In *Conference on Language Modeling*, 2025. URL https://openreview.net/forum?id=lEQnUI5lEA.

Peifeng Wang, Zhengyang Wang, Zheng Li, Yifan Gao, Bing Yin, and Xiang Ren. SCOTT: Self-consistent chain-of-thought distillation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 5546–5558, Toronto, Canada, 2023a. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.304. URL https://aclanthology.org/2023.acl-long.304/.

Steven H. Wang, Antoine Scardigli, Leonard Tang, Wei Chen, Dmitry Levkin, Anya Chen, Spencer Ball, Thomas Woodside, Oliver Zhang, and Dan Hendrycks. MAUD: An expert-annotated legal NLP dataset for merger agreement understanding. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 16369–16382, Singapore, 2023b. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.1019. URL https://aclanthology.org/2023.emnlp-main.1019/.

Yi-Cheng Wang, Wei-An Wang, and Chu-Song Chen. Document-level numerical reasoning across single and multiple tables in financial reports. *arXiv preprint arXiv:2604.03664*, 2026. doi: 10.48550/arXiv.2604.03664. URL https://arxiv.org/abs/2604.03664.

Wisedocs. Medical Long Context Reasoning (MLCR). GitHub repository, 2026. URL https://github.com/Wisedocs-AI/medical-long-context-reasoning. Accessed: 2026-09-20.

Anita Williams Woolley, Christopher F. Chabris, Alex Pentland, Nada Hashmi, and Thomas W. Malone. Evidence for a collective intelligence factor in the performance of human groups. *Science*, 330(6004):686–688, 2010. doi: 10.1126/science.1193147.

Feng Wu, Shlomo Zilberstein, and Xiaoping Chen. Online planning for multi-agent systems with bounded communication. *Artificial Intelligence*, 175(2):487–511, 2011. doi: 10.1016/j.artint.2010.09.008.

Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, Ahmed Hassan Awadallah, Ryen W. White, Doug Burger, and Chi Wang. AutoGen: Enabling next-gen LLM applications via multi-agent conversation. *arXiv preprint arXiv:2308.08155*, 2023. doi: 10.48550/arXiv.2308.08155. URL https://arxiv.org/abs/2308.08155.

Yue Wu, Xuan Tang, Tom Mitchell, and Yuanzhi Li. Smartplay: A benchmark for llms as intelligent agents. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=3wleyb0j5P.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. Qwen2.5 Technical Report. *arXiv preprint arXiv:2412.15115*, 2024. doi: 10.48550/arXiv.2412.15115. URL https://arxiv.org/abs/2412.15115.

Andrew Chi-Chih Yao. Some complexity questions related to distributive computing (preliminary report). In *Proceedings of the Eleventh Annual ACM Symposium on Theory of Computing*, pp. 209–213. Association for Computing Machinery, 1979. doi: 10.1145/800135.804414.

Wenhan Yu, Xinbo Lin, Lanxin Ni, Jinhua Cheng, and Lei Sha. Benchmarking multi-step legal reasoning and analyzing chain-of-thought effects in large language models. *Information Processing & Management*, 64(1):105030, 2027. doi: 10.1016/j.ipm.2026.105030. URL https://doi.org/10.1016/j.ipm.2026.105030.

Serdar Yüksel and Tamer Başar. Networked control systems as stochastic team decision problems: A general introduction. In *Systems and Control*, Systems and Control: Foundations and Applications, pp. 11–75. Birkhäuser, 2013. doi: 10.1007/978-1-4614-7085-4_2.

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

Yilun Zhao, Yitao Long, Tintin Jiang, Chengye Wang, Weiyuan Chen, Hongjun Liu, Xiangru Tang, Yiming Zhang, Chen Zhao, and Arman Cohan. FinDVer: Explainable claim verification over long and hybrid-content financial documents. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 14739–14752, Miami, Florida, USA, 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.818. URL https://aclanthology.org/2024.emnlp-main.818/.

Changxi Zhu, Mehdi Dastani, and Shihan Wang. A survey of multi-agent deep reinforcement learning with communication. *Autonomous Agents and Multi-Agent Systems*, 38(1):4, 2024. doi: 10.1007/s10458-023-09633-6.

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

A Deferred Table for Section 5.2: Communication-Trajectory Analysis

Table 9: **Communication-trajectory analysis.** Arrows compare the base and trained policies. All entries are percentages. "Same final-decision policy" is the LLM-judged score obtained by applying one common GPT-5.5 final-decision policy to every communication transcript.

	LLM-judged score	Final-decision completion	Same final-decision policy	PIUs appearing in the communication transcript	Exactly repeated public messages
FinDVer validation	21.85 → 41.08	51.3 → 84.6	34.10 → 46.72	38.0 → 53.9	18.9 → 50.9
FinLongDocQA test	33.50 → 41.13	65.0 → 92.5	49.60 → 44.82	63.1 → 63.1	23.1 → 48.6

B Deferred Conclusion

We introduce `TeamGym`, a gym for constructing team decision problem instances and for evaluating and training participant and coordinator policies under controlled PIU distributions, communication structures, and token budgets. Across 8 language models and 12 source corpora, distributing the same PIUs across participants reduces episode success rate by 22.9 percentage points relative to providing them to a single model, and requiring participants to retrieve those PIUs from source material reduces it by a further 19.9 points. The effectiveness of a communication structure also varies across PIU-distribution conditions. Team decision-making therefore depends not only on individual model capabilities, but also on how participants retrieve, communicate, and integrate distributed information.

Across two fixed communication structures, policy training improves final-decision quality and induces emergent coordination in participation, speaker selection, and communication. With coordinator control, joint two-policy training produces a compact information-elicitation protocol; without a coordinator, training improves end-to-end final-decision quality and completion while increasing participation. Tracing PIUs through these trajectories shows that increased participation makes more pivotal information available on FinDVer validation, but not on held-out FinLongDocQA. This distinction illustrates how `TeamGym` diagnoses how training changes team decision making, rather than reporting only whether performance improves.

These training demonstrations use one run each, and the improvements under Coordinator-selected speaker are measured on CUAD-derived validation problem instances used for checkpoint selection. On MAUD-derived problem instances, the learned query and messaging patterns transfer more clearly than final-decision quality. Extensions of `TeamGym` could relax the shared-payoff assumption to study strategic information transmission, malicious or unreliable participants, and teams with partially aligned objectives. They could also introduce explicit retrieval costs and allow participants to choose which source material to retrieve. By separating retrieval, communication, and final decisions while holding the underlying problem fixed, `TeamGym` provides a controlled foundation for diagnosing and training team decision-making capabilities.

C Additional Related Work and Benchmark Comparisons

Mechanism design and communication. Multistage communication games, communication equilibria, and communication requirements study communication before a decision (Myerson, 1986; Forges, 1986; Segal, 2007). Cheap-talk models instead study communication with preference misalignment (Crawford & Sobel, 1982).

Human-AI and group decision making. Human-AI decision making motivates evaluating AI systems as team participants and examining the interaction process rather than only the final decision (Lai et al., 2021; Herath Pathirannehelage et al., 2025; Seeber et al., 2020). Hidden-profile studies show that human teams can fail to reveal uniquely held information, and collective-intelligence research relates interaction structure to performance (Stasser & Titus, 1985; Woolley et al., 2010). Work on limited attention, working memory, and information overload motivates finite resource constraints (Kahneman, 1973; Cowan, 2001; Eppler & Mengis, 2004). These findings motivate our

controlled comparisons, but TeamGym does not evaluate human-AI teams or treat its token budgets as models of human cognition. Work using LLMs to simulate social settings has a different objective: plausible social behavior is the object of study, whereas TeamGym treats the participant model as a controlled component (Park et al., 2022; 2023; Hansen et al., 2026).

Multi-agent LLM systems and benchmark contrasts. CAMEL, AutoGen, MetaGPT, and ChatDev study collaboration through prompted agent interaction (Li et al., 2023; Wu et al., 2023; Hong et al., 2024; Qian et al., 2024). PEER learns collaborative writing, MDAgents studies adaptive medical collaboration, and other work studies debate and collaborative post-training (Schick et al., 2023; Kim et al., 2024; Du et al., 2024; Park et al., 2025). CooperBench evaluates cooperative coding over repositories (Khatua et al., 2026). These systems study agent coordination, but TeamGym fixes source-grounded PIUs, controls the PIU distribution and whether participants receive PIUs directly or source material containing them, and measures whether bounded communication makes the PIUs available for a final decision. Multi-agent debate commonly gives agents the same problem; TeamGym gives participants different initial local observations.

MultiAgentBench evaluates coordination under several communication topologies (Zhu et al., 2025). HiddenBench evaluates whether teams surface and combine directly presented hidden facts (Li et al., 2025; 2026), while SILO-BENCH studies role-free collaboration on synthetic distributed-computation problems (Zhang et al., 2026). Appendix C.1 gives compact comparisons with both benchmarks. Collaborative Gym studies a user or user simulator collaborating with one LLM agent, while our unit of analysis is a team with decision-relevant evidence divided among participants (Shao et al., 2026). Single-agent suites such as SmartPlay and AgentBench do not require recovery of evidence held by teammates (Wu et al., 2024; Liu et al., 2024).

The original evaluations for the finance, medical, and software-engineering source corpora in Table 1 are single-agent or centralized. FinDVer and FinLongDocQA study financial verification and long-document financial question answering (Zhao et al., 2024; Wang et al., 2026); MLCR and EvidenceBench provide long medical-record cases and hypothesis-linked biomedical evidence (Wisedocs, 2026; Wang et al., 2025); and SWE Atlas Codebase QnA studies repository comprehension (Raghavendra et al., 2026). TeamGym uses the evaluated source corpora to construct team decision problem instances with controlled PIU distributions.

C.1 Closest Benchmark Comparisons

HiddenBench. HiddenBench is the closest benchmark to TeamGym because both study team decisions under decentralized information structures. HiddenBench provides broad coverage of Hidden Profile settings (Version 4 evaluates 65 problem instances and 15 models) and examines whether directly presented participant-specific facts enter public discussion and improve group voting (Li et al., 2025; 2026). TeamGym complements this breadth by treating the source-grounded information and interaction environment as controlled experimental variables: it varies PIU distribution, whether participants receive PIUs directly or source material containing them, communication structure, and token budgets while holding the underlying problem and evidence requirements fixed. Paired contrasts and PIU-distribution–communication-structure regressions quantify how performance changes across these conditions, while PIU-level process tracing localizes plausible stages at which required evidence is lost. TeamGym additionally supports a participant-only stage, a coordinator-only continuation, and joint two-policy training under Coordinator-selected speaker, together with training of a shared participant policy under Current-speaker selection, with the participant that ends communication making the final decision. Thus, HiddenBench provides a broader Hidden Profile evaluation, whereas TeamGym provides a more controlled, source-grounded environment for studying how PIU distribution and communication structure jointly relate to team performance.

SILO-BENCH. SILO-BENCH studies role-free agents solving synthetic distributed-computation problems under direct-message, broadcast, and shared-file protocols, with programmatically known answers and weak-scaling evaluations from 2 to 100 agents (Zhang et al., 2026). This design is particularly useful for examining how emergent organization, consensus, and communication cost change as both team size and the global problem grow. TeamGym instead holds each real-world decision problem and its evidence requirements fixed while varying PIU distribution, whether participants receive PIUs directly or source material containing them, communication structure, and

token budgets. Its communication gap compares Single-Agent All-PIU with Multi-Agent PIU-Only, while its retrieval gap compares Multi-Agent PIU-Only with Multi-Agent Full-Context. These outcome-based contrasts are distinct from SILO-BENCH's log-based Communication-Reasoning Gap. `TeamGym` also supports a participant-only stage, a coordinator-only continuation, and joint two-policy training under Coordinator-selected speaker, together with training of a shared participant policy under Current-speaker selection, with the participant that ends communication making the final decision. Consequently, SILO-BENCH characterizes how emergent collaboration behaves under weak scaling, whereas `TeamGym` more directly examines how controlled PIU distributions and communication structures relate to performance on fixed, source-grounded team decisions.

D Conceptual Illustrations of Communication Structures and PIU Distributions

Figure 4 illustrates the interaction flows associated with the five communication structures in Table 10. Figure 5 illustrates the five PIU-allocation patterns in Table 2.

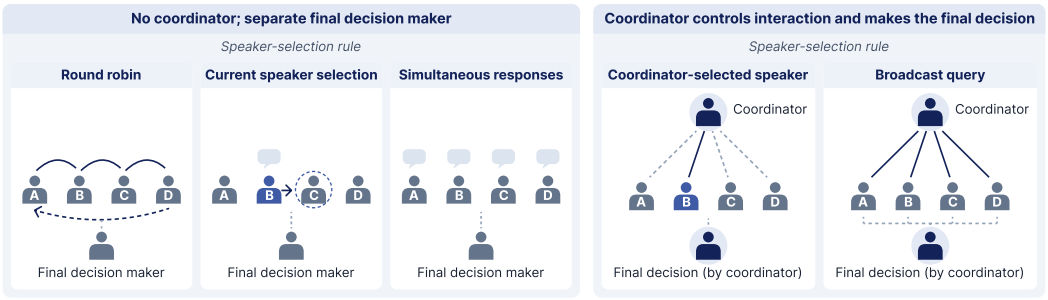


Figure 4: **Communication structures ($\mathcal{C}$).** The three structures on the left have no coordinator and use a separate final decision maker. In the two structures on the right, a coordinator controls interaction and makes the final decision.

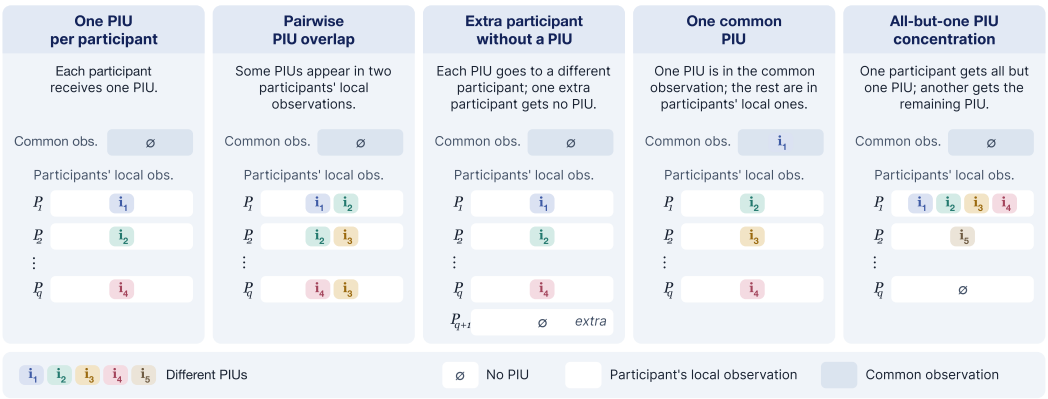


Figure 5: **Five PIU-distribution conditions.** Each condition distributes the $\mathcal{I}_q^{\min}$ across the initial local and common observations without changing q or d^{ref}. Colored circles denote distinct PIUs; dashed and solid boxes denote the initial common and local observations, respectively. Not every problem instance admits every condition.

Communication control and final decision maker	Speaker-selection rule
No coordinator; separate final decision maker	Round robin
	Current-speaker selection
	Simultaneous responses
Coordinator controls communication and makes the final decision	Coordinator-selected speaker
	Broadcast query

Table 10: **Communication structures used in evaluation ($\mathcal{C}$).** The structures vary in who controls communication and makes the final decision, and in the *speaker-selection rule*, which specifies which participants may send messages in each communication round.

E Communication-Structure Taxonomy, Implementation, and Relation to Prior Work

Message generation and structure-specific implementation. At each communication round, the environment constructs the current observation permitted by the communication structure and resource budgets. The applicable participant or coordinator policy receives this observation and generates an action permitted for its role, such as a participant message, coordinator query and speaker selection, or final decision. The implementation uses role-specific prompt templates and structured output schemas, parses the returned action, and applies structure-specific fallback rules. Model and decoding settings are reported with the corresponding evaluation or training setup. Under Current-speaker selection, an omitted or invalid next-participant selection defaults to the next participant in the fixed cyclic order.

Relation to prior work. The three structures without coordinator control are related to public deliberation and multi-agent debate (Gerardi & Yariv, 2007; Du et al., 2024), whereas the two coordinator-controlled structures are related to mediated communication and fusion-center models (Myerson, 1986; Forges, 1986; Varshney, 1997). Their speaker-selection rules also draw on mechanisms studied in conversation analysis and multi-agent systems (Sacks et al., 1974; Wu et al., 2023; Kim et al., 2019). The five structures are not intended as an exhaustive taxonomy of communication structures.

F Problem-Instance Construction and Statistics

We report source-specific construction choices, checks, and problem-instance statistics for the legal, finance, medical, and software-engineering source corpora. Sections 3.1–3.2 define the terminology and general construction requirements; here we provide implementation details and source-specific statistics.

Source terminology.

- A *source corpus* is the collection of records, annotations, or other artifacts from which problem instances are constructed.
- The *underlying source materials* for a problem instance q are the particular records or artifacts selected from its source corpus.
- *Source material* denotes the portions of the underlying source materials placed in participants' initial observations.

F.1 Decision Sufficiency and Pivotal Information

For a problem instance q and an evidence subset S, let $v_q(S)$ denote the corresponding *information view*: the decision objective, decision format, any other fixed public information, and the evidence in S. The view excludes omitted source information, the reference decision, evaluator-side annotations, and other hidden construction metadata.

Let $\mathcal{Q}_q$ be a comparison class containing q and other admissible problem instances that preserve q's public decision specification and set $\mathcal{D}_q$ of possible final decisions. Admissibility is defined relative to the problem family: an admissible instance must satisfy the same domain constraints and evaluation rule while potentially differing in information not contained in the given view. For an information view v, define

$$\Omega_q(v) = \{\widetilde{q} \in \mathcal{Q}_q : \widetilde{q} \text{ is consistent with } v\}.$$

Thus, the information in v does not distinguish q from the other instances in $\Omega_q(v)$.

For each $\widetilde{q} \in \mathcal{Q}_q$, let

$$\mathcal{D}^\star_{\widetilde{q}} = \arg\max_{d \in \mathcal{D}_q} U_{\widetilde{q}}(d)$$

be its set of optimal final decisions. We call S *decision-sufficient* for q if

$$\bigcap_{\widetilde{q} \in \Omega_q(v_q(S))} \mathcal{D}^\star_{\widetilde{q}} \neq \varnothing.$$

Equivalently, the information view identifies at least one final decision that is optimal for every admissible problem instance consistent with that view.

We call $\mathcal{I}^{\min}_q = \{i_1, \ldots, i_k\}$ the *pivotal information* for q, and each $i_j \in \mathcal{I}^{\min}_q$ a *pivotal information unit (PIU)*, if

$$\bigcap_{\widetilde{q} \in \Omega_q(v_q(\mathcal{I}^{\min}_q))} \mathcal{D}^\star_{\widetilde{q}} \neq \varnothing$$

and, for every $i_j \in \mathcal{I}^{\min}_q$,

$$\bigcap_{\widetilde{q} \in \Omega_q(v_q(\mathcal{I}^{\min}_q \setminus \{i_j\}))} \mathcal{D}^\star_{\widetilde{q}} = \varnothing.$$

The pivotal information therefore identifies a common optimal decision, whereas after removing any PIU, no final decision is guaranteed to be optimal across all admissible instances consistent with the remaining information.

The construction procedure does not enumerate $\Omega_q(v_q(S))$. Instead, it operationalizes the complete-set and leave-one-out conditions using source annotations, reference reasoning, grounding checks, and model-based sufficiency checks. The resulting $\mathcal{I}^{\min}_q$ need not be unique: another inclusion-minimal evidence set may also identify the decision. Separately, the centralized checks under Single-Agent All-PIU test whether the reference decision can be produced from the selected pivotal information.

F.2 General Construction Procedure

Reference decisions and reasoning. Each problem instance is constructed from one or more source questions; the decision objective may retain one question or combine related questions. When a reference decision and the reasoning used to produce it accompany the source questions, we retain them. Otherwise, we construct the reference decision from the available source annotations and underlying source materials. When the reasoning is unavailable, we generate reasoning conditioned on the reference decision; answer-conditioned rationale generation provides one example of this procedure (Wang et al., 2023a).

Candidate-unit and removal checks. The fixed decision objective, reference decision, and reasoning used to produce it identify the information in the underlying source materials that supports the required answer components or reasoning steps. Dividing this information into candidate units is source-specific; a candidate unit may be a sentence, paragraph, or code block. The construction procedure requires each retained candidate unit to be necessary for at least one required answer component or reasoning step. When the retained or generated reasoning does not establish whether this requirement holds, a model-based check using a model such as GPT-5.5 first requires the model to produce the reference decision. The procedure then removes one candidate unit at a time and asks whether uncertainty remains about the reference decision given the decision objective and remaining units. The removed unit is retained when uncertainty remains and excluded otherwise. This process continues until removing any retained unit leaves uncertainty about the reference decision; the retained units form $\mathcal{I}^{\min}_q$. When necessity is established from the retained or generated reasoning, the released construction code also provides the model-based removal check as an additional check.

PIU and source-material distribution. For Pairwise PIU overlap, we use a seeded random ordering to assign PIUs to the participant positions defined by the condition. Appendix F.3.8 reports an additional Multi-LexSum analysis that orders PIUs using embedding-based similarity while preserving the condition. After the PIU distribution is fixed, the underlying source materials are divided into units without dividing a PIU. Units that do not contain a PIU are distributed across participants using source-specific procedures that may account for their distance from a participant's PIUs and balance the lengths of participants' initial local observations. Afterward, we verify each participant's initial local observation together with the initial common observation and, when necessary and consistent with the source, truncate the material. For example, legal source materials may be truncated, whereas source code is not truncated when doing so would prevent the program from functioning.

Participant identifiers and participant descriptions. A participant description is a prompt-visible phrase describing a participant's subject matter or function without disclosing the specific content, values, or conclusions of the PIUs assigned to that participant. Participant identifiers and participant descriptions disclosed to every decision maker form part of the initial common observation. Because participant descriptions indicate which participant is likely to hold relevant information, decision makers can use them to inform speaker selection. Participant descriptions do not determine the structural roles of participant, coordinator, or separate final decision maker defined in Section 3.1. The source-specific procedures below describe their construction and leakage checks.

F.3 LEGAL

Table 11 summarizes the six legal problem families and their source corpora.

Table 11: **Legal problem families and source corpora.**

Source corpus	Underlying source materials	Reference decision	Problem family
CUAD	Expert-annotated commercial contracts	Constructed alternative	Contract-profile decision
MAUD	Expert-curated merger-agreement provisions	Constructed alternative	Merger-review decision
eCFR Casebook	Regulations and controlled transaction facts	Written conclusion	Compliance analysis
MSLR	Anonymized enforcement case files	Final sanction record	Enforcement conclusion
Multi-LexSum	Documents from a litigation collection	Case-grounded decision	Procedural analysis
SARA	Tax facts and supplied statutory provisions	Numeric result	Tax calculation

F.3.1 COMMON LEGAL CONSTRUCTION PROCEDURE

Step 1. Select the underlying source materials. A problem instance draws from exactly one contract, agreement, transaction specification, case file, litigation collection, or tax case, according to its source corpus.

Step 2. Specify the decision objective and format. For CUAD and MAUD, a construction model uses one or more questions and their grounded excerpts from the source corpus to construct a new finite-choice legal-review question and reference decision. It constructs a decision set $\mathcal{D}_q$ of mutually exclusive, legally plausible alternatives with comparable specificity. The other alternatives have similar complexity to the reference decision, and the alternatives are written as natural legal profiles rather than explicit sums of answer components. For CUAD and MAUD, the grounded excerpts used to support the reference decision are candidate units. The other four source corpora use written or numeric final decisions. Construction uses the reference decision and the reasoning used to produce it to identify candidate units in the underlying source materials. The source-specific procedures below determine which candidate units form $\mathcal{I}_q^{\min}$.

Step 3. Identify the pivotal information. We apply the PIU construction requirements in Section 3.2. If the checks identify multiple inclusion-minimal sets that satisfy these requirements, CUAD and MAUD retain one as the pivotal information before constructing the PIU distribution. eCFR Casebook, MSLR, and Multi-LexSum identify the pivotal information directly from units needed for distinct reasoning steps.

Step 4. Define the PIU distribution. We apply the PIU-distribution conditions in Table 2 subject to the construction requirements in Section 3.2. Some conditions are ineligible for problem instances with only two PIUs; we do not add artificial PIUs to force a condition.

Step 5. Construct participant descriptions. Each participant receives a participant description based on the PIUs assigned to that participant and not included in the initial common observation. For each PIU, a construction model produces a short phrase describing its broad subject or function. The phrase must not disclose the PIU's specific content, values, conclusions, or effect on the final decision. The phrases corresponding to the PIUs assigned to a participant are combined deterministically to form the participant description for that participant. A participant assigned no PIU outside the initial common observation receives the participant description *context reviewer*, followed by the participant number. The participant descriptions form part of the initial common observation. For eCFR Casebook, MSLR, Multi-LexSum, and SARA, a separate model-based check assesses this requirement, and descriptions that fail the check are revised.

Step 6. Check initial observations and complete problem-instance construction. We apply the participant-level source-material check in Section 3.2. Removable non-PIU information that permits one participant to produce the correct final decision alone is deleted and the check is repeated; a problem instance is rejected if such information cannot be removed. Finite-choice labels are permuted after construction to prevent position regularities.

F.3.2 Problem Instances and PIU Distributions in the Legal Evaluation

The legal model comparison evaluates the same 163 retained problem instances under the three evaluation conditions. Because CUAD and MAUD contain substantially more retained problem instances than the other legal source corpora, we use 30 problem instances from each, with at most one problem instance per contract or agreement. For eCFR Casebook, MSLR, Multi-LexSum, and SARA, we use every retained problem instance. Each problem instance uses the One PIU per participant condition in the two team conditions.

The PIU-distribution–communication-structure evaluation uses 514 PIU distributions under the five PIU-distribution conditions in Table 2. For each of the eight models, the evaluation design crosses each PIU distribution with the five communication structures in Table 10 under Multi-Agent PIU-Only and Multi-Agent Full-Context.

F.3.3 Legal Problem-Instance and PIU-Distribution Statistics

Figure 6 reports the PIU count and PIU length for the 163 problem instances in the legal evaluation. It also reports participant count and the length of source material provided to participants for the 514 PIU distributions under the five PIU-distribution conditions in Table 2. For the first two measures, each point is one problem instance; for the latter two, each point is one PIU distribution.

Source-material length sums the material provided to each participant after applying the initial-local-observation token budget of 40,000 tokens. Repeated material is counted for each participant who receives it because it contributes separately to that participant's retrieval burden. PIU length is measured from the PIU text supplied in Multi-Agent PIU-Only. All lengths use the same `cl100k_base` tokenizer used for initial-local-observation budgeting. Table 12 reports the corresponding means; measures that depend on PIU distribution are first averaged within problem instance and then across problem instances.

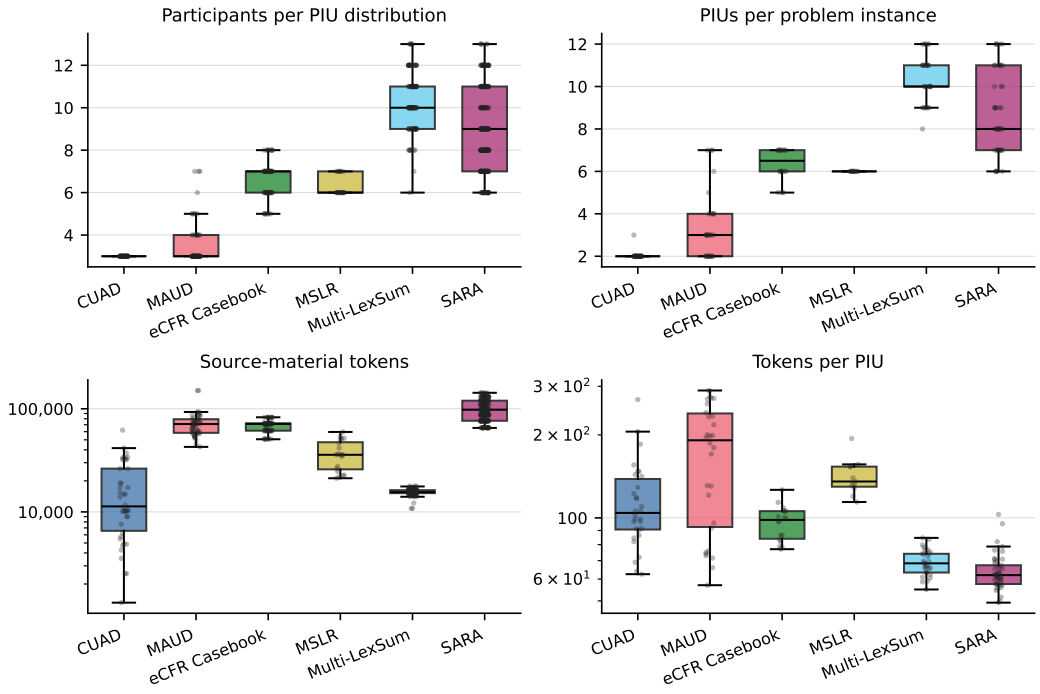


Figure 6: **Problem-instance and PIU-distribution statistics in the legal evaluation.** In the PIU-count and PIU-length panels, each point is one problem instance. In the participant-count and source-material-length panels, each point is one PIU distribution. Boxes show the median and interquartile range; token-length panels use logarithmic axes.

Table 12: **Mean problem-instance and PIU-distribution statistics for the legal evaluation.** Problem-instance measures use all 163 problem instances. PIU-distribution measures use 514 distributions. For measures that depend on PIU distribution, we first average within problem instance and then across problem instances. Source-material lengths reflect the material provided after applying the initial-local-observation token budget of 40,000 tokens.

Source corpus	Problem instances	PIU distributions	Participants/ distribution	PIUs	Source-material tok./ distribution	Source-material tok./ participant	PIU tok./ unit
CUAD	30	42	3.0	2.0	15,644.9	5,215.0	116.4
MAUD	30	53	3.5	3.4	70,859.8	20,759.0	174.8
eCFR Casebook	18	61	6.7	6.3	68,383.3	10,270.3	96.3
MSLR	11	20	6.3	6.0	35,948.0	5,688.9	141.7
Multi-LexSum	29	113	10.4	10.4	15,621.4	1,535.4	69.2
SARA	45	225	8.9	8.7	97,936.4	10,969.9	64.3
All source corpora	163	514	6.7	6.4	55,744.2	9,555.2	103.9

F.3.4 CUAD Problem-Instance Construction

Source corpus and retained problem instances. CUAD (Hendrycks et al., 2021) provides contract-level annotations across commercial clause categories. We do not use the original CUAD label as the final decision. A question such as "does this contract contain source-code escrow?" does not require multiple PIUs because one annotated span can determine the final decision. Instead, we use contract–label pairs with two to five positive annotated spans to construct finite-choice problem instances. Contract–label pairs with negative labels are excluded because the decision objective requires combining information rather than proving absence from a long contract.

The construction procedure produces 542 candidate problem instances. Of these, 539 admit at least one PIU distribution under the conditions in Table 2, and 538 are retained after the common legal

checks and the CUAD-specific checks below. The 538 retained problem instances, which are reported as the constructed CUAD problem instances in Table 3, are drawn from 289 distinct contracts. Most retained problem instances (514) contain two PIUs; the remaining 24 contain at least three. For the legal model comparison, we use the first retained problem instance from each previously unseen contract in the released order until 30 contracts are represented.

Decision objective and PIU construction. For each eligible contract–label pair, a construction model selects positive annotated spans and constructs a finite-choice problem instance whose final decision depends on their content.

Each candidate unit is one CUAD-annotated span. If a span contains several legal facts, they remain within one candidate unit; if two spans express similar legal content, they remain separate candidate units.

Using the selected candidate units, construction follows the common legal procedure in Appendix F.3 to construct the question, decision set, and reference decision. The alternatives use neutral, opaque identifiers, with their legal content placed in their definitions. The model-based check in Section 3.2 determines which candidate units form $\mathcal{I}_q^{\min}$. Because CUAD problem instances use a finite-choice decision format, each constructed problem instance must admit exactly one reference decision. Ambiguous questions such as "is this clause risky?" or "is this protection good enough?" are rejected unless the alternatives are specified so that the problem instance admits one reference decision.

Table 13 summarizes the CUAD label families used in construction.

Table 13: **CUAD label families used in problem-instance construction.** The decision objective specifies a legal profile supported by multiple annotated spans rather than the original CUAD label.

CUAD label family	Information in annotated spans	Decision objective
Source Code Escrow	deposit, release trigger, post-release use right, update duty, cost	source-code continuity protection strength
ROFR/ROFO/ROFN	covered opportunity, notice, exercise period, matching or negotiation right, expiration	transaction restriction strength
Insurance	coverage type, coverage limit, certificate, cancellation notice, additional insured, remedy	insurance burden or protection level
Audit Rights	recordkeeping, records review, facility or system inspection, notice, frequency, cost	audit burden level
License Grant	grant, field of use, territory, duration, sublicense, use restriction	license sufficiency for intended use
Minimum Commitment	amount, period, purchase or revenue metric, shortfall remedy, exceptions	commercial commitment materiality
Cap on Liability	cap amount or formula, excluded damages, carveouts, special warranty or IP rule	liability protection profile

Source material and PIU distribution. For CUAD, the underlying source material is the source contract, rather than only the annotated spans that determine the reference decision. The source contract is divided into contiguous units using legal section headings first, then paragraphs, sentences, and words as progressively weaker split points. A unit does not divide a PIU. Units containing PIUs are provided to the participants assigned those PIUs under the PIU distribution.

Units that do not contain a PIU are distributed based on their overlap with participant descriptions and PIUs, their proximity to PIUs in the source contract, their legal section, and the lengths of participants' initial local observations. A unit with overlap or proximity to a PIU is, when possible, provided to a participant not assigned that PIU.

Example participant descriptions. The CUAD label is not provided to participants. Table 14 provides example participant descriptions constructed under the common legal procedure.

CUAD-specific construction checks. In addition to the common legal checks in Appendix F.3, CUAD requires each candidate unit to be a positive annotated span copied exactly from the source

Table 14: **Example participant descriptions for CUAD problem instances.**

CUAD label family	Example participant descriptions
Source Code Escrow	deposit reviewer; release-condition reviewer; use-right reviewer; update/cost reviewer
ROFR/ROFO/ROFN	covered-opportunity reviewer; notice reviewer; exercise-period reviewer; post-expiration reviewer
Insurance	coverage-type reviewer; coverage-limit reviewer; certificate/notice reviewer; special-coverage reviewer
Audit Rights	records reviewer; access reviewer; operations-inspection reviewer; process/cost reviewer

contract with a recoverable source location. Each problem instance must contain at least two PIUs. The units used to divide the source contract must collectively cover it without dividing a PIU. A problem instance is rejected if it fails any of these checks.

F.3.5 MAUD Problem-Instance Construction

Source corpus and retained problem instances. MAUD (Wang et al., 2023b) pairs merger agreements with expert-curated annotations of their provisions. The source corpus contains 25,827 annotation rows from 153 agreements. Construction uses 100 agreements and produces 557 retained problem instances from 318 agreement–category pairs. Of those pairs, 239 produce two problem instances and 79 produce one. Among the 557 retained problem instances, 178 contain two PIUs and 379 contain at least three. The legal evaluation uses 30 fixed problem instances from distinct agreements, as described in Appendix F.3.2.

The source corpus organizes annotation rows under seven legal categories. The retained problem instances use the four categories in Table 15.

Table 15: **MAUD categories used in problem-instance construction.** Each problem instance uses provision excerpts from one agreement and one category.

MAUD category	Information in provision excerpts	Decision objective
Deal protection and related provisions	no-shop limits, fiduciary-out conditions, matching periods, superior-proposal conditions, and termination-fee provisions	board-process or termination-trigger profile
Conditions to closing	representation bring-down timing and standard, materiality treatment, and fundamental-representation treatment	closing-condition risk profile
Material adverse effect	base definition, carveouts, disproportionate-impact treatment, and pandemic or market exceptions	material-adverse-effect risk-allocation profile
Operating and efforts covenant	ordinary-course covenant, negative covenants, buyer consent rights, regulatory-efforts language, and emergency exceptions	pre-closing control profile

Decision objective and PIU construction. Many MAUD row-level questions can be answered from one provision excerpt. We therefore use the annotated questions and excerpts to construct a finite-choice problem instance whose final decision depends on multiple provisions from the same agreement and category. The alternatives describe the combined legal profile implied by those provisions rather than reproduce the MAUD row-level annotations.

A MAUD row may contain a curated provision excerpt rather than a complete agreement section, and some rows join passages separated by omitted text. Construction removes annotation artifacts, divides such rows at omitted-text markers, and does not reconstruct omitted text. Each resulting provision excerpt is a candidate unit. Construction then locates the candidate unit in the agreement, using exact matching when possible and normalization when necessary. The model-based check in Section 3.2 determines which candidate units form $\mathcal{I}_q^{\min}$. The MAUD row-level annotation is used to organize candidate units but is not itself a PIU.

Source material and PIU distribution. For MAUD, the underlying source material is the merger agreement. The agreement is divided into contiguous units without dividing a PIU. Units containing PIUs are provided to the participants assigned those PIUs under the PIU distribution. The remaining units are distributed based on their overlap with participant descriptions and PIUs, their proximity to PIUs in the agreement, their legal section, and the lengths of participants' initial local observations. When an initial local observation is truncated under the applicable resource budget, every PIU assigned to that participant remains present.

MAUD-specific construction checks. In addition to the common legal checks in Appendix F.3, MAUD requires every PIU to be a provision excerpt from a source annotation row and to have a recoverable location in the agreement. Each problem instance must contain at least two PIUs from one agreement and one retained category. The units used to divide the agreement must preserve every PIU. A problem instance is rejected if it fails any of these checks.

F.3.6 eCFR Casebook Problem-Instance Construction

Source corpus and retained problem instances. eCFR Casebook uses regulatory text from 12 C.F.R. Parts 1 and 3 together with controlled transaction memos (Office of the Federal Register, 2026b;c;a). The regulatory text covers bank investment securities and related capital requirements. Each transaction memo states the facts needed to apply one or more of these provisions. Construction produces 20 candidate problem instances. Eighteen are retained after the common legal checks and the eCFR Casebook-specific checks below. The legal evaluation uses all 18 retained problem instances.

Decision objective and PIU construction. Each problem instance asks for a short written compliance conclusion about one transaction memo. The final decision must state the controlling regulatory status or rule, the relevant transaction facts, any required threshold, exception, or calculation, and the resulting action. These requirements form the answer components evaluated in the final decision. The evaluator for free-form decisions in Appendix I.1.2 compares the final decision with the reference decision and these answer components.

Construction selects regulatory provisions whose application depends on more than one condition. Candidate units consist of regulatory text copied from the eCFR and transaction facts copied from the memo. The model-based check in Section 3.2 determines which candidate units form $\mathcal{I}_q^{\min}$. The retained problem instances contain five to seven PIUs.

Source material and PIU distribution. For eCFR Casebook, the underlying source materials are the eCFR provisions and the transaction memo. These materials are divided into long excerpts without dividing a PIU. Each excerpt containing a PIU is provided to a participant assigned that PIU under the PIU distribution, and PIUs assigned to other participants are removed from that participant's source material. Each participant's source material has a target length of 12,000 tokens; across the retained problem instances, the resulting lengths range from 9,846 to 11,830 tokens.

eCFR Casebook-specific construction checks. In addition to the common legal checks in Appendix F.3, eCFR Casebook requires every PIU to occur in the regulatory text or transaction memo. Construction checks that each PIU appears in the initial local observation of every participant assigned that PIU and not in the initial local observations of participants not assigned that PIU. A retained problem instance must also pass the model-based solvability check with all PIUs centralized. A problem instance is rejected if it fails any of these checks.

F.3.7 MSLR Problem-Instance Construction

Source corpus and retained problem instances. MSLR (Yu et al., 2027) provides anonymized Chinese administrative-enforcement case files. The source corpus used in construction contains 1,389 case files, of which 613 identify one target party, contain the required source material, and state a final judgment. Construction selects 100 candidate problem instances. After the common legal checks and the MSLR-specific checks below, 11 problem instances are retained. The legal evaluation uses all 11 retained problem instances.

Decision objective and PIU construction. Each problem instance asks for a short written enforcement conclusion for one target party. The reference decision combines the material nonpublic information and its timing, the basis for the party's knowledge or access, the party's trading conduct, the legal characterization, and the final administrative handling. The final judgment supplies the final administrative handling. These requirements form the answer components evaluated in the final decision. The evaluator for free-form decisions in Appendix I.1.2 compares the final decision with the reference decision and these answer components.

Construction identifies six exact, nonoverlapping excerpts from the case file. The excerpts cover the event and timing, knowledge or access, trading conduct, legal characterization, and final handling. Each candidate unit retains its location in the case file. We apply the PIU construction requirements in Section 3.2; every retained MSLR problem instance contains six PIUs.

Source material and PIU distribution. For MSLR, the underlying source material is the complete anonymized case file. Each participant receives the case file with the exact excerpts corresponding to the PIUs assigned to that participant. Passages corresponding to PIUs not assigned to that participant are removed before the source material is provided.

MSLR-specific construction checks. In addition to the common legal checks in Appendix F.3, MSLR requires each problem instance to identify one target party and to contain a final judgment. Every candidate unit must be copied exactly from the case file, have a recoverable source location, and not overlap another candidate unit. A retained problem instance must also pass the model-based solvability check with all PIUs centralized. A problem instance is rejected if it fails any of these checks.

F.3.8 Multi-LexSum Problem-Instance Construction

Source corpus and retained problem instances. Multi-LexSum (Shen et al., 2022) contains 4,539 civil-litigation collections. Of these, 3,086 contain between three and 18 source documents and have a Multi-LexSum summary within the length range used in construction. A model-based check of source grounding retains 48 candidate problem instances, and construction selects 30 for the remaining checks. After the common legal checks and the Multi-LexSum-specific checks below, 29 problem instances are retained. The legal evaluation uses all 29 retained problem instances.

Decision objective and PIU construction. Each problem instance asks for a written procedural or legal conclusion that requires two to four interdependent conclusions from the documents in one litigation collection. The question and reference decision are based on the original documents rather than the Multi-LexSum summary. Construction removes from the question the names and source-specific identifiers that would identify the litigation collection. The evaluator for free-form decisions in Appendix I.1.2 compares the final decision with the reference decision and the required answer components.

Construction identifies decision-relevant information in the original documents and states each candidate unit concisely. Each candidate unit records the sentence identifiers in the original documents that support it. Each candidate unit must distinguish the reference decision from at least one alternative considered during construction. We apply the PIU construction requirements in Section 3.2.

A model-based check compares each candidate unit with the supporting sentences and requires the original documents to support the complete reference decision. Each retained problem instance contains eight to twelve PIUs.

Source material and PIU distribution. For Multi-LexSum, the underlying source materials are the original documents in one litigation collection. The source material provided to a participant contains the original-document sentences supporting each PIU assigned to that participant, together with additional passages from the same litigation collection. Units containing PIUs and units containing the additional passages are distributed without dividing a PIU.

Multi-LexSum-specific construction checks. In addition to the common legal checks in Appendix F.3, Multi-LexSum requires the question to admit a determinate procedural or legal conclusion

without revealing that conclusion or the identity of the litigation collection. The candidate units must use at least three original-document excerpts, with at least two candidate units supported by each excerpt. Every candidate unit and the reference decision must pass the model-based check of source grounding. A retained problem instance must also pass the model-based solvability check with all PIUs centralized. A problem instance is rejected if it fails any of these checks.

Sensitivity to the ordering used for Pairwise PIU overlap. The PIU-distribution conditions in Table 2 specify which participants initially observe each PIU, but they do not prescribe an ordering based on similarity between PIUs. We therefore conduct an exploratory analysis using two additional Pairwise PIU overlap distributions for each of the 29 retained Multi-LexSum problem instances. One uses a seeded random ordering. The other uses an ordering that places PIUs with similar embedding representations next to one another.

We represent each PIU using Cohere Embed v4 (Cohere, 2025), with the clustering input type and 1,536-dimensional float embeddings, and normalize each representation. For a PIU ordering π, we compute

$$\frac{1}{|\mathcal{I}_q^{\min}| - 1} \sum_{j=1}^{|\mathcal{I}_q^{\min}|-1} \cos\big(z(i_{\pi_j}), z(i_{\pi_{j+1}})\big) ,$$

where $z(\cdot)$ maps a PIU to its embedding representation. We find the ordering that maximizes this quantity using exact dynamic programming with deterministic tie breaking. The seeded comparison uses a deterministic random permutation.

For either ordering, participant $j < |\mathcal{I}_q^{\min}|$ initially observes i_{π_j} and $i_{\pi_{j+1}}$, and the final participant initially observes $i_{\pi_{|\mathcal{I}_q^{\min}|}}$. The two distributions therefore have the same problem instance, pivotal information, number of participants, total number of PIU assignments, and number of PIUs in each participant's initial local observation. They differ only in which PIUs occupy these positions.

We evaluate both distributions in Multi-Agent PIU-Only using Qwen3-235B-A22B, Coordinator-selected speaker, 15 participant responses, and the resource budgets used in the legal model comparison. We apply the evaluation procedure for free-form decisions in Appendix I.1.2 using GPT-5.4 as the model judge. Table 16 reports the paired results.

Table 16: **Sensitivity to the ordering used to construct Pairwise PIU overlap distributions for Multi-LexSum.** Values average over the same 29 problem instances. Smaller ± terms are standard errors across problem instances.

Ordering	Mean adjacent cosine similarity	Episode success (%)	LLM-judged score (%)	Answer-component coverage (%)
Seeded random	.449	13.8 ± 6.4	65.2 ± 4.3	80.7 ± 3.4
Maximum similarity	.598	27.6 ± 8.3	71.9 ± 3.5	87.4 ± 2.8

The maximum-similarity ordering increases the intended construction measurement: mean adjacent cosine similarity increases from .449 to .598. Its episode success rate, LLM-judged score, and answer-component coverage are descriptively higher than under the seeded random ordering. The paired difference in the LLM-judged score is 6.7 percentage points, with a problem-instance-bootstrap 95% confidence interval of $[-3.1, 16.3]$ percentage points. Thus, this exploratory analysis does not establish a difference between the two PIU distributions. We report it to document a construction based on PIU similarity, not to rank PIU-distribution procedures or select an allocation using evaluation outcomes.

Sensitivity to the assignment of participant descriptions.

Following Step 5 of the common legal construction procedure (Appendix F.3), each participant receives a participant description constructed from the broad subject or function of its assigned PIU. The description must not disclose the PIU's specific content, values, conclusions, or effect on the final decision. The 301 participant descriptions used across the 29 retained Multi-LexSum problem instances pass the model-based check specified in that procedure.

We construct two versions of each problem instance. In one version, each participant description is attached to the participant assigned the PIU from which the description was constructed. In the paired version, the same participant descriptions are permuted across participants. The permutation preserves every participant description, including repeated descriptions, while ensuring that the description attached to every participant changes. Apart from which participant description is attached to each participant, the paired versions have the same problem instance, pivotal information, PIU distribution, initial local observations, communication structure, and resource budgets.

We evaluate both versions in Multi-Agent PIU-Only using Qwen3-235B-A22B, Coordinator-selected speaker, 15 participant responses, and the resource budgets used in the legal model comparison. We apply the evaluation procedure for free-form decisions in Appendix I.1.2 using GPT-5.4 as the model judge. Table 17 reports the paired results.

Table 17: **Sensitivity to the assignment of participant descriptions for Multi-LexSum.** Values average over the same 29 problem instances. Smaller $\pm$ terms are standard errors across problem instances.

Participant descriptions	Episode success rate (%)	LLM-judged score (%)	Answer-component coverage (%)
Constructed from participants' assigned PIUs	24.1 $\pm$ 8.1	77.6 $\pm$ 1.9	84.8 $\pm$ 2.4
Same descriptions permuted across participants	0.0 $\pm$ 0.0	46.9 $\pm$ 4.8	67.2 $\pm$ 3.6

Permuting the participant descriptions decreases the mean LLM-judged score by 30.7 percentage points. The LLM-judged score is lower after permutation for 24 problem instances, higher for three, and unchanged for two; a two-sided exact sign test gives $p = 4.9 \times 10^{-5}$. Every PIU appears in the communication transcript in all 29 problem instances when participant descriptions are constructed from participants' assigned PIUs, compared with 18 of 29 problem instances when the same descriptions are permuted across participants. Thus, under Coordinator-selected speaker, performance depends on whether the participant descriptions correspond to the PIUs assigned to those participants. This analysis isolates the assignment of participant descriptions; it does not compare communication structures or procedures for constructing participant descriptions.

F.3.9 SARA Problem-Instance Construction

Source corpus and retained problem instances. SARA (Holzenberger et al., 2020) provides 100 tax cases with case facts, statutory text, and calculation and reasoning steps. Of these source cases, 83 yield a candidate problem instance and 63 satisfy the source and PIU requirements described below. Forty-five pass the repeated model-based solvability checks with all PIUs centralized and are retained. The retained problem instances contain 393 PIUs, with six to twelve PIUs per problem instance (mean 8.7), and four to eight answer components (mean 5.5). The legal evaluation uses all 45 retained problem instances.

Decision objective and PIU construction. Construction executes the supplied calculation and reasoning steps and selects an intermediate conclusion that depends on multiple steps. The selected conclusion defines a numeric decision objective, such as a deduction, taxable income, or a tax amount, rather than requiring every problem instance to recover the original total-tax result. Construction traces the steps backward from that conclusion to the facts and rules that determine it and fixes the reference decision before writing the question.

Each candidate unit states one fact or rule needed for at least one of the selected calculation and reasoning steps. Source-stated facts and statutory rules retain their source locations. When a required relation appears only in the supplied calculation and reasoning steps, construction states that relation in ordinary legal language and records its basis. A calculated intermediate result is not repeated as a candidate unit when the supplied facts and rules already determine it. The model-based sufficiency and necessity check determines which candidate units form $\mathcal{I}_q^{\min}$.

The final decision must state the requested numeric result and a concise derivation. The evaluator described in Appendix I.1.2 compares the requested numeric result with the reference value and evaluates the answer components that form the derivation.

Source material and PIU distribution. For SARA, the supplied calculation and reasoning steps are used during problem-instance construction but are not provided to participants. Each participant's source material contains the fact or rule stated by each PIU assigned to that participant, together with the case facts and statutory text. Information stating a PIU not assigned to that participant is removed before the source material is provided.

SARA-specific construction checks. In addition to the common legal checks in Appendix F.3, SARA requires every candidate unit to have a recorded basis in the case facts, statutory text, or supplied calculation and reasoning steps. A fact or rule must resolve each condition needed to determine the selected conclusion, including a condition that rules out an exception. We apply the PIU construction requirements in Section 3.2. Each problem instance then undergoes five model-based solvability checks with all PIUs centralized and is retained when at least four succeed. A problem instance is rejected if it fails any of these checks.

F.4 FINANCE

We construct finance problem instances from FinDVer (Zhao et al., 2024) and FinLongDocQA (Wang et al., 2026). Each source corpus corresponds to one problem family, which we call the FinDVer and FinLongDocQA problem families, respectively. The procedure below reports the choices specific to these source corpora and problem families. Table 18 summarizes the finance problem families and their source corpora.

Table 18: **Finance problem families and source corpora.**

Source corpus	Underlying source materials	Reference decision	Problem family
FinDVer	SEC filings with source questions and annotated contexts	Written filing-review answer	Filing-review reasoning
FinLongDocQA	Company-year annual reports with source questions and answers	Written annual-report answer	Annual-report reasoning

F.4.1 COMMON FINANCE CONSTRUCTION PROCEDURE

Step 1. Select the underlying source materials. A problem instance uses exactly one U.S. Securities and Exchange Commission (SEC) filing or company-year annual report. Native questions, answers, labels, source-text references, and available calculations, code, and reasoning may be used during construction but do not form part of the initial common observation.

Step 2. Divide the underlying source materials. The filing or report is divided into complete paragraphs and tables while preserving their source locations. Adjacent fragments may be combined only when a label in one fragment requires a value in the other to be complete. A table is not divided into individual cells or values.

Step 3. Identify candidate units. For each source question, a model-based source-grounding check identifies every paragraph, table, or combination of an adjacent label and value that contributes a report-specific value, definition, period, unit, entity scope, condition, exception, or contradiction needed to reconstruct the source annotation. Before problem-instance construction, the procedure records whether candidate units are exact duplicates, whether one fully contains another, whether one can be derived from another, and whether they partially overlap. A candidate unit is removed only when a retained candidate unit contains its entire contribution; partial overlap alone does not support removal.

Step 4. Specify the decision objective and format. Construction combines one or more source questions and at least two candidate units into a problem instance with a free-form decision format. It fixes the reference decision, answer components, the candidate units supporting each answer component, and the evaluation criteria for values, units, periods, signs, scopes, and conclusions.

Step 5. Identify the pivotal information. Each candidate unit is one paragraph, complete table, or combined adjacent label and value. We apply the PIU construction and removal checks in Section 3.2, using the answer components to assess completeness, correctness, and source grounding. A redundant candidate unit is removed before the checks are repeated.

Step 6. Construct participant descriptions. Each participant receives a source-corpus-specific reviewer description. These descriptions identify the participant as a reviewer of a filing or annual report and include a participant number. Construction records broad information features for PIUs, but participant descriptions and the participant directory do not disclose PIU values, calculations, conclusions, or their effect on the reference decision.

Step 7. Define the PIU distribution and distribute source material. Each finance problem instance uses the One PIU per participant condition in the two team evaluation conditions. The additional PIU distributions reported in Table 19 use the other conditions from Table 2. After each PIU distribution is fixed, the underlying source materials are distributed across participants. The source material provided to each participant is checked for information that contains or supports reconstruction of a PIU assigned to another participant. Such material is removed while every PIU assigned to that participant is retained.

This source-material check concerns meaning rather than exact wording. Exact repetition of a PIU assigned to another participant is sufficient for removal but is not required: a passage or collection of passages is also removed when it supports reconstruction of the report-specific value, period, unit, entity scope, condition, exception, or conclusion contributed by that PIU. Partial topical overlap alone does not support removal. After revising the source material, the procedure repeats the participant-level check while preserving every PIU assigned to that participant.

Step 8. Complete problem-instance construction. A problem instance is retained only after the specified participants, pivotal information, PIU distribution, reference decision, and distributed source material pass the applicable checks.

F.4.2 Problem Instances and PIU Distributions in the Finance Evaluation

The construction pipeline produces 389 FinDVer problem instances and 249 FinLongDocQA problem instances. Following the same fixed-subset convention used for CUAD and MAUD, the finance evaluation uses 30 problem instances from each source corpus under the three evaluation conditions. The FinDVer subset is selected deterministically in released order within source-material strata, with 20 Form 10-K filings and 10 Form 10-Q filings. The FinLongDocQA subset is selected deterministically in released order within year strata, with 9 reports from 2022, 10 from 2023, and 11 from 2024. For each language-model policy, these conditions contain 90 FinDVer episodes and 90 FinLongDocQA episodes. Each problem instance uses the One PIU per participant condition in the two team conditions.

We also construct PIU distributions under the other conditions in Table 2. For each distribution, the initial local observation of each participant is checked for information that contains or supports reconstruction of a PIU assigned to another participant. Table 19 reports the number of evaluated problem instances admitting each PIU-distribution condition.

Table 19: **Finance problem-instance counts by PIU-distribution condition.**

PIU-distribution condition	FinDVer	FinLongDocQA
One PIU per participant	30	30
Pairwise PIU overlap	26	23
Extra participant without a PIU	30	28
One common PIU	28	23
All-but-one PIU concentration	30	29
PIU distributions, total	144	133

F.4.3 Finance Problem-Instance and PIU-Distribution Statistics

Source-corpus and problem-instance counts. Each problem instance uses underlying source materials from one filing or company-year report. Underlying source materials from different filings or reports are not combined. Table 20 separates the available source-material pools from the processed and retained construction outputs.

Table 20: **Finance source-corpus and problem-instance counts.**

Statistic	FinDVer	FinLongDocQA
Source questions in the complete release	2,400	7,527
Source questions considered during construction	700	7,527
Available underlying source materials	255 filings	1,329 company-years
Constructed problem instances	389	249

The finance evaluation uses a fixed subset of 30 constructed problem instances from each source corpus, as described in Appendix F.4.2.

Problem-instance and PIU-distribution statistics. Each problem instance requires a source-grounded written final decision. Source questions are used only during construction; the constructed question does not provide the corpus-provided answer or its label, explanation, or calculation. It also omits the PIU identity and hidden evaluation annotations. The fixed FinDVer evaluation set records 231 answer components across 70 source questions and uses 20 Form 10-K filings and 10 Form 10-Q filings. The fixed FinLongDocQA evaluation set records 64 answer components across 63 source questions and uses 9 company-year reports from 2022, 10 from 2023, and 11 from 2024. Across all 60 evaluated problem instances, every referenced source-material unit and PIU has a recorded source location, and no participant initially receives every PIU.

Figure 7 reports the PIU count and PIU length for the 60 problem instances in the finance evaluation. It also reports participant count and the length of source material provided to participants for the 277 PIU distributions under the five PIU-distribution conditions in Table 2. For the PIU-count and PIU-length panels, each point is one problem instance; for the participant-count and source-material-length panels, each point is one PIU distribution.

Source-material length sums the material provided to each participant. Repeated material is counted for each participant who receives it because it contributes separately to that participant's initial local observation. The source-material lengths are measured after the participant-level leakage-removal step and after applying the 40,000-token initial-local-observation cap. PIU length is measured from the PIU text supplied in Multi-Agent PIU-Only. All lengths use the same `cl100k_base` tokenizer used for the legal statistics.

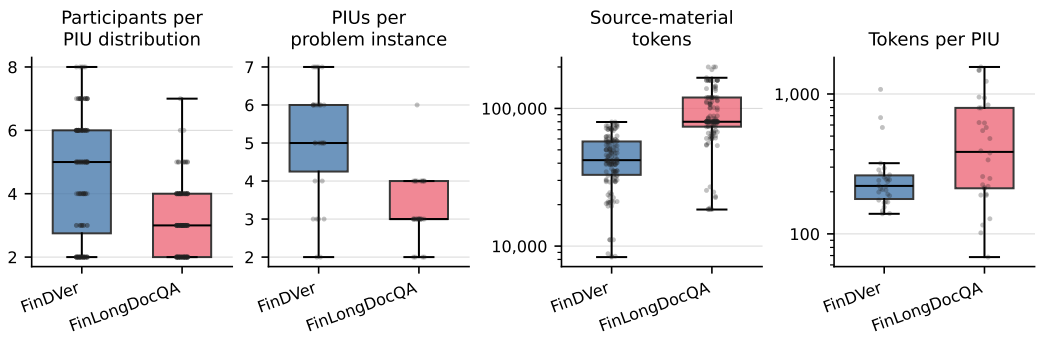


Figure 7: **Problem-instance and PIU-distribution statistics in the finance evaluation.** In the PIU-count and PIU-length panels, each point is one problem instance. In the participant-count and source-material-length panels, each point is one PIU distribution. Boxes show the median and interquartile range; token-length panels use logarithmic axes.

Table 21: **Mean problem-instance and PIU-distribution statistics for the finance evaluation.** Problem-instance measures use the 60 fixed evaluation problem instances. PIU-distribution measures use 277 distributions. For measures that depend on PIU distribution, we first average within problem instance and then across problem instances. Source-material lengths reflect the material provided after participant-level leakage removal and the 40,000-token initial-local-observation cap.

Source corpus	Problem instances	PIU distributions	Participants/ distribution	PIUs	Source-material tok./ distribution	Source-material tok./ participant	PIU tok./ unit
FinDVer	30	144	4.5	5.1	44,038.1	11,887.5	271.3
FinLongDocQA	30	133	3.1	3.3	91,711.8	31,168.6	546.0
All source corpora	60	277	3.8	4.2	67,874.9	21,528.1	408.6

Evaluation criteria. In both finance source corpora, a PIU may support several answer components, and one answer component may require several PIUs. Written final decisions are graded using problem-specific criteria for values, units, periods, signs, scopes, calculations, and qualifying conclusions.

F.4.4 FinDVer Problem-Instance Construction

Source corpus and constructed problem instances. FinDVer provides entailment, numeric, and knowledge-oriented source questions linked to SEC filings. Construction uses the FinDVer source corpus's `testmini` split and retains the source location for each Form 10-K or Form 10-Q filing. The procedure constructs 389 problem instances. The finance evaluation uses the fixed 30-instance subset described in Appendix F.4.2.

Decision objective and PIU construction. Construction does not use the original FinDVer question as the decision objective. Instead, it constructs a free-response filing-review task that asks for related filing-specific findings, calculations, comparisons, or classifications without providing the source values or expected conclusions. The reference decision is a source-grounded written final decision. Hidden evaluation annotations record the answer components, the links between answer components and PIUs, and problem-specific evaluation criteria. For numeric answer components, the criteria record the value, unit, sign, period, and tolerance. For textual and classification answer components, the criteria record the accepted conclusion and required scope.

Construction reconstructs each source annotation from complete filing paragraphs or tables and records overlap, containment, and derived equivalence among the candidate units before forming one filing-review problem instance. The task-generation step selects complete candidate units. It prefers a broad coherent cluster when distinct candidate units concern the same company, period, transaction, segment, liquidity topic, accounting policy, operating trend, or filing-review objective. Candidate units concerning unrelated information are excluded, and a candidate unit may be retained with another partially overlapping unit only when the task and evaluation criteria use a distinct contribution from each. Every PIU must be needed for at least one answer component, and every selected PIU is linked to the component or reasoning step it supports.

Source material and PIU distribution. Complete filing material is distributed across participants. Surrounding passages or tables that directly repeat, tabulate, or clearly paraphrase a PIU assigned to another participant are removed. BM25 lexical scores provide candidate pointers for review, after which a model-based check inspects the complete packet and determines whether any context contains or supports reconstruction of such a PIU. The procedure performs one removal pass without redistributing the remaining source material. Every assigned PIU remains in the corresponding participant's initial local observation.

Participant descriptions. Participant descriptions are finance-specific reviewer labels. They may describe a broad filing or report topic, but they do not disclose PIU values, calculations, conclusions, or their effect on the reference decision.

FinDVer-specific construction checks. In addition to the common finance checks in Appendix F.4.1, a model-based source-grounding check verifies the calculations, comparisons, scopes, and links between PIUs and answer components. Additional checks require the pivotal information to support a complete and correct final decision, require public information without PIUs to be insufficient, and reject candidate PIUs that are redundant under the answer-component links. A problem instance is rejected if it fails these checks or the common finance checks.

F.4.5 FinLongDocQA Problem-Instance Construction

Source corpus and constructed problem instances. Each FinLongDocQA source question provides its answer type and annotated page numbers. Its reasoning and code are not used in construction. The procedure constructs 249 problem instances. The finance evaluation uses the fixed 30-instance subset described in Appendix F.4.2.

Decision objective and PIU construction. Construction does not use a source question verbatim as the decision objective. It constructs a free-response annual-report task that asks for a written memo, calculation, explanation, or numbered review. The task may use one source question or combine several source questions from the same company-year report. When the selected source questions do not share one analytical purpose, construction asks a concise multi-part review rather than inventing a causal relationship or comparison. It records a reference decision, answer components, links between PIUs and answer components, evaluation criteria, and errors that make a final decision incorrect. Written final decisions are evaluated using the recorded problem-specific evaluation criteria.

For each source question, construction uses the annotated pages to identify every paragraph, table, or combined adjacent label and value needed for the answer's values, definitions, periods, units, scope, and exceptions. These candidate units then follow the division and overlap requirements in Appendix F.4.1. Each selected evidence unit is one complete mapped block. A block is not duplicated into multiple PIUs for separate values, rows, or claims. Every selected PIU must support at least one report-specific claim, and removing that PIU must leave at least one linked claim not fully grounded even when all other PIUs remain available. A PIU may support several claims, and one claim may require several PIUs.

Source material and PIU distribution. Each PIU is provided to the participant assigned that PIU. Other report passages and tables are provided to the participant with the shortest initial local observation, using distance from the page containing that participant's PIU to resolve a tie. A model-based check determines whether one passage or table, or several in combination, supports reconstruction of a PIU assigned to another participant. The source material is revised until every PIU is preserved and such information is removed.

When unrelated rows of a large table independently state a PIU assigned to another participant, the table is restricted to the rows supporting the answer component associated with the PIU assigned to that participant.

Participant descriptions. Participant descriptions are finance-specific reviewer labels. They may describe a broad filing or report topic, such as liquidity and capital resources or deferred tax assets and liabilities, but they do not disclose PIU values, calculations, conclusions, or their effect on the reference decision.

FinLongDocQA-specific construction checks. In addition to the common finance checks in Appendix F.4.1, a problem instance is rejected if a participant does not receive a PIU assigned to that participant or if source material still supports reconstruction of a PIU assigned to another participant. Additional checks require the reference answer to satisfy every recorded claim, preserve material units and periods, and use no outside facts. A problem instance is rejected if it fails these checks or the common finance checks.

F.5 Medical

We construct medical problem instances from MLCR (Wisedocs, 2026) and EvidenceBench (Wang et al., 2025). Each source corpus corresponds to one problem family: MLCR medical-record

reasoning and EvidenceBench biomedical-evidence reasoning. The procedure below reports the choices specific to these source corpora and problem families. Table 22 summarizes the medical problem families and their source corpora.

Table 22: **Medical problem families and source corpora.**

Source corpus	Underlying source materials	Reference decision	Problem family
MLCR	Case-level visit-summary documents with source questions and answers	Written medical-record review	Medical-record reasoning
EvidenceBench	Biomedical-paper sentence pools with hypotheses and evidence annotations	Written biomedical-evidence answer	Biomedical-evidence reasoning

F.5.1 Common Medical Construction Procedure

Step 1. Select the underlying source materials. A problem instance uses source materials from one medical source record. For EvidenceBench, this is one biomedical paper represented as an ordered sentence pool. For MLCR, this is one case whose source material may contain many visit summary documents. Source materials from different papers or cases are not combined.

Step 2. Divide the underlying source materials. EvidenceBench source materials are divided into paper sentences, preserving sentence indices and sentence types. MLCR source materials are divided into exact sentence-like atoms and paragraph spans with raw character offsets, document identifiers, and checksums. Adjacent atoms or sentences may be combined only when needed to preserve a patient, date, encounter, negation, measurement unit, uncertainty, abbreviation, comparison, or clinical scope.

Step 3. Identify candidate units. Construction uses source annotations to identify candidate evidence units. EvidenceBench starts from annotated optimal-evidence anchors. MLCR first maps source question-answer pairs to exact source atoms in the case record. In both families, candidate units are source-faithful evidence spans rather than final PIUs: they become PIUs only after task construction and the boundary checks show that they are answer-relevant and necessary.

Step 4. Specify the decision objective and format. Construction creates a free-response medical task with a written reference decision and a hidden claim-level rubric. The public task does not reveal PIU identities, candidate-unit identifiers, source annotation labels, hidden rubric claims, reference decisions, or expected values and conclusions. Required claims record medical scope, patient or study population, dates or periods, comparisons, outcomes, direction, uncertainty, units, tolerances, and causal or associational strength when applicable.

Step 5. Identify the pivotal information. We apply the PIU construction requirements in Section 3.2. The pivotal information must support a complete and grounded answer to the public task, while a missing PIU must leave at least one required claim unsupported or incorrect. A model-based solver and separate judge check the all-PIU condition, a no-PIU condition, and leave-one-out conditions. The solver sees only the public task and displayed source material; the judge uses the hidden rubric to evaluate claim correctness and grounding. Correct guesses that are not grounded in displayed material do not satisfy the grounding check.

Step 6. Define the PIU distribution and distribute source material. Each medical problem instance uses the One PIU per participant condition in the two team evaluation conditions. The additional PIU distributions reported in Table 23 use the other conditions from Table 2. After each PIU distribution is fixed, source material is distributed across participants. The material provided to each participant is checked for information that contains or supports reconstruction of a PIU assigned to another participant. Such material is removed while every PIU assigned to that participant is retained.

Step 7. Complete problem-instance construction. A problem instance is retained only after the public task, pivotal information, PIU distribution, reference decision, hidden rubric, and distributed source material pass the applicable checks.

F.5.2 Problem Instances and PIU Distributions in the Medical Evaluation

The final medical evaluation uses 29 MLCR problem instances and 45 EvidenceBench problem instances under the three evaluation conditions. For each language-model policy, these conditions contain 87 MLCR episodes and 135 EvidenceBench episodes.

We also construct PIU distributions under the other conditions in Table 2. For each distribution, the initial local observation of each participant is checked for information that contains or supports reconstruction of a PIU assigned to another participant. Table 23 reports the number of evaluated problem instances admitting each PIU-distribution condition.

Table 23: **Medical problem-instance counts by PIU-distribution condition.**

PIU-distribution condition	MLCR	EvidenceBench
One PIU per participant	29	45
Pairwise PIU overlap	19	36
Extra participant without a PIU	28	45
One common PIU	19	36
All-but-one PIU concentration	19	45
PIU distributions, total	114	207

F.5.3 Medical Problem-Instance and PIU-Distribution Statistics

Source-corpus and problem-instance counts. Each problem instance uses one EvidenceBench paper or one MLCR case. For MLCR, the case may contain many underlying visit-summary source documents, but cases are not combined. Table 24 separates the available source-material pools from the fixed evaluation problem instances.

Table 24: **Medical source-corpus and problem-instance counts.**

Statistic	MLCR	EvidenceBench
Source questions or hypotheses in the construction pool	205 source questions	293 test-split hypotheses
Available underlying source materials	1,071 visit-summary documents	293 biomedical-paper sentence pools
Evaluated problem instances	29	45
Evaluated PIU distributions	114	207

Problem-instance and PIU-distribution statistics. Each medical problem instance requires a source-grounded written final decision. The constructed public question omits the source-provided answer, expected conclusion, PIU identities, and hidden claim rubric. Across the 74 evaluated problem instances, every PIU has a source location, and no participant initially receives every PIU in the team evaluation conditions.

Figure 8 reports the PIU count and PIU length for the 74 problem instances in the final medical evaluation. It also reports participant count and the length of source material provided to participants for the 321 PIU distributions under the five PIU-distribution conditions in Table 2. For the PIU-count and PIU-length panels, each point is one problem instance; for the participant-count and source-material-length panels, each point is one PIU distribution.

Source-material length sums the material provided to each participant. Repeated material is counted for each participant who receives it because it contributes separately to that participant's initial local

observation. The source-material lengths are measured after the participant-level leakage-removal step and after applying the 40,000-token initial-local-observation cap. PIU length is measured from the PIU text supplied in Multi-Agent PIU-Only. All lengths use the same `cl100k_base` tokenizer used for the legal statistics.

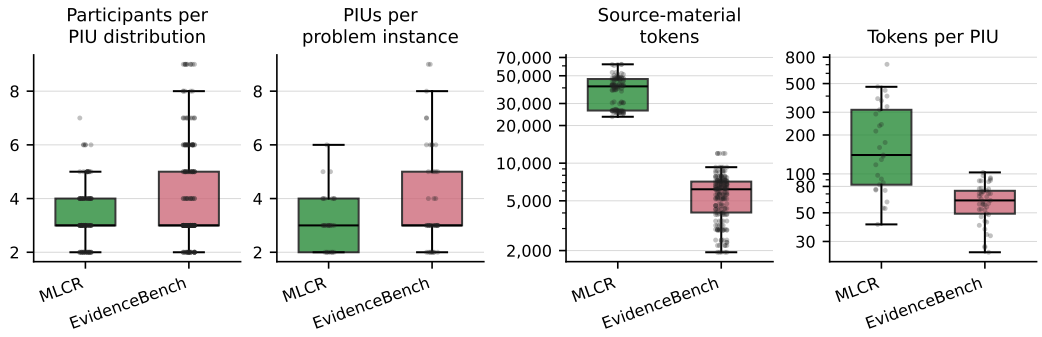


Figure 8: **Problem-instance and PIU-distribution statistics in the medical evaluation.** In the PIU-count and PIU-length panels, each point is one problem instance. In the participant-count and source-material-length panels, each point is one PIU distribution. Boxes show the median and interquartile range; token-length panels use logarithmic axes.

Table 25: **Mean problem-instance and PIU-distribution statistics for the medical evaluation.** Problem-instance measures use the 74 fixed evaluation problem instances. PIU-distribution measures use 321 distributions. For measures that depend on PIU distribution, we first average within problem instance and then across problem instances. Source-material lengths reflect the material provided after participant-level leakage removal and the 40,000-token initial-local-observation cap.

Source corpus	Problem instances	PIU distributions	Participants/ distribution	PIUs	Source-material tok./ distribution	Source-material tok./ participant	PIU tok./ unit
MLCR	29	114	3.2	3.2	38,119.9	13,080.7	208.5
EvidenceBench	45	207	4.0	4.0	5,678.0	1,727.4	62.7
All source corpora	74	321	3.7	3.7	18,391.7	6,176.7	119.8

Evaluation criteria. In both medical source corpora, a PIU may support several required claims, and one required claim may require several PIUs. Written final decisions are graded using problem-specific criteria for clinical scope, study or patient population, dates or periods, measurements, units, direction, uncertainty, comparison, and causal or associational strength.

F.5.4 MLCR Problem-Instance Construction

Source corpus and retained problem instances. MLCR provides long medical-record cases with source questions, reference answers, and visit-summary source documents (Wisedocs, 2026). The evaluated construction uses 29 problem instances. Because an MLCR case may contain many visit summaries, a single problem instance can draw source material from multiple source documents within the same case, but it never combines source material across cases.

Decision objective, reference decision, and evaluation criteria. Construction first maps each eligible source question-answer pair to exact source atoms that materially reconstruct the answer. It then groups one or more source question-answer pairs from the same case into a clinically coherent free-response task. When selected source questions do not share one clinical or claims-review purpose, construction uses a concise multi-part review rather than inventing a causal relationship or common theme. The public task asks for the medical-record review and does not reveal source question identifiers, candidate identifiers, PIU identifiers, or case-specific expected values and conclusions.

The hidden rubric records required claims, the selected evidence units that support each claim, and the criteria for clinical scope, dates, encounters, measurements, units, direction, negation, uncertainty,

and causal strength. The reference decision must satisfy all required claims using only the selected source evidence. Generic medical knowledge may connect observations but cannot replace missing patient-specific evidence.

Candidate-unit and PIU construction. MLCR source documents are divided into exact source atoms while preserving raw offsets and document identifiers. Each candidate unit is a contiguous span of atoms from one source document. Adjacent atoms are included only when needed to preserve patient, date, encounter, negation, measurement unit, uncertainty, or clinical scope. Non-adjacent evidence and repeated equivalents are kept separate rather than merged.

A model-based grounding check verifies that every required claim and every material statement in the reference decision are grounded in the selected exact source evidence, and that each selected unit makes a unique necessary contribution. Boundary checks then test the complete selected set, a no-PIU condition, and leave-one-out conditions. A selected candidate becomes a PIU only after these checks show that the full set is sufficient and that removing the candidate breaks at least one claim it supports.

Source material and PIU distribution. MLCR source material is distributed across participants under the PIU distribution. The participant assigned a PIU receives the canonical source span for that PIU. Additional source atoms from the same case are assigned to participants subject to the 40,000-token local-observation budget. Multi-Agent PIU-Only presents only the assigned PIU text. Multi-Agent Full-Context presents the participant's medical-record source material with neutral source-file structure. Single-Agent All-PIU presents all pivotal information. All three conditions use the same public task, and hidden source mappings, candidate identifiers, PIU identifiers, rubric mappings, and reference answers are not shown to decision makers.

MLCR-specific construction checks. For each participant and each PIU assigned to another participant, a model-based packet-leakage check determines whether the participant's complete full-context packet directly contains, duplicates, paraphrases, or permits reconstruction of that foreign PIU, including by combining multiple packet passages. Source material containing the participant's assigned PIU is not removed. If noncanonical source material reconstructs a foreign PIU, removable contexts are deleted and the packet is checked again. If the participant's own canonical source span itself reveals a foreign PIU, construction records this as canonical overlap and either merges ownership as allowed by the admissible allocation or rejects the allocation when strict one-each ownership is required. A problem instance or allocation is rejected if leakage remains without removable source material or the check does not converge.

F.5.5 EVIDENCEBENCH PROBLEM-INSTANCE CONSTRUCTION

Source corpus and retained problem instances. EvidenceBench provides biomedical hypotheses, full papers represented as ordered sentence pools, sentence types, and sentence-level annotations of the minimum evidence required for each hypothesis (Wang et al., 2025). We use the test split. Forty-five problem instances pass the construction checks, and all 45 are used in the medical evaluation.

We treat the sentences identified by these annotations as source locations used during construction rather than as PIUs. For every source record, construction records the instance and paper identifiers, original hypothesis, ordered sentence indices, sentence types, paper-sentence count, split, and source-file hash as hidden source annotations. Up to two neighboring sentences may be added only to restore an incomplete referent, comparison, abbreviation, or measurement context; the complete neighborhood is not automatically included in a candidate unit.

Decision objective, reference decision, and evaluation criteria. Construction turns the source hypothesis and grounded evidence annotations into a biomedical question, a reference decision, and separately evaluated answer components. The decision objective omits the annotated sentence indices, PIU identities, candidate-unit identifiers, reference decision, and hidden construction information. The hypothesis guides the question but is not necessarily reused verbatim: the question may instead ask for the population, exposure or intervention, comparator, outcome, direction, evidentiary strength, or required limitation established by the paper evidence. The reference decision uses only selected exact evidence and preserves medical scope and the source's causal or associational strength.

Answer components and candidate units are distinct. An answer component may require several candidate units that make complementary contributions, and one candidate unit may support several answer components. The hidden evaluation criteria therefore record the links between answer components and candidate units. For some answer components, every linked candidate unit is required; for others, any one of several linked candidate units is sufficient. The recorded links state which part of an answer component each candidate unit supports. Construction also records, for every candidate unit, whether it is retained and, if it is excluded, the reason for exclusion.

Candidate-unit construction and checks. Construction preserves a one-to-one relation between the sentences identified by the source annotations and candidate units. Each such sentence produces exactly one candidate unit, and distinct source sentences are not merged even when they jointly support the same answer component. The candidate unit contains the source sentence verbatim and may add only the minimal contiguous neighboring context needed to make it interpretable. It may not include another sentence identified by the source annotations as context. Construction checks complete one-to-one coverage of the source annotations, unique identifiers, contiguous and nonoverlapping sentence spans, exact agreement with the paper text, and absence of another sentence identified by the source annotations. Relations among candidate units are represented by their links to answer components rather than by merging their source locations.

A separate model-based check receives the decision objective, reference decision, hidden evaluation criteria, and the complete candidate-unit set, including excluded candidate units. For each retained candidate unit, it checks relevance to the decision objective, direct support, and necessity under the recorded links to answer components; for each excluded candidate unit, it checks that exclusion is justified. It also checks that the decision objective is natural rather than artificially expanded to accommodate unrelated evidence, that every answer component and the reference decision are source-grounded, and that population, comparison, outcome, direction, uncertainty, and causal scope are preserved. A failed check permits bounded revision of the decision objective, answer components, and links while the exact candidate units remain fixed. At least two annotated source sentences and two retained candidate units are required, but construction imposes no fixed upper bound.

PIU construction and checks. Construction then applies the model-based removal check described in Section 3.2, using a model and a separate model judge. The model receives only the decision objective, the paper excerpts available in the current check, and neutral evidence numbers. It does not receive candidate-unit, PIU, annotated-sentence, answer-component, or omitted-unit identifiers. The model judge receives the hidden evaluation criteria and reference decision and separately records whether each answer component is correct and grounded in the material provided to the model. A correct guess is not treated as grounded.

With all retained candidate units, every answer component must be correct and grounded, the final decision must be complete, and it must contain no critical error. For each removal required by Section 3.2, at least one answer component linked to the removed candidate unit must cease to be satisfied. The recorded links are also checked against the source references in the model's response: every linked unit is required when the answer component requires all of them, whereas at least one linked unit is required when any one is sufficient. After every check passes, each retained candidate unit forms one PIU; candidate units that jointly support one answer component remain separate PIUs.

Source material and PIU distribution. The retained construction uses One PIU per participant. Each source sentence containing a PIU is reserved for the participant assigned that PIU. Overlap between the source sentences containing distinct PIUs is rejected. Every remaining sentence from the EvidenceBench paper sentence pool is then assigned to exactly one participant, with seeded assignment balancing the estimated lengths of the initial local observations; the resulting division is exhaustive and disjoint. Multi-Agent PIU-Only presents only the exact text of the assigned PIU. Multi-Agent Full-Context presents the participant's paper sentences with neutral source identifiers. Single-Agent All-PIU presents all pivotal information. All three conditions use the same decision objective, and hidden candidate-unit identifiers, PIU identifiers, annotated source locations, links to answer components, reference source locations, and source-material check information are removed from the decision makers' observations.

EvidenceBench-specific construction checks. For each participant and each PIU assigned to another participant, a model-based check considers the participant's complete initial local observation rather than isolated lexical matches. The initial local observation contains a PIU assigned to another participant only when an exact duplicate, clear paraphrase, equivalent reported result, or combination of source passages supports that PIU's complete contribution at the same medical scope. Shared topics, partial overlap in population or outcome, a different measurement, a general mechanistic association, or a plausible inference requiring external knowledge do not suffice.

When such reconstruction is possible, the removable additional source passages that support the reconstruction are deleted, and every participant and PIU assigned to another participant are checked again. Source material containing the participant's assigned PIU is never deleted. If that material alone supports a PIU assigned to another participant, the problem instance is rejected because a candidate unit is incorrectly defined. The procedure permits at most three revisions and rejects the problem instance if the initial local observation still contains another participant's PIU, no removable source material exists, final-decision validation fails, or the checks do not converge.

F.6 Software Engineering

We construct software-engineering problem instances from SWE Atlas Codebase QnA (Raghavendra et al., 2026) and DeepSWE (Huang et al., 2026). Table 26 summarizes the source corpora, problem families, evaluation criteria, and evaluation scope. The two source corpora use separate construction procedures, described in the following subsections.

F.6.1 Source Corpora and Evaluation Scope

Table 26: **Software-engineering source corpus and evaluation status.**

Source corpus	Problem family	Evaluation criteria	Evaluated problem instances
SWE Atlas Codebase QnA	Multi-file and runtime code comprehension	Every answer component is complete and correct	32
DeepSWE	Long-horizon repository-level code repair	Official repository verifier passes	12

Shared construction conventions. Both source corpora fix a repository and base commit, identify exact source-grounded PIUs, assign each PIU to one participant in collaborative conditions, and withhold reference answers and hidden qualification records from the evaluated agents. Source-specific procedures determine how code, tests, runtime observations, and additional context are collected, grouped, distributed, and checked for leakage and format salience.

The SWE Atlas evaluation uses all 32 retained instances and eight language-model policies across the three evaluation conditions. For DeepSWE, we archive intermediate construction and allocation artifacts for 21 screened candidate tasks, then complete construction for a frozen 12-task cohort selected to preserve programming-language coverage and variation in original benchmark difficulty. These 12 tasks constitute both the constructed and evaluated DeepSWE problem instances reported in Table 3. The source-specific construction and evaluation procedures are described below.

F.6.2 Software-Engineering Problem-Instance Statistics

Figure 9 and Table 27 report structural statistics for both software-engineering source corpora. For SWE Atlas, the problem-instance measures use the 32 one-each instances, while distribution measures use the 157 valid distributions across the five allocation modes. For DeepSWE, both sets of measures use the fixed 12-task evaluation cohort and its functional one-each allocations. Tasks outside this fixed cohort are not included in these descriptive statistics.

The code/source-unit column counts Atlas context units and DeepSWE code fragments. The runtime-observation column counts explicitly recorded execution observations. PIU length is measured from the evidence assigned to each PIU. Full source-material length uses the underlying Atlas source units

or the DeepSWE code archive, observations, and public-check catalog. The distribution column reports assigned participant packets; for DeepSWE these are the available PIU evidence packets before finalized full-context distractor packing. All lengths use the `cl100k_base` tokenizer.

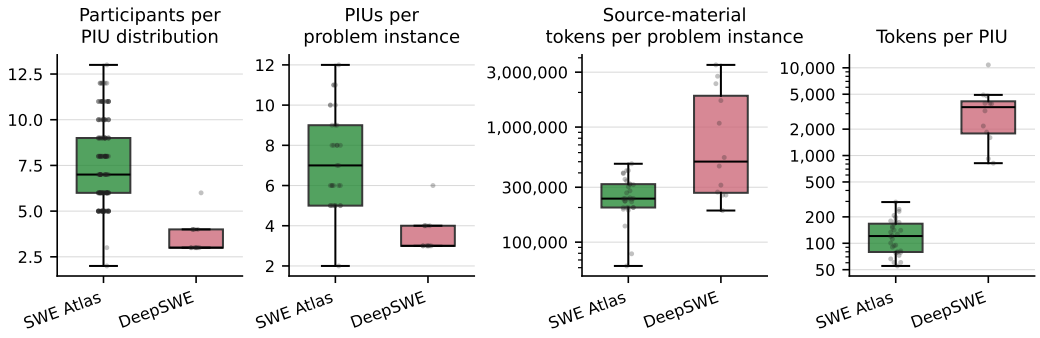


Figure 9: **Software-engineering problem-instance and PIU-distribution statistics.** In the PIU-count and PIU-length panels, each point is one problem instance. In the participant-count panel, each point is one PIU distribution; in the source-material-length panel, each point is one problem instance. SWE Atlas includes its five valid allocation modes; DeepSWE includes the 12 selected functional one-each allocations. Boxes show the median and interquartile range; token-length panels use logarithmic axes.

Table 27: **Software-engineering structural statistics.**

Problem-instance measures.

Source corpus	Problem instances	PIUs/ instance	Code/source units/instance	Runtime observations/instance	Full source-material tok./instance	PIU tok./ unit
SWE Atlas	32	7.3	236.5	0.0	261,951.8	130.2
DeepSWE	12	3.6	16.8	7.2	1,140,635.7	3570.6

PIU-distribution measures.

Source corpus	PIU distributions	Participants/ distribution	Source-material tok./ distribution
SWE Atlas	157	7.6	245,444.0
DeepSWE	12	3.6	59,474.2

F.6.3 SWE Atlas Codebase QnA Problem-Instance Construction

Source corpus and decision objective. SWE Atlas Codebase QnA asks for a source-grounded technical decision about a repository fixed at a recorded base commit. Construction loads the task question, positive and negative evaluation criteria, corpus-provided answer, Docker image, and base commit. The reference decision and evaluation criteria are used for hidden construction checks but are not shown to the agents that search the repository or receive the released evidence. Thirty-two of the 124 official entries pass the completed construction procedure and are retained.

Evidence collection and candidate units. Construction collects exact source, test, configuration, documentation, and, when a criterion requires observed behavior, runtime evidence from the unchanged Docker checkout. Runtime commands use the installed repository tools, do not use the network or install packages, and record their output, exit status, commit, and repository state. A runtime observation is included only with the source or test evidence needed to interpret it; runtime output alone does not form a PIU.

Candidate units are exact contiguous source-material spans that establish one answer contribution. Source, test, and runtime spans are grouped when they jointly establish one indivisible proposition; duplicates and fully contained contributions are merged, while independent contributions remain separate. Navigation results and model-written summaries are excluded. The construction does not target a fixed PIU count and retains at least two PIUs per collaborative problem instance.

Grounding, sufficiency, and necessity checks. Construction evaluates the complete candidate set, the decision objective with no candidate units, and each leave-one-out removal. The solver receives only the exact evidence available in the current check, while a separate judge assesses correctness and source grounding. A candidate is retained only when the complete set supports every required positive and negative criterion, the no-unit condition fails, and removing that candidate causes the linked answer component to fail. Candidate grouping or evidence collection is revised and the checks are repeated when these conditions do not hold.

PIU distribution and evaluation conditions. The evaluated distribution uses One PIU per participant. Source or test units and runtime output containing a PIU are reserved for its owner; additional context is assigned without duplication unless the allocation condition explicitly permits overlap. Multi-Agent PIU-Only provides the assigned PIU evidence, Multi-Agent Full-Context adds neutral additional repository material and actual runtime output, and Single-Agent All-PIU provides all pivotal information in a fixed order. The common observation identifies the task, repository, base commit, and response format without exposing PIU ownership or reference answers.

Atlas-specific leakage and salience checks. For every participant, a model-based check determines whether the initial local observation contains or supports a PIU assigned to another participant, including information reconstructable by combining multiple passages. Topical overlap alone does not count. Revisions may remove or reassign additional context but may not remove the owner's PIU evidence. The procedure also checks that PIU-bearing material is not identifiable from position, wrapper, annotation, or conspicuous length, and excludes instances that still contain foreign PIU information after bounded revision.

F.6.4 DeepSWE Problem-Instance Construction

DeepSWE provides long-horizon repository-level code-repair tasks, each defined by a pinned base repository, a public task instruction, and an official hidden verifier. A task may admit multiple verifier-passing patches, and repeated attempts by the same model may follow different search and implementation paths. The original DeepSWE evaluation accounts for this variation through repeated rollouts: its analysis uses three independent attempts by the same model on each task, while its final evaluation uses approximately four rollouts per task and model configuration. Successful GPT-5.5 repair trajectories take roughly 80 agent steps on average (Huang et al., 2026).

Our goal is not to reproduce this unrestricted long-horizon search, but to control the evidence required for collaboration under a fixed execution budget. We therefore define a DeepSWE PIU as a unit of functional evidence needed to support a successful repair under a fixed, bounded protocol. The complete PIU set is constructed so that, together with the public task instruction, it supports a verifier-passing patch using a restricted action space and five atomic edit turns.

Instruction-only screening. We begin with 40 DeepSWE tasks. For each task, the model receives the public task instruction but no PIUs or additional code evidence. We give the model the same five-turn repair budget used in the subsequent evidence-bundle checks and run three independent attempts. We continue PIU construction only for tasks that fail all three instruction-only attempts. This screen removes tasks whose instructions already provide enough information to locate and implement a repair within the bounded protocol. The instruction-only runs therefore serve as the no-evidence, or "none," condition for construction.

Constructor-only materials and evidence collection. For each candidate task, the constructor may inspect the reference patch, hidden-test patch and identifiers, public-check definitions and results, and the pinned base repository to locate implementation boundaries and audit evidence coverage. These materials are construction-only: the reference patch, hidden tests, verifier internals, hidden expected values, and constructor rationales are never copied into solver-visible evidence or exposed to evaluated agents. The public task instruction is common input to every condition and is not itself a PIU.

Construction collects exact task-relevant fragments from the unmodified base commit and deterministic runtime observations produced from the same checkout. Each observation records the executed probe, its output and exit status, and the repository state needed to interpret it. Observations

may describe public base-repository behavior but do not reveal hidden assertions, hidden expected values, or solution code. Solver-visible evidence is therefore composed of base-commit code and reproducible base-repository behavior, not pieces of the reference patch.

Functional PIU construction. A DeepSWE task may admit multiple implementations that pass the official verifier. Searching for a minimal evidence set relative to one reference patch could therefore steer the agent toward only one implementation path despite the existence of other valid repairs. We instead define each DeepSWE PIU to represent one coherent functional behavior or contract required by the verifier, rather than a fragment of a particular reference implementation.

A functional PIU may combine several exact code fragments with deterministic runtime observations when they jointly specify an implementation boundary, such as parsing and validation, state accumulation, lifecycle and cleanup, persistence, public-API wiring, or error propagation. Supporting types, callers, interfaces, and exports are included when they are required to implement that behavior. Each selected code fragment or observation is assigned to exactly one PIU. Functional labels, evidence-selection rationales, and hidden failure information remain constructor metadata and are not included in participant packets.

The purpose of these PIUs is not to reproduce the reference patch. Each PIU is grounded in base-repository code and reproducible runtime behavior, and the complete PIU set must support a verifier-passing repair under the bounded protocol.

Bounded evidence-bundle checks. We inspect candidate PIU bundles with the same source-restricted repair interface used in evaluation. The patch author has five atomic edit turns and may invoke checks from a frozen public-check catalog. Repository mutation is restricted to `replace`, `write`, and `delete` operations, and the agent may terminate before using all five turns. After termination, the official hidden verifier is run once. Success is determined solely by whether the final patch passes this verifier; matching the reference patch is not required.

Each complete PIU bundle is evaluated in three independent qualification runs. These runs form the complete-evidence, or "all," condition for construction. When all three instruction-only runs fail but the complete bundle enables a successful repair within the same five-turn protocol, the bundle provides operational evidence of sufficiency under that bounded protocol.

Repeated qualification outcomes and targeted leave-one-PIU-out checks are used diagnostically to refine the evidence bundle. We inspect whether removing a PIU reduces repair success or removes a functional boundary needed for the repair, and use the result to identify missing, irrelevant, or redundant evidence. Because repair trajectories are stochastic and multiple patches may pass the verifier, these leave-one-out checks are not treated as a formal proof of necessity over every possible repair procedure. They diagnose the contribution of each functional PIU under the fixed five-turn protocol.

Final evaluation cohort. After screening, evidence-bundle construction, and qualification, we freeze a 12-task evaluation cohort from the initial 40 tasks. The cohort contains four Go, four Python, two JavaScript, and two TypeScript tasks and preserves both programming-language coverage and variation in original benchmark difficulty. These 12 tasks are the constructed and evaluated DeepSWE problem instances reported in Table 3.

PIU allocation and evaluation conditions. The final functional PIUs are assigned one per participant. In Single-Agent All-PIU, one patch author receives the ordered union of all PIU evidence. In Multi-Agent PIU-Only, each participant receives only its assigned functional PIU. In Multi-Agent Full-Context, each participant receives its assigned PIU together with non-PIU base-repository code and additional runtime context.

Non-PIU code fragments in the Full-Context packets are distributed exclusively across participants. Additional runtime context uses only previously collected observations; selected non-owned observations may be duplicated as distractors across participant packets. No participant receives the entire repository. The collaborative transcript is sequential and coordinator-selected, and the final patch author acts on the public transcript using the same five-turn repair interface and public-check catalog. No condition exposes PIU identifiers, the reference patch, hidden-test information, hidden expected values, or constructor rationales.

Table 28: **Independent validation of two representative PIU-construction regimes.** Values are passing judgments or instances over the corresponding validation denominator, with percentages and 95% intervals in brackets. Instance- level intervals are Wilson intervals; leave-one-out intervals use an instance-cluster bootstrap with 10,000 resamples.

Source corpus	Leave-one-out pivotality	Participant-level source-material check
MAUD	64/67 (95.5%) [89.5, 100.0]	18/20 (90.0%) [69.9, 97.2]
FinLongDocQA	65/67 (97.0%) [92.3, 100.0]	20/20 (100.0%) [83.9, 100.0]

This is targeted independent validation of two representative construction regimes and is not an independent revalidation of every retained instance or source corpus. An unclear, invalid, failed, or missing judgment is counted as non-passing.

Repeated-run aggregation and failure handling. To account for stochastic repair trajectories, we schedule three independent runs for each task–model–condition combination. Within each task and condition, we first average the available verifier outcomes across repetitions and then average across the 12 tasks, giving each task equal weight.

A completed verifier failure, an ordinary unsuccessful repair, or a confirmed context-window failure receives a score of zero and remains in the denominator. A run is marked unavailable and excluded only when no valid task outcome can be obtained because of provider-capacity, API or serving, container or Docker, or missing-verifier-result failures. Availability exclusions are recorded separately from completed task failures.

F.7 Independent Validation of PIU Construction

All problem instances undergo checks during PIU construction, and the participant-level source-material check is performed after source-material distribution. Centralized solver success additionally establishes operational answerability, but does not by itself validate leave-one-out pivotality or the participant-level source-material check. We therefore conducted an additional post-construction audit of 20 instances from the fixed 30-instance MAUD evaluation set and 20 instances from the fixed 30-instance FinLongDocQA evaluation set. We used GPT-5.4, which was not used for instance construction, as the primary evaluator. The sample was frozen before evaluation. The audit covered every selected instance under every PIU leave-one-out removal and each participant's initial local observation together with the initial common observation. Solver outputs and answer accuracy were not used as evidence in these judgments.

For the leave-one-out pivotality check, a removal passes only when the linked answer component becomes unsupported or genuinely unresolved. The participant-level source-material check passes at the instance level only when no participant's initial local observation together with the initial common observation grounds the complete reference decision.

Across the two validated construction regimes, the leave-one-out results provide the strongest evidence that retained PIUs are individually pivotal. The participant-level source-material check finds that no participant's initial observation is sufficient for a complete grounded decision in any selected FinLongDocQA instance and identifies two MAUD instances in which one participant's initial observation together with the initial common observation is sufficient.

G Evaluation Results Across Domains and Source Corpora

The first three subsections disaggregate the model–domain results in Figure 2. Every reported comparison uses the same problem instances, their pivotal information, and the problem-specific evaluator across the three evaluation conditions within a source corpus. The legal, finance, medical, and software-engineering evaluations use One PIU per participant. All team episodes use Coordinator-selected speaker with 15 participant responses before the final decision. Within each model–source-corpus comparison, the two multi-agent conditions hold this PIU distribution, communication structure, and the source-corpus-specific resource budgets fixed. The results do not pool different communication structures or PIU-distribution conditions within an episode. For the three evaluation

conditions in Figure 2, confirmed context-window failures count as unsuccessful episodes, whereas episodes without a valid outcome because of provider-capacity or other infrastructure failures are unavailable.

G.1 Source-Corpus Results

Table 29 reports episode success rate for each source corpus and gives equal weight to the models. The model–domain results in Figure 2 pool problem instances across the source corpora within each model.

Table 29: **Evaluation results by source corpus.** Values are episode success rates in percent. Models receive equal weight within each source corpus. The Models column gives scheduled coverage; unavailable estimates are omitted condition by condition. Smaller $\pm$ terms are binomial standard errors propagated through each fixed-model average.

Domain	Source corpus	Problem instances	Models	Single-Agent All-PIU	Multi-Agent PIU-Only	Multi-Agent Full-Context
Legal	CUAD	30	8	90.4 ± 1.7	72.5 ± 2.7	62.9 ± 2.8
	MAUD	30	8	88.3 ± 1.8	67.9 ± 2.2	60.0 ± 2.7
	eCFR Casebook	18	8	88.9 ± 2.2	47.9 ± 3.5	18.1 ± 2.5
	MSLR	11	8	57.9 ± 4.0	45.5 ± 4.6	8.0 ± 2.6
	Multi-LexSum	29	8	62.5 ± 2.3	30.2 ± 2.3	20.2 ± 2.1
	SARA	45	8	63.6 ± 1.9	37.2 ± 2.0	8.1 ± 1.4
Finance	FinDVer	30	8	59.2 ± 2.6	25.4 ± 1.9	17.1 ± 1.8
	FinLongDocQA	30	8	62.5 ± 2.4	46.6 ± 2.3	22.9 ± 1.6
Medical	MLCR	29	8	79.3 ± 2.5	44.8 ± 3.0	12.5 ± 2.0
	EvidenceBench	45	8	49.4 ± 2.4	30.8 ± 2.0	6.9 ± 1.3
Software Engineering	SWE Atlas Codebase QnA	32	8	49.2 ± 2.6	34.8 ± 2.8	8.2 ± 1.6
	DeepSWE	12	8	9.0 ± 1.6	2.4 ± 1.4	2.1 ± 1.3

G.2 Model and Source-Corpus Results

Table 30 reports the complete three-condition model and source-corpus results underlying the domain and source-corpus summaries, together with available Single-Agent Full-Context estimates. In that condition, one model receives the ordered union of the source material assigned under the same fixed PIU distribution. Each row averages episode success over the fixed problem instances for one model and source corpus. Within a row, the two multi-agent conditions use the same fixed PIU distributions, communication structure, and resource budgets. The table does not pool the additional PIU-distribution conditions or communication structures. For legal Single-Agent Full-Context, the episode-success denominator includes evaluated inputs and structured-interface failures at the final-decision stage, with the latter counted as unsuccessful episodes. Serving-context rejections and other evaluation errors are excluded. When fewer than the fixed inputs contribute to an estimate, the table shows its denominator as n/N. An em dash in the Single-Agent Full-Context column means that the current archive does not provide a comparable success estimate; it does not convert an unavailable result into a failure.

Table 30: **Evaluation results by model and source corpus.** Values are episode success rates in percent. Smaller ± terms are binomial standard errors over each condition's episode-success denominator. For the three primary conditions, confirmed context-window failures are counted as unsuccessful episodes, while provider-capacity and other infrastructure failures remain unavailable. Single-Agent Full-Context excludes context-window, infrastructure, and judge failures, while output failures are counted as unsuccessful episodes. For legal Single-Agent Full-Context, serving-context rejections and other evaluation errors are excluded from this denominator, while structured-interface failures at the final-decision stage are included as unsuccessful episodes. When fewer than the fixed inputs contribute to an estimate, its denominator is shown as n/N. An em dash denotes that no comparable estimate is available. Appendix G.3 reports the corresponding coverage and failure counts.

Domain	Source corpus	Model	Problem instances	Single-Agent All-PIU	Single-Agent Full-Context	Multi-Agent PIU-Only	Multi-Agent Full-Context
Legal	CUAD	GPT-5.5	30	96.7 ± 3.3	86.7 ± 6.2	93.3 ± 4.6	90.0 ± 5.5
		Claude Sonnet 4.5	30	100.0 ± 0.0	90.0 ± 5.5	53.3 ± 9.1	43.3 ± 9.0
		Claude Haiku 4.5	30	90.0 ± 5.5	93.3 ± 4.6	70.0 ± 8.4	60.0 ± 8.9
		Gemma 3 4B IT	30	60.0 ± 8.9	30.0 ± 8.4	50.0 ± 9.1	40.0 ± 8.9
		Gemma 3 12B IT	30	90.0 ± 5.5	53.3 ± 9.1	70.0 ± 8.4	36.7 ± 8.8
		Gemma 3 27B IT	30	93.3 ± 4.6	50.0 ± 9.1	80.0 ± 7.3	70.0 ± 8.4
		Qwen3-32B	30	93.3 ± 4.6	83.3 ± 7.6 $n = 24/30$	73.3 ± 8.1	76.7 ± 7.7
		Qwen3-235B-A22B	30	100.0 ± 0.0	93.3 ± 4.6	90.0 ± 5.5	86.7 ± 6.2
Legal	MAUD	GPT-5.5	30	100.0 ± 0.0	96.7 ± 3.3	100.0 ± 0.0	100.0 ± 0.0
		Claude Sonnet 4.5	30	96.7 ± 3.3	80.0 ± 7.3	3.3 ± 3.3	30.0 ± 8.4
		Claude Haiku 4.5	30	90.0 ± 5.5	93.3 ± 4.6	73.3 ± 8.1	56.7 ± 9.0
		Gemma 3 4B IT	30	46.7 ± 9.1	17.2 ± 7.0 $n = 29/30$	33.3 ± 8.6	40.0 ± 8.9
		Gemma 3 12B IT	30	93.3 ± 4.6	72.4 ± 8.3 $n = 29/30$	73.3 ± 8.1	63.3 ± 8.8
		Gemma 3 27B IT	30	86.7 ± 6.2	58.6 ± 9.1 $n = 29/30$	76.7 ± 7.7	63.3 ± 8.8
		Qwen3-32B	30	96.7 ± 3.3	—	86.7 ± 6.2	33.3 ± 8.6
		Qwen3-235B-A22B	30	96.7 ± 3.3	96.7 ± 3.3	96.7 ± 3.3	93.3 ± 4.6
Legal	eCFR Casebook	GPT-5.5	18	100.0 ± 0.0	88.9 ± 7.4	83.3 ± 8.8	55.6 ± 11.7
		Claude Sonnet 4.5	18	100.0 ± 0.0	100.0 ± 0.0	55.6 ± 11.7	55.6 ± 11.7
		Claude Haiku 4.5	18	100.0 ± 0.0	88.9 ± 7.4	72.2 ± 10.6	27.8 ± 10.6
		Gemma 3 4B IT	18	50.0 ± 11.8	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0
		Gemma 3 12B IT	18	83.3 ± 8.8	16.7 ± 8.8	27.8 ± 10.6	0.0 ± 0.0
		Gemma 3 27B IT	18	83.3 ± 8.8	11.1 ± 7.4	22.2 ± 9.8	0.0 ± 0.0
		Qwen3-32B	18	94.4 ± 5.4	—	66.7 ± 11.1	0.0 ± 0.0
		Qwen3-235B-A22B	18	100.0 ± 0.0	94.4 ± 5.4	55.6 ± 11.7	5.6 ± 5.4
Legal	MSLR	GPT-5.5	11	100.0 ± 0.0	27.3 ± 13.4	81.8 ± 11.6	36.4 ± 14.5
		Claude Sonnet 4.5	11	81.8 ± 11.6	9.1 ± 8.7	63.6 ± 14.5	9.1 ± 8.7
		Claude Haiku 4.5	11	81.8 ± 11.6	0.0 ± 0.0	27.3 ± 13.4	0.0 ± 0.0
		Gemma 3 4B IT	11	9.1 ± 8.7	0.0 ± 0.0	0.0 ± 0.0	9.1 ± 8.7
		Gemma 3 12B IT	11	27.3 ± 13.4	0.0 ± 0.0	63.6 ± 14.5	0.0 ± 0.0
		Gemma 3 27B IT	11	18.2 ± 11.6	9.1 ± 8.7	36.4 ± 14.5	0.0 ± 0.0
		Qwen3-32B	11	63.6 ± 14.5	28.6 ± 17.1 $n = 7/11$	36.4 ± 14.5	9.1 ± 8.7
		Qwen3-235B-A22B	11	81.8 ± 11.6	45.5 ± 15.0	54.5 ± 15.0	0.0 ± 0.0
Legal	Multi-LexSum	GPT-5.5	29	100.0 ± 0.0	72.4 ± 8.3	93.1 ± 4.7	72.4 ± 8.3
		Claude Sonnet 4.5	29	93.1 ± 4.7	41.4 ± 9.1	24.1 ± 7.9	0.0 ± 0.0
		Claude Haiku 4.5	29	82.8 ± 7.0	31.0 ± 8.6	48.3 ± 9.3	31.0 ± 8.6
		Gemma 3 4B IT	29	6.9 ± 4.7	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0
		Gemma 3 12B IT	29	31.0 ± 8.6	3.4 ± 3.4	3.4 ± 3.4	0.0 ± 0.0
		Gemma 3 27B IT	29	37.9 ± 9.0	6.9 ± 4.7	6.9 ± 4.7	3.4 ± 3.4
		Qwen3-32B	29	48.3 ± 9.3	24.1 ± 7.9	24.1 ± 7.9	31.0 ± 8.6
		Qwen3-235B-A22B	29	100.0 ± 0.0	20.7 ± 7.5	41.4 ± 9.1	24.1 ± 7.9
Legal	SARA	GPT-5.5	45	93.3 ± 3.7	40.0 ± 7.3	91.1 ± 4.2	22.2 ± 6.2
		Claude Sonnet 4.5	45	93.3 ± 3.7	13.3 ± 5.1	42.2 ± 7.4	4.4 ± 3.1
		Claude Haiku 4.5	45	86.7 ± 5.1	0.0 ± 0.0	75.6 ± 6.4	8.9 ± 4.2
		Gemma 3 4B IT	45	2.2 ± 2.2	0.0 ± 0.0 $n = 30/45$	0.0 ± 0.0	0.0 ± 0.0
		Gemma 3 12B IT	45	46.7 ± 7.4	3.3 ± 3.3 $n = 30/45$	6.7 ± 3.7	6.7 ± 3.7
		Gemma 3 27B IT	45	48.9 ± 7.5	0.0 ± 0.0 $n = 30/45$	15.6 ± 5.4	11.1 ± 4.7
		Qwen3-32B	45	46.7 ± 7.4	—	24.4 ± 6.4	4.4 ± 3.1

Continued on next page

Table 30 continued

Domain	Source corpus	Model	Problem instances	Single-Agent All-PIU	Single-Agent Full-Context	Multi-Agent PIU-Only	Multi-Agent Full-Context
		Qwen3-235B-A22B	45	91.1 ± 4.2	13.3 ± 5.1	42.2 ± 7.4	6.7 ± 3.7
Finance	FinDVer	GPT-5.5	30	86.7 ± 6.2	36.7 ± 8.8	86.7 ± 6.2	63.3 ± 8.8
		Claude Sonnet 4.5	30	86.7 ± 6.2	53.3 ± 9.1	70.0 ± 8.4	56.7 ± 9.0
		Claude Haiku 4.5	30	73.3 ± 8.1	36.7 ± 8.8	0.0 ± 0.0	0.0 ± 0.0
		Gemma 3 4B IT	30	3.3 ± 3.3	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0
		Gemma 3 12B IT	30	33.3 ± 8.6	0.0 ± 0.0	3.3 ± 3.3	3.3 ± 3.3
		Gemma 3 27B IT	30	43.3 ± 9.0	6.7 ± 4.6	23.3 ± 7.7	3.3 ± 3.3
		Qwen3-32B	30	70.0 ± 8.4	40.0 ± 21.9 $n = 5/30$	0.0 ± 0.0	0.0 ± 0.0
		Qwen3-235B-A22B	30	76.7 ± 7.7	16.7 ± 6.8	20.0 ± 7.3	10.0 ± 5.5
Finance	FinLongDocQA	GPT-5.5	30	100.0 ± 0.0	73.3 ± 8.1	93.3 ± 4.6	90.0 ± 5.5
		Claude Sonnet 4.5	30	96.7 ± 3.3	61.5 ± 9.5 $n = 26/30$	83.3 ± 6.8	70.0 ± 8.4
		Claude Haiku 4.5	30	56.7 ± 9.0	53.8 ± 9.8 $n = 26/30$	3.3 ± 3.3	0.0 ± 0.0
		Gemma 3 4B IT	30	0.0 ± 0.0	4.8 ± 4.6 $n = 21/30$	0.0 ± 0.0	0.0 ± 0.0
		Gemma 3 12B IT	30	33.3 ± 8.6	9.5 ± 6.4 $n = 21/30$	10.0 ± 5.5	0.0 ± 0.0
		Gemma 3 27B IT	30	53.3 ± 9.1	23.8 ± 9.3 $n = 21/30$	53.3 ± 9.1	6.7 ± 4.6
		Qwen3-32B	30	80.0 ± 7.3	50.0 ± 35.4 $n = 2/30$	63.3 ± 8.8	0.0 ± 0.0
		Qwen3-235B-A22B	30	80.0 ± 7.3	21.4 ± 7.8 $n = 28/30$	66.7 ± 8.6	16.7 ± 6.8
Medical	MLCR	GPT-5.5	29	96.6 ± 3.4	31.0 ± 8.6	65.5 ± 8.8	37.9 ± 9.0
		Claude Sonnet 4.5	29	86.2 ± 6.4	37.9 ± 9.0	72.4 ± 8.3	17.2 ± 7.0
		Claude Haiku 4.5	29	82.8 ± 7.0	31.0 ± 8.6	58.6 ± 9.1	17.2 ± 7.0
		Gemma 3 4B IT	29	51.7 ± 9.3	0.0 ± 0.0	6.9 ± 4.7	0.0 ± 0.0
		Gemma 3 12B IT	29	79.3 ± 7.5	3.4 ± 3.4	24.1 ± 7.9	0.0 ± 0.0
		Gemma 3 27B IT	29	79.3 ± 7.5	6.9 ± 4.7	34.5 ± 8.8	3.4 ± 3.4
		Qwen3-32B	29	75.9 ± 7.9	0.0 ± 0.0 $n = 12/29$	41.4 ± 9.1	6.9 ± 4.7
		Qwen3-235B-A22B	29	82.8 ± 7.0	31.0 ± 8.6	55.2 ± 9.2	17.2 ± 7.0
Medical	EvidenceBench	GPT-5.5	45	93.3 ± 3.7	31.1 ± 6.9	88.9 ± 4.7	20.0 ± 6.0
		Claude Sonnet 4.5	45	57.8 ± 7.4	24.4 ± 6.4	48.9 ± 7.5	13.3 ± 5.1
		Claude Haiku 4.5	45	53.3 ± 7.4	20.0 ± 6.0	44.4 ± 7.4	4.4 ± 3.1
		Gemma 3 4B IT	45	13.3 ± 5.1	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0
		Gemma 3 12B IT	45	33.3 ± 7.0	0.0 ± 0.0	13.3 ± 5.1	2.2 ± 2.2
		Gemma 3 27B IT	45	44.4 ± 7.4	8.9 ± 4.2	6.7 ± 3.7	2.2 ± 2.2
		Qwen3-32B	45	37.8 ± 7.2	8.9 ± 4.2	20.0 ± 6.0	6.7 ± 3.7
		Qwen3-235B-A22B	45	62.2 ± 7.2	8.9 ± 4.2	24.4 ± 6.4	6.7 ± 3.7
Software Engineering	SWE Atlas Codebase QnA	GPT-5.5	32	84.4 ± 6.4	61.9 ± 10.6 $n = 21/32$	43.8 ± 8.8	25.0 ± 7.7
		Claude Sonnet 4.5	32	81.2 ± 6.9	0.0 ± 0.0 $n = 1/32$	50.0 ± 8.8	25.0 ± 7.7
		Claude Haiku 4.5	32	75.0 ± 7.7	100.0 ± 0.0 $n = 1/32$	50.0 ± 8.8	9.4 ± 5.2
		Gemma 3 4B IT	32	3.1 ± 3.1	0.0 ± 0.0 $n = 2/32$	3.1 ± 3.1	0.0 ± 0.0
		Gemma 3 12B IT	32	31.2 ± 8.2	0.0 ± 0.0 $n = 2/32$	28.1 ± 7.9	0.0 ± 0.0
		Gemma 3 27B IT	32	43.8 ± 8.8	0.0 ± 0.0 $n = 2/32$	40.6 ± 8.7	0.0 ± 0.0
		Qwen3-32B	32	31.2 ± 8.2	—	25.0 ± 7.7	3.1 ± 3.1
		Qwen3-235B-A22B	32	43.8 ± 8.8	0.0 ± 0.0 $n = 2/32$	37.5 ± 8.6	3.1 ± 3.1
Software Engineering	DeepSWE	GPT-5.5	12	72.2 ± 12.9	—	19.4 ± 11.4	16.7 ± 10.8
		Claude Sonnet 4.5	12	0.0 ± 0.0	—	0.0 ± 0.0	0.0 ± 0.0
		Claude Haiku 4.5	12	0.0 ± 0.0	—	0.0 ± 0.0	0.0 ± 0.0
		Gemma 3 4B IT	12	0.0 ± 0.0	—	0.0 ± 0.0	0.0 ± 0.0
		Gemma 3 12B IT	12	0.0 ± 0.0	—	0.0 ± 0.0	0.0 ± 0.0
		Gemma 3 27B IT	12	0.0 ± 0.0	—	0.0 ± 0.0	0.0 ± 0.0
		Qwen3-32B	12	0.0 ± 0.0	—	0.0 ± 0.0	0.0 ± 0.0
		Qwen3-235B-A22B	12	0.0 ± 0.0	—	0.0 ± 0.0	0.0 ± 0.0

Legal structured-interface failures. The reported episode success rate is a strict end-to-end measure. We define a *structured-interface failure* as an episode that does not produce a valid final

decision because a participant response or final-decision response cannot be parsed under the required structured response format. A valid final decision is one that is successfully parsed and passed to the problem-specific evaluator. Structured-interface failures remain unsuccessful episodes.

Across the primary legal evaluation, each of the three evaluation conditions contains 1,304 episodes. In 276 multi-agent episodes, a participant response cannot be parsed and the episode terminates before the final-decision stage. Four additional episodes reach the final-decision stage but do not produce a valid final-decision response. Of these 280 structured-interface failures, 254 are produced by the two Claude models: 250 occur during participant interaction and four at the final-decision stage.

The failure records do not support reliable semantic rescoring. For failed responses, the archive retains only the first 500 characters and does not record the complete raw response, provider finish reason, or output-token count. We therefore cannot reliably distinguish schema noncompliance from generation truncation or determine what a model would have produced had generation continued.

Table 31: **Outcome decomposition for the primary legal evaluation.** Structured-interface failures remain unsuccessful episodes in the reported episode success rate. The final column is a diagnostic conditional rate computed only over episodes that produce a valid final decision.

Evaluation condition	Strict success	Valid final decision, unsuccessful	Structured-interface failure	Success given a valid final decision
Single-Agent All-PIU	982/1,304 (75.3%)	318/1,304 (24.4%)	4/1,304 (0.3%)	75.5%
Multi-Agent PIU-Only	650/1,304 (49.8%)	531/1,304 (40.7%)	123/1,304 (9.4%)	55.0%
Multi-Agent Full-Context	404/1,304 (31.0%)	747/1,304 (57.3%)	153/1,304 (11.7%)	35.1%

Among episodes that produce a valid final decision, the corresponding diagnostic success rates are 75.5%, 55.0%, and 35.1%. The communication and retrieval gaps therefore remain 20.5 and 19.9 percentage points, respectively. These conditional rates are descriptive rather than alternative estimates of end-to-end performance: producing a valid final decision is itself an outcome that can depend on the model and evaluation condition.

As an additional sensitivity analysis, excluding both Claude models gives strict aggregate legal episode success rates of 69.8%, 49.4%, and 32.3% in Single-Agent All-PIU, Multi-Agent PIU-Only, and Multi-Agent Full-Context, respectively. Conditional on producing a valid final decision, the corresponding rates are 69.8%, 49.7%, and 33.0%. The strict gaps are 20.4 and 17.1 percentage points, and the conditional gaps are 20.1 and 16.7 points.

In the additional Single-Agent Full-Context evaluation, Claude Sonnet 4.5 and Claude Haiku 4.5 produce seven and two structured-interface failures, respectively, at the final-decision stage. These evaluated inputs remain unsuccessful episodes; serving-context rejections and other evaluation errors remain excluded from the episode-success denominator as described above.

G.3 Single-Agent Full-Context Evaluation Counts

Some Single-Agent Full-Context model inputs exceed the serving endpoint's context limit or otherwise fail before evaluation. The current coverage audit spans the 11 source corpora other than DeepSWE; Single-Agent Full-Context was not run for the DeepSWE cohort. Table 32 separately reports evaluated problem-instance–model pairs, context-window failures, output failures, and other unavailable failures across domains. The percentages use all fixed problem-instance–model pairs for each domain–model row as the denominator. Episode-success rates include output failures as unsuccessful episodes but exclude context-window, infrastructure, and judge failures. Table 30 reports the resulting failed/evaluable count and uses an em dash only when no comparable success estimate is available.

Across these 11 source corpora and all models, 2,099 of 2,632 scheduled inputs (79.7%) were normally evaluated, 50 (1.9%) had output failures that remain in the episode-success denominator, 460 (17.5%) exceeded a context window, and 23 (0.9%) had other unavailable failures. Thus, 2,149 inputs contribute to Single-Agent Full-Context episode-success estimates. The Full-Context values

Table 32: **Single-Agent Full-Context evaluation counts by domain and model.** The table covers the 11 source corpora other than DeepSWE, for which this condition was not run. Percentages in parentheses use all fixed problem-instance–model pairs in each domain–model row. Context-window, infrastructure, and judge failures are excluded from the episode-success denominator; output failures are included as unsuccessful episodes.

Domain	Model	Planned	Evaluated, n (%)	Context unavailable, n (%)	Output failures, n (%)	Other failures, n (%)
Legal	GPT-5.5	163	163 (100.0)	0 (0.0)	0 (0.0)	0 (0.0)
Legal	Claude Sonnet 4.5	163	156 (95.7)	0 (0.0)	7 (4.3)	0 (0.0)
Legal	Claude Haiku 4.5	163	161 (98.8)	0 (0.0)	2 (1.2)	0 (0.0)
Legal	Gemma 3 4B IT	163	147 (90.2)	16 (9.8)	0 (0.0)	0 (0.0)
Legal	Gemma 3 12B IT	163	147 (90.2)	16 (9.8)	0 (0.0)	0 (0.0)
Legal	Gemma 3 27B IT	163	147 (90.2)	16 (9.8)	0 (0.0)	0 (0.0)
Legal	Qwen3-32B	163	60 (36.8)	100 (61.3)	0 (0.0)	3 (1.8)
Legal	Qwen3-235B-A22B	163	163 (100.0)	0 (0.0)	0 (0.0)	0 (0.0)
Finance	GPT-5.5	60	42 (70.0)	0 (0.0)	18 (30.0)	0 (0.0)
Finance	Claude Sonnet 4.5	60	53 (88.3)	4 (6.7)	3 (5.0)	0 (0.0)
Finance	Claude Haiku 4.5	60	55 (91.7)	4 (6.7)	1 (1.7)	0 (0.0)
Finance	Gemma 3 4B IT	60	48 (80.0)	9 (15.0)	3 (5.0)	0 (0.0)
Finance	Gemma 3 12B IT	60	51 (85.0)	9 (15.0)	0 (0.0)	0 (0.0)
Finance	Gemma 3 27B IT	60	51 (85.0)	9 (15.0)	0 (0.0)	0 (0.0)
Finance	Qwen3-32B	60	7 (11.7)	53 (88.3)	0 (0.0)	0 (0.0)
Finance	Qwen3-235B-A22B	60	57 (95.0)	2 (3.3)	1 (1.7)	0 (0.0)
Medical	GPT-5.5	74	67 (90.5)	0 (0.0)	7 (9.5)	0 (0.0)
Medical	Claude Sonnet 4.5	74	71 (95.9)	0 (0.0)	3 (4.1)	0 (0.0)
Medical	Claude Haiku 4.5	74	72 (97.3)	0 (0.0)	2 (2.7)	0 (0.0)
Medical	Gemma 3 4B IT	74	72 (97.3)	0 (0.0)	2 (2.7)	0 (0.0)
Medical	Gemma 3 12B IT	74	73 (98.6)	0 (0.0)	1 (1.4)	0 (0.0)
Medical	Gemma 3 27B IT	74	74 (100.0)	0 (0.0)	0 (0.0)	0 (0.0)
Medical	Qwen3-32B	74	57 (77.0)	17 (23.0)	0 (0.0)	0 (0.0)
Medical	Qwen3-235B-A22B	74	74 (100.0)	0 (0.0)	0 (0.0)	0 (0.0)
SWE	GPT-5.5	32	21 (65.6)	11 (34.4)	0 (0.0)	0 (0.0)
SWE	Claude Sonnet 4.5	32	1 (3.1)	22 (68.8)	0 (0.0)	9 (28.1)
SWE	Claude Haiku 4.5	32	1 (3.1)	22 (68.8)	0 (0.0)	9 (28.1)
SWE	Gemma 3 4B IT	32	2 (6.2)	30 (93.8)	0 (0.0)	0 (0.0)
SWE	Gemma 3 12B IT	32	2 (6.2)	30 (93.8)	0 (0.0)	0 (0.0)
SWE	Gemma 3 27B IT	32	2 (6.2)	30 (93.8)	0 (0.0)	0 (0.0)
SWE	Qwen3-32B	32	0 (0.0)	30 (93.8)	0 (0.0)	2 (6.2)
SWE	Qwen3-235B-A22B	32	2 (6.2)	30 (93.8)	0 (0.0)	0 (0.0)
All domains	All models	2632	2099 (79.7)	460 (17.5)	50 (1.9)	23 (0.9)

reported in Table 30 therefore distinguish episode success from evaluation coverage rather than treating unavailable inputs as incorrect final decisions.

The nine legal Claude entries labeled Output failures in the table, seven for Sonnet and two for Haiku, are structured-interface failures at the final-decision stage, as described in Appendix G.2, and are included as unsuccessful episodes rather than treated as serving-context rejections.

G.4 Synthesis and Error-Uptake Analysis

This appendix reports the labeling criteria and transcript interventions for correct synthesis and error uptake summarized in Section 4.3.

G.4.1 Definitions and Design

A PIU is *faithfully communicated* when a participant message preserves its meaning without changing its value, entity, scope, qualification, or effect. An *error* is an incorrect or unsupported sentence or component that could affect the final decision. *Error uptake* occurs when a later participant uses an error from a previous visible message to produce a new sentence; rejecting or correcting the earlier error does not count. The intended *synthesis* criterion is an inference that combines at least two distinct PIUs; repeating one PIU or merely proposing an answer does not count. A synthesis inference is *correct* when it is supported by the reference PIUs and reference decision. An *explicit answer proposal* directly states the final choice or requested answer.

The analysis uses 474 frozen Multi-Agent PIU-Only trajectories: 79 problem instances crossed with Round robin, Current-speaker selection, and Simultaneous responses, for Qwen3-32B and Qwen3-235B-A22B participants; Section 4.3 abbreviates the latter as Qwen3-235B. Every participant receives exactly one PIU. A separate GPT-5.5 auditor labels sentence-level components. Participants are not rerun. Instead, we reconstruct the transcript supplied to the final decision maker and regenerate the final decision. The intervention results pool Round robin and Current-speaker selection, giving 158 original trajectories for each participant-model–final-decision-maker combination; intervention effects are paired within trajectory, and confidence intervals are paired 95% bootstrap intervals.

G.4.2 Correct-Synthesis Intervention

For each original transcript T, let $E(T)$ contain the reference versions of only the PIUs faithfully communicated in T. PIUs that are absent or communicated incorrectly are excluded. Let $S(T)$ contain the correct synthesis claims generated in T, after removing explicit answer proposals. We compare $E(T)$ with $E(T) + S(T)$. The two inputs therefore contain the same PIUs and differ only in whether they include correct synthesis.

Table 33: **Effect of correct synthesis on episode success.** Effects and paired 95% bootstrap confidence intervals are in percentage points.

Participant model	Final decision maker	$E(T)$	$E(T) + S(T)$	Effect	95% CI
Qwen3-32B	Qwen3-32B	54.4%	69.0%	+14.6	[7.0, 22.2]
Qwen3-32B	GPT-5.5	69.6%	75.3%	+5.7	[1.9, 10.1]
Qwen3-235B-A22B	Qwen3-235B-A22B	60.1%	77.8%	+17.7	[10.1, 25.3]
Qwen3-235B-A22B	GPT-5.5	78.5%	86.7%	+8.2	[3.2, 13.3]

Correct synthesis improves episode success in all four comparisons. The effect is larger for the Qwen3 final decision maker than for GPT-5.5 for both participant models. Because $S(T)$ excludes explicit answer proposals, this effect cannot be attributed to direct answer copying.

Scope qualification. The intended synthesis definition requires at least two distinct PIUs. The auditor-labeled set used here also includes some inferences that combine one PIU with information from the publicly visible decision options. Accordingly, the reported effects apply to this broader auditor-labeled set and should not be interpreted as isolating synthesis across distinct PIUs.

G.4.3 Error-Uptake Intervention

The control is the reconstructed original transcript. In the intervention, we remove later sentences produced using an error from a previous visible message, while retaining the sentence containing the original error. This isolates the additional effect of later participants taking up the error. A positive effect means that removing error-uptake sentences improves episode success.

Table 34: **Effect of removing error uptake.** Effects and paired 95% bootstrap confidence intervals are in percentage points.

Participant model	Final decision maker	Effect	95% CI
Qwen3-32B	Qwen3-32B	+2.5	[−1.3, 6.3]
Qwen3-32B	GPT-5.5	+1.3	[−2.5, 5.1]
Qwen3-235B-A22B	Qwen3-235B-A22B	+5.7	[1.9, 10.1]
Qwen3-235B-A22B	GPT-5.5	+7.6	[3.2, 12.7]

Removing error uptake improves episode success for Qwen3-235B-A22B participants under both final decision makers. The Qwen3-32B estimates have the same sign, but their confidence intervals include zero. These effects are smaller than the corresponding correct-synthesis effects in all four comparisons and do not by themselves explain the communication-structure ordering. We do not

apply the error-uptake intervention to Simultaneous responses because participants cannot observe one another; incorrect or inconsistent simultaneous messages are independent rather than propagated errors.

G.4.4 Explicit Answer Proposals

As a separate check, we remove explicit answer proposals from the original Round robin and Current-speaker selection transcripts. The effect is -5.7 percentage points for Qwen3-32B participants and final decision makers and -10.8 points for Qwen3-235B-A22B participants and final decision makers. Explicit answer proposals therefore also help the Qwen3 final decision makers. They cannot, however, explain the correct-synthesis results in Table 33, where such proposals are excluded from $S(T)$.

Overall, adding correct synthesis improves episode success in all four comparisons. Removing error uptake has smaller effects, with confidence intervals excluding zero only for Qwen3-235B-A22B participants. The synthesis effects are smaller with GPT-5.5 than with the Qwen3 final decision makers.

G.5 PIU-Distribution–Communication-Structure Interaction Estimates

Table 35 reports the complete probability-scale interaction estimates from the problem-family and model fixed-effects specifications described in Section 4.4.

For PIU-distribution condition ℓ and communication structure c, the reported value is the average difference-in-differences in predicted episode success probability:

$$100\,\mathbb{E}[\{p_e(\ell, c) - p_e(\ell, c_0)\} - \{p_e(\ell_0, c) - p_e(\ell_0, c_0)\}]\,,$$

where $p_e(\ell, c)$ is episode e's predicted success probability under (ℓ, c), ℓ_0 and c_0 are the PIU-distribution and communication-structure reference categories, and the expectation averages over the evaluated models and problem instances.

Table 35: **PIU-distribution–communication-structure interactions across the evaluated problem families.** Entries are average interaction estimates in episode-success percentage points; smaller $\pm$ terms are problem-instance-clustered standard errors. Rows and columns are relative to One PIU per participant and Coordinator-selected speaker, respectively. Yellow cells and asterisks mark 95% confidence intervals excluding zero.

Evaluation condition	PIU-distribution condition	Broadcast query	Round robin	Current-speaker selection	Simultaneous responses
Multi-Agent PIU-Only $N = 50,438$ joint $p < .001$	Pairwise PIU overlap	$4.5 \pm 1.4^{*}$	$4.4 \pm 1.5^{*}$	$4.5 \pm 1.9^{*}$	1.2 ± 1.5
	Extra participant without a PIU	0.6 ± 1.5	1.8 ± 1.3	-0.3 ± 1.6	-1.1 ± 1.6
	One common PIU	-0.6 ± 1.5	1.3 ± 1.5	2.7 ± 1.8	$-3.3 \pm 1.5^{*}$
	All-but-one PIU concentration	-1.1 ± 1.6	$4.3 \pm 1.5^{*}$	$6.5 \pm 2.1^{*}$	-0.4 ± 1.6
Multi-Agent Full-Context $N = 48,499$ joint $p < .001$	Pairwise PIU overlap	0.4 ± 1.2	2.3 ± 1.2	-1.5 ± 1.5	0.9 ± 1.3
	Extra participant without a PIU	-1.7 ± 1.3	-0.2 ± 1.2	-2.3 ± 1.3	-0.6 ± 1.1
	One common PIU	-0.4 ± 1.4	0.6 ± 1.2	$-3.6 \pm 1.5^{*}$	-1.4 ± 1.4
	All-but-one PIU concentration	-0.9 ± 1.5	-0.9 ± 1.3	-0.8 ± 1.6	$-4.0 \pm 1.5^{*}$

G.6 Robustness Analysis with Problem-Instance Fixed Effects

The main analysis in Section 4.4 includes problem-family and model fixed effects. As a stricter within-problem-instance comparison, we replace the problem-family fixed effects with problem-instance fixed effects:

$$y_e \sim \text{Bernoulli}(p_e), \qquad \text{logit}(p_e) = \beta_0 + \alpha_{m_e} + \delta_{q_e} + \pi_{\ell_e} + \lambda_{c_e} + \omega_{\ell_e, c_e}.$$

Because each problem instance belongs to exactly one problem family, the problem-instance fixed effects absorb the problem-family fixed effects rather than entering the model alongside them. In both analyses, we assign episode success of zero to 3,448 attempted episodes that failed and exclude unavailable cells. The reference categories, weighting, and problem-instance-clustered standard errors are otherwise the same as in the main analysis.

Table 36: **Robustness analysis with problem-instance fixed effects.** Entries are average interaction estimates in episode-success percentage points; smaller ± terms are problem-instance-clustered standard errors. Columns replace Coordinator-selected speaker, and the four displayed PIU-distribution rows replace One PIU per participant. Light yellow cells and asterisks mark 95% confidence intervals that exclude zero.

Evaluation condition	PIU-distribution condition	Broadcast query	Round robin	Current-speaker selection	Simultaneous responses
Multi-Agent PIU-Only $N = 50,438$ joint $p < .001$	Pairwise PIU overlap	$4.8 \pm 1.5^{*}$	$4.7 \pm 1.5^{*}$	$4.6 \pm 1.9^{*}$	1.6 ± 1.5
	Extra participant without a PIU	0.6 ± 1.5	1.9 ± 1.3	-0.3 ± 1.6	-1.0 ± 1.6
	One common PIU	-0.5 ± 1.5	1.5 ± 1.5	2.7 ± 1.8	-2.9 ± 1.6
	All-but-one PIU concentration	-1.1 ± 1.6	$4.2 \pm 1.5^{*}$	$6.3 \pm 2.1^{*}$	-0.3 ± 1.6
Multi-Agent Full-Context $N = 48,499$ joint $p < .001$	Pairwise PIU overlap	0.4 ± 1.2	$2.4 \pm 1.2^{*}$	-1.5 ± 1.6	1.0 ± 1.3
	Extra participant without a PIU	-1.8 ± 1.3	-0.2 ± 1.2	-2.3 ± 1.2	-0.7 ± 1.1
	One common PIU	-0.4 ± 1.4	0.6 ± 1.2	$-3.6 \pm 1.5^{*}$	-1.3 ± 1.4
	All-but-one PIU concentration	-1.0 ± 1.5	-0.9 ± 1.3	-0.8 ± 1.6	$-3.9 \pm 1.5^{*}$

The problem-instance fixed effects absorb differences in absolute difficulty among the evaluated problem instances. The reported interactions therefore compare changes across PIU distributions and communication structures within the same problem instances. The same five positive Multi-Agent PIU-Only interactions and the same two negative Multi-Agent Full-Context interactions have 95% confidence intervals that exclude zero as in the main analysis, with similar point estimates. The positive Pairwise PIU overlap–Round robin interaction in Multi-Agent Full-Context has a 95% confidence interval that excludes zero only with problem-instance fixed effects. The negative One common PIU–Simultaneous responses Multi-Agent PIU-Only interaction has a 95% confidence interval that includes zero with problem-instance fixed effects.

G.7 Interpreting PIU-Distribution–Communication-Structure Interactions

G.7.1 Matched Ordering Comparison

An interaction measures how the difference in episode success rate between communication structures changes across PIU distributions. To make that change directly interpretable, we retain every problem-instance–model pair observed under all four combinations of One common PIU and Extra participant without a PIU with Current-speaker selection and Simultaneous responses. This gives 1,750 pairs from 225 problem instances across 11 problem families and holds the problem instance and model fixed. In Multi-Agent PIU-Only, Current-speaker selection trails Simultaneous responses by 4.3 percentage points under One common PIU and by 8.6 points under Extra participant without a PIU. Its deficit is therefore 4.3 points smaller under One common PIU (standard error 1.6), and the 95% confidence interval excludes zero. Under Multi-Agent Full-Context, the corresponding change is 0.7 points (standard error 1.1), and its 95% confidence interval includes zero.

Table 37 reports the same matched comparison for the six legal problem families. Here, the ordering reverses in Multi-Agent PIU-Only: Current-speaker selection leads Simultaneous responses by 3.9 percentage points under One common PIU but trails it by 9.8 points under Extra participant without a PIU. The difference changes by 13.8 points (standard error 2.9), and its 95% confidence interval

excludes zero. Under Multi-Agent Full-Context, the corresponding change is 2.6 points (standard error 1.9), and its 95% confidence interval includes zero.

Table 37: **Multi-Agent PIU-Only ordering reversal between Current-speaker selection and Simultaneous responses across the six legal problem families.** Entries are episode success rates in percent over 711 problem-instance–model pairs observed under all four combinations, from 89 problem instances across 6 problem families. The final column reports Current-speaker selection minus Simultaneous responses; the bottom entry is the difference between the two row contrasts. The asterisk marks a 95% confidence interval, based on standard errors clustered by problem instance, that excludes zero.

PIU-distribution condition	Current-speaker selection	Simultaneous responses	Difference
One common PIU	45.4	41.5	$+3.9$
Extra participant without a PIU	43.0	52.9	-9.8
Difference between contrasts			$+13.8^{*}$

Selected episode trajectory. A selected SARA problem instance illustrates one possible source of the ordering reversal across the legal problem families under Extra participant without a PIU. The problem asks for Alice's 2017 employer excise tax. Six agricultural workers each received $\$5{,}012$, so total wages are $\$30{,}072$. This exceeds the $\$20{,}000$ agricultural-employer threshold; each payment is below the $\$7{,}000$ per-worker cap; and the tax is six percent of covered wages, or $\$1{,}804.32$. The PIU distribution places the PIUs corresponding to the payment and rule components in participants' initial local observations and adds one participant without a PIU. Both episodes use Qwen3-235B-A22B for the participants and the separate final decision maker and differ only in communication structure.

Table 38: **Selected Extra participant without a PIU episode trajectory.** The problem instance, model, PIU distribution, and problem-specific evaluator are held fixed. Descriptions of the communication transcript are shortened without changing their claims.

Stage	Current-speaker selection	Simultaneous responses
Payment reports	The first speakers report $\$5{,}012$ payments to George and Cameron. Because the amounts match, the second report is interpreted as a possible duplicate.	Participants independently report payments to George, Cameron, Emily, Dan, Bob, and Fred, each with a distinct worker name.
Effect on later messages	The participant without a PIU repeats that only one payment has been established. Later speakers accommodate this framing and characterize their differently named payments as further duplicates.	Each message is generated before any message from the round is added to the communication transcript. The message from the participant without a PIU therefore does not change what the other six participants communicate.
Separate final decision maker	Uses total wages of $\$5{,}012$, concludes that the $\$20{,}000$ threshold is not met, and returns zero tax.	Recognizes six payments, computes $6 \times 5{,}012 = 30{,}072$ dollars, and returns $0.06 \times 30{,}072 = 1{,}804.32$ dollars.
Evaluation	Incorrect	Correct

In this episode, conditioning later messages on earlier messages permits an initial duplicate-payment interpretation to propagate under Current-speaker selection. Under Simultaneous responses, each message is generated before any message from the round is added to the communication transcript, leaving the separate final decision maker with the payment reports and the message from the participant without a PIU. The trajectory was selected after observing whether the two episodes succeeded and does not establish this mechanism; the ordering result for the legal problem families rests on the problem-instance–model comparison above.

G.7.2 Simultaneous responses with All-but-one PIU concentration

The negative Multi-Agent Full-Context estimate for All-but-one PIU concentration with Simultaneous responses in Table 35 is an interaction estimate, not an absolute episode success rate. It compares the difference in episode success rate between Simultaneous responses and Coordinator-selected speaker

across two PIU distributions. Relative to One PIU per participant, All-but-one PIU concentration shifts this comparison by 4.0 percentage points toward Coordinator-selected speaker (standard error 1.5). Equivalently, the difference in episode success rate between Simultaneous responses and Coordinator-selected speaker is 4.0 points lower under All-but-one PIU concentration. The corresponding Multi-Agent PIU-Only estimate is -0.4 percentage points (standard error 1.6), and its 95% confidence interval includes zero.

The definitions of the two conditions suggest a possible explanation. Under All-but-one PIU concentration, one participant receives all but one of the PIUs and another receives the remaining PIU. Under Simultaneous responses, all participants send a message in one round, and each message is generated before any message from that round is added to the communication transcript. In Multi-Agent Full-Context, the participant receiving all but one of the PIUs must retrieve them from its initial local observation and communicate them in one message. Neither that participant nor the participant receiving the remaining PIU can condition its message on the other's message. Concentrating the PIUs therefore reduces the potential benefit of collecting messages simultaneously from different participants and leaves the separate final decision maker to combine the independently generated messages.

This account is consistent with the corresponding estimate in Multi-Agent PIU-Only having a 95% confidence interval that includes zero, where participants' initial local observations contain PIUs alone rather than source material containing them. It remains an interpretation: the regression does not isolate retrieval or the inability to condition on another message as a causal mechanism.

G.8 Legal Token-Budget Ablations: Complete Results and Diagnostics

For space, Section 4.5 summarizes the principal findings and defers the complete token-budget analyses to this appendix. We preserve the full results presentation below, followed by additional implementation details, budget-binding diagnostics, and results by model and source corpus.

G.8.1 Complete Results Deferred from Section 4.5

We study private-note, message, and local-transcript token budgets under Multi-Agent Full-Context with One PIU per participant and Coordinator-selected speaker. For a problem instance with N participants, every budget setting permits N participant messages before the final decision. We hold the coordinator policy and final-decision token budget fixed.

Private-note and message token budgets. Each participant message is generated in two stages. The participant first forms a private working note from its initial local observation and local transcript. A second stage receives the note and the same local transcript and produces the message added to the communication transcript; the note is not shared. The message token budget makes the departure from A3 explicit. Because participants receive source material containing their PIUs rather than the PIUs alone, they must identify the information needed for a correct final decision before producing bounded messages, a requirement related to A2. We cross private-note token budgets of 64, 128, and 256 tokens with message token budgets of 32, 64, 128, and 256 tokens. Table 39 reports GPT-5.5 results over 24 CUAD and 29 Multi-LexSum problem instances; Section G.8.2 provides the complete implementation and results by model and source corpus.

Local-transcript token budgets. We separately limit the newly available message text in each participant's local transcript. This limit is motivated by work on limited human attention and working memory (Kahneman, 1973; Cowan, 2001), without assuming that language models share the same cognitive mechanisms. We compare first-in, first-out (FIFO) and random contiguous-window reading with GPT-5.5 on the 29 Multi-LexSum problem instances. Section G.8.3 provides the complete implementation and reports how often the token budgets bind.

Table 39: **Private-note and message token budgets.** GPT-5.5 episode success rates (%) over 53 problem instances; small ± terms are standard errors.

Private-note tokens	Message tokens 32	64	128	256
64	45.3 ± 6.8	67.9 ± 6.4	71.7 ± 6.2	69.8 ± 6.3
128	60.4 ± 6.7	69.8 ± 6.3	81.1 ± 5.4	75.5 ± 5.9
256	60.4 ± 6.7	67.9 ± 6.4	83.0 ± 5.2	81.1 ± 5.4

Figure 10: **Local-transcript token budgets.** The dashed line marks the unlimited baseline; shaded bands show uncertainty.

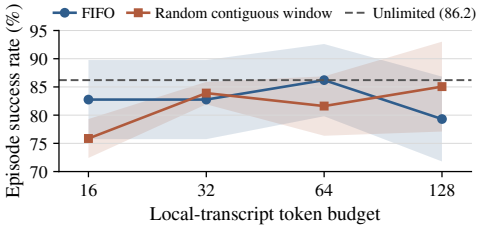


Table 39 shows the largest gains as the message token budget increases from 32 to 128 tokens. Averaged across private-note token budgets, episode success rate rises from 55.3% to 78.6%, but does not improve further with 256 message tokens; variation across private-note token budgets is smaller than variation across message token budgets.

Figure 10 shows no monotonic relationship between the local-transcript token budget and episode success rate. FIFO remains within 6.9 percentage points of the unlimited baseline, whereas the random-window mean at 16 tokens is 10.3 points lower. Thus, these problem instances do not show a uniform performance penalty from a smaller local-transcript token budget.

G.8.2 Private-Note and Message Token Budgets

For a problem instance with N participants, Coordinator-selected speaker permits N participant messages followed by the coordinator's final decision. Each participant message uses two model generations with the same local transcript. The first generation receives the participant's initial local observation and produces a private working note. The second receives that note, but not the initial local observation, and produces the message added to the communication transcript. The private working note is not added to the communication transcript or shown to another decision maker.

The experiment crosses private-note token budgets of 64, 128, and 256 tokens with message token budgets of 32, 64, 128, and 256 tokens. The token budget for each participant's initial local observation is 40,000 tokens, and participant access to the communication transcript is not limited. The coordinator policy and final-decision token budget remain fixed. The 256-token message condition is the largest evaluated message token budget within each private-note row; the grid does not include an unlimited-message condition.

The imposed limits bind as intended. GPT-5.5 participant messages are truncated in approximately 93%, 57%, 23%, and less than 1% of episodes as the message token budget increases, while private-note truncation falls from 82% to 4%. Qwen3-32B does not show a consistent relationship between either token budget and episode success rate. Table 40 reports the source-corpus-specific results for both models.

G.8.3 Local-Transcript Token Budgets

When a participant is selected to send a message, its current directed coordinator query is read first and charged to the local-transcript token budget. Any remaining capacity is used to read previously unread message text. FIFO reading exposes the earliest unread tokens. Random contiguous-window reading instead selects one contiguous window from the previously unread message text, reproducibly for a fixed problem instance, participant, message, and seed. Unread message text remains available when that participant is next selected. The participant also retains its initial local observation, its prior messages, and every local-transcript excerpt it has previously read. The coordinator's observation is not limited and therefore contains the complete communication transcript.

We fix the private-note and message token budgets at 256 and 128 tokens and compare local-transcript token budgets of 16, 32, 64, and 128 tokens with unlimited reading. A 256-token condition provides an additional check at the largest evaluated token budget. FIFO is evaluated once and random contiguous-window reading is evaluated with three seeds. Accordingly, Table 41 reports binomial

Table 40: **Private-note and message token budgets by source corpus and model.** Episode success rates (%); small ± terms are binomial standard errors over 24 CUAD or 29 Multi-LexSum problem instances.

Private-note tokens	Message tokens	GPT-5.5 CUAD	GPT-5.5 Multi-LexSum	Qwen3-32B CUAD	Qwen3-32B Multi-LexSum
64	32	70.8 ± 9.3	24.1 ± 7.9	79.2 ± 8.3	3.4 ± 3.4
64	64	70.8 ± 9.3	65.5 ± 8.8	75.0 ± 8.8	0.0 ± 0.0
64	128	70.8 ± 9.3	72.4 ± 8.3	91.7 ± 5.6	3.4 ± 3.4
64	256	70.8 ± 9.3	69.0 ± 8.6	83.3 ± 7.6	0.0 ± 0.0
128	32	79.2 ± 8.3	44.8 ± 9.2	75.0 ± 8.8	3.4 ± 3.4
128	64	79.2 ± 8.3	62.1 ± 9.0	75.0 ± 8.8	6.9 ± 4.7
128	128	79.2 ± 8.3	82.8 ± 7.0	83.3 ± 7.6	3.4 ± 3.4
128	256	87.5 ± 6.8	65.5 ± 8.8	75.0 ± 8.8	3.4 ± 3.4
256	32	79.2 ± 8.3	44.8 ± 9.2	75.0 ± 8.8	0.0 ± 0.0
256	64	66.7 ± 9.6	69.0 ± 8.6	79.2 ± 8.3	0.0 ± 0.0
256	128	83.3 ± 7.6	82.8 ± 7.0	83.3 ± 7.6	3.4 ± 3.4
256	256	79.2 ± 8.3	82.8 ± 7.0	70.8 ± 9.3	3.4 ± 3.4

standard errors over the 29 fixed problem instances for FIFO and standard deviations across the three episode-success rates for random reading.

Table 41: **Local-transcript token budgets.** Episode success rates (%) and changes from the unlimited FIFO baseline. Small ± terms are binomial standard errors for FIFO and standard deviations over three seeds for random contiguous-window reading.

Local-transcript tokens	FIFO	Change from baseline	Random contiguous window	Change from baseline
16	82.8 ± 7.0	-3.4	75.9 ± 3.4	-10.3
32	82.8 ± 7.0	-3.4	83.9 ± 2.0	-2.3
64	86.2 ± 6.4	+0.0	81.6 ± 5.3	-4.6
128	79.3 ± 7.5	-6.9	85.1 ± 8.0	-1.1
256	86.2 ± 6.4	+0.0	82.8 ± 6.9	-3.4
Unlimited	86.2 ± 6.4	0.0	—	—

The local-transcript token budget binds at every participant message with budgets of 16 and 32 tokens and at approximately 95%, 90%, and 83% of participant messages with budgets of 64, 128, and 256 tokens. At 16 tokens, the coordinator query consumes the complete local-transcript token budget, so FIFO and random contiguous-window reading expose the same prefix; their observed difference reflects run-to-run model variation, not a different selected excerpt. The 256-token condition reaches 86.2% under FIFO and 82.8 ± 6.9% under random reading. Together with the non-monotonic results in Figure 10, these diagnostics do not support a monotonic relationship between the local-transcript token budget and episode success rate on these problem instances.

G.9 Trajectory Examples

We present two representative episode trajectories that illustrate how PIUs move from participants' initial local observations into the communication transcript and final decision.

G.9.1 Dependency Case and Episode Trajectory: CVS 2022

Problem and dependency structure. This representative FinLongDocQA problem asks for the ratio of CVS Health's unique digital customers to 2022 revenue in each of three segments, expressed as millions of customers per $1 billion of revenue, and then asks which segment has the largest ratio. Under the One PIU per participant condition, Participant A holds the common customer count, while Participants B–D each hold one segment denominator.

PIU ID	PIU	Participant assigned the PIU	Answer components using the PIU
1	47M unique digital customers	Participant A	Q1, Q2, Q3, Q4
2	Health Care Benefits revenue: $91.4B	Participant B	Q1, Q4
3	Pharmacy Services revenue: $169.2B	Participant C	Q2, Q4
4	Retail revenue: $106.6B	Participant D	Q3, Q4

The PIUs used by the four answer components are therefore

$$Q_1 = \{1, 2\}, \qquad Q_2 = \{1, 3\}, \qquad Q_3 = \{1, 4\}, \qquad Q_4 = \{1, 2, 3, 4\}.$$

Counting references component by component gives ten PIU uses, but only four distinct PIUs. No participant can compute even one requested ratio alone, and the final comparison requires all four PIUs.

Reference calculation. Using the same 47M company-wide customer count as the numerator for each requested segment calculation gives

$$\frac{47}{91.4} = 0.51, \qquad \frac{47}{169.2} = 0.28, \qquad \frac{47}{106.6} = 0.44.$$

Thus Health Care Benefits has the highest ratio, at 0.51M unique digital customers per $1B of revenue.

Observed success by evaluation condition. The table below reports the result for this problem instance under each model and evaluation condition. A check means that all four answer components were satisfied; a cross means that all four were missing in the final decision.

Model	Single-Agent All-PIU	Multi-Agent PIU-Only	Multi-Agent Full-Context
Claude Sonnet 4.5	✓	✓	✓
Claude Haiku 4.5	✓	×	×

Successful episode trajectory: Sonnet 4.5. In Single-Agent All-PIU, the model directly read the four facts, computed 0.51, 0.28, and 0.44, and selected Health Care Benefits. In both multi-agent conditions, all four PIUs entered the communication transcript in the responses to the first four coordinator queries: A disclosed 47M; B disclosed $91.4B; C disclosed $169.2B; and D disclosed $106.6B. The coordinator used the same company-wide customer count in all three ratios, performed the divisions, and returned the same ranking. The semantic judge found all four answer components correct and grounded in all three conditions.

Failure episode trajectory 1: Haiku 4.5, Multi-Agent PIU-Only. The coordinator issued 15 queries to A–D, but every participant response was empty. No PIU entered the communication transcript. The final decision explicitly refused to calculate because it lacked the customer count and segment revenues. The final decision consequently omitted all four answer components. This is a communication failure rather than an arithmetic failure: the same model solved the problem in Single-Agent All-PIU.

Failure episode trajectory 2: Haiku 4.5, Multi-Agent Full-Context. Participant A found the consolidated 47M customer count, but initially reported that the needed segment table was unavailable. Participant C then incorrectly attached $169.2B to Health Care Benefits and could not recover the other two denominators; later participant messages converged on the claim that the accessible excerpts were insufficient. After 15 coordinator queries, the coordinator returned "unable to complete" instead of the three ratios. The final decision omitted all four answer components. Here the source material created an evidence-retrieval and fact-mapping failure even though every PIU remained present in the source material provided to its assigned participant.

Case-study implication. The evaluation holds the problem instance fixed while exposing two distinct collaboration bottlenecks. Multi-Agent PIU-Only can fail because a PIU in a participant's

initial local observation never enters the communication transcript; Multi-Agent Full-Context can fail because relevant facts are lost or misassigned within much longer source material. In Single-Agent All-PIU, both models solve the problem instance when the PIUs are centralized.

G.9.2 Composite-Problem Episode Trajectory: Kirkland's 2023

Generation is not execution. This representative FinDVer problem contains six questions, but Q1–Q5 are not generated during the episode and Q6 is not appended afterward. All six questions, their links to PIUs, the reference decision, and the PIU distribution are frozen together as one composite problem instance before evaluation. The episode tests whether a coordinator can recover and combine that already-fixed evidence.

> **Data generation:** 10-K source → candidate units → six PIUs → Q1–Q5 analyses and Q6 synthesis → verification of the links between answer components and PIUs → PIU distribution → retained problem instance.
>
> **Evaluation execution:** initial common observation → coordinator queries → evidence accumulates in the communication transcript → Q1–Q5 and Q6 final decision → answer-component correctness and grounding judgment.

The construction question is therefore "which facts must jointly determine the reference decision?"; the evaluation question is "can the model find those facts and combine only what the filing supports?"

From filing disclosures to one analyst problem. The six-PIU problem joins four review axes that an equity or credit analyst would normally consider together.

Review axis	Filing evidence	Composite item
Supply chain	Sourcing diversification and country-of-origin mix	Q1
Profitability	310-bp gross-margin improvement and drivers	Q2
Execution risk	Capital constraints and tightly managed cash flow	Q3
Financing	Term-loan terms, balance-date status, later draws	Q4–Q5
Integrated judgment	Operating improvement plus liquidity pressure	Q6

Q6 asks whether the complete narrative is fully supported: Kirkland's was improving operations through sourcing diversification and gross-margin improvement, while still facing liquidity pressure and using new financing as it executed a financial turnaround strategy. This is not a vote over Q1–Q5; it combines evidence with different scopes and dates into one bounded filing interpretation.

Distributed evidence discovery, queries 1–6. Every participant had the generic participant description "financial filing context reviewer," so the coordinator initially did not know the PIU-ownership mapping.

Query	Participant	Item	Evidence recovered
1	A	Q4	No term-loan balance at Feb. 3; $34M credit borrowing
2	B	Q1	Diversification and fiscal 2023 country mix
3	C	Q3, Q5	Tight cash management; later $6M and $5M draws
4	D	Q3	Capital shortage could limit strategic execution
5	E	Q4	$12M term loan; at least $5M initial draw required
6	F	Q2	310-bp margin improvement, drivers, and offset

Query 1 illustrates genuine discovery rather than replay of a reference-decision ordering. The coordinator asked A for sourcing information, but A did not own that PIU. A instead disclosed its financing evidence. The coordinator used the updated communication transcript to query another reviewer for the still-missing sourcing component, and recovered all six PIUs by query 6.

Cross-source consistency checks, queries 7–15. Once the decision-relevant facts were present, the remaining questions tested how strongly they could be connected. F confirmed that its margin paragraph did not itself call the improvement part of the financial turnaround. E confirmed that its term-loan-terms paragraph did not state a strategic use of proceeds. By contrast, C's paragraph jointly disclosed later borrowing, tightly managed cash flow, and execution of the financial turnaround strategy. The defensible synthesis is therefore that the company used additional financing while executing the turnaround under cash pressure–not that a particular loan directly caused or funded a particular operating improvement.

Successful three-stage episode trajectory: Sonnet 4.5, Multi-Agent PIU-Only. 1. Distributed evidence discovery: queries 1–6 mapped the six anonymous reviewers to sourcing, margin, capital-risk, term-loan, balance-date, and post-year-end evidence. **2. Checking support for cross-passage conclusions:** queries 7–15 distinguished direct filing statements from unsupported cross-passage inferences and preserved the February 3 balance-date versus subsequent-borrowing distinction. **3. Integrated conclusion:** the final decision reported Q1–Q5 and concluded that Q6 was fully supported. The episode used the complete budget of 15 participant responses, and all six answer components were correct and grounded.

Efficiency and interpretation. The episode trajectory is deliberately informative rather than optimal. An initial misdirected query was necessary because PIU ownership was hidden, whereas several queries after query 6 repeated checks of whether the filing supported the proposed cross-passage conclusions after the core facts were already complete. For presentation, the 15 coordinator queries and responses are most clearly compressed into the three stages above. The example then shows more than parallel subproblem solving: it shows distributed evidence recovery, verification of permissible cross-source links, and construction of one realistic operational-and-liquidity judgment.

H Literature Motivation for the PIU-Distribution–Communication-Structure Interaction

Section 4.4 tests whether differences in episode success rate between communication structures change across PIU distributions. Three complementary literatures motivate this comparison and the interaction specification.

Team decision-making under a decentralized information structure. Team decision theory studies decision makers who pursue a shared objective but have different information available to them (Radner, 1962; Marschak & Radner, 1972; Nayyar et al., 2013). This is the basic structure of our evaluation conditions with multiple participants: the participants share the episode objective, while PIU distribution determines which PIUs are available to which participants. Consistent with this formulation, we hold the problem instance and model fixed while varying PIU distribution. The literature does not imply that a particular PIU distribution must perform best, because performance also depends on whether the distributed PIUs enter communication and support a correct final decision.

Communication under a decentralized information structure. Communication complexity and distributed-function computation study how separately informed parties exchange the information needed to compute a shared target (Yao, 1979; Orlitsky & Roche, 2001). Work on knowledge in distributed systems likewise formalizes communication as changing the information available to decision makers (Halpern & Moses, 1990). In our setting, communication structure determines who may send a message at each point in the episode and which earlier messages that participant can observe, while PIU distribution determines where the relevant information is initially available. We study PIU distribution and communication structure jointly because differences in episode success rate among communication structures may vary across PIU distributions.

PIUs entering communication. Hidden-profile experiments show that information observed by only one participant may fail to enter group discussion (Stasser & Titus, 1985). This evidence is directly relevant to PIU distributions that place a PIU with only one participant or concentrate multiple PIUs with one participant. It also explains why the proportion of participants who send a message is

not by itself sufficient: episode success depends on whether the communication structure elicits and preserves the PIUs needed for a correct final decision.

Implications for the empirical specification. The specification includes PIU-distribution and communication-structure main effects and their interaction, allowing differences among communication structures to vary across PIU distributions. Because both are represented as categorical conditions and every non-reference pair receives an interaction term, the specification does not impose that their relationships with episode success are independent. The Bernoulli model and logit link reflect that episode success is binary, while problem-family and model fixed effects adjust for differences across problem families and models. Standard errors are clustered by problem instance, and Appendix G.6 reports the stricter specification with problem-instance fixed effects. The analysis estimates empirical relationships among the evaluated conditions; these literatures motivate the comparison but do not make the interaction estimates causal tests of any one theory.

I Policy-Training Data, Evaluation, and Diagnostic Analyses

This appendix documents the data, evaluation, training configurations, and diagnostic analyses for the two policy-training demonstrations in Section 5. We first describe the CUAD-derived and MAUD-derived free-form decisions and their evaluator. We then report the demonstration under Coordinator-selected speaker and the demonstration under Current-speaker selection, with the participant that ends communication making the final decision.

Relation to assumptions A1, A2, and A3. Both demonstrations address A1 by training and comparing language-model policy collections rather than assuming that a jointly optimal policy collection is available before execution. The Coordinator-selected speaker demonstration additionally departs from A2 because participants derive messages from source material, whereas the Current-speaker selection demonstration provides PIUs directly. Both demonstrations depart from A3 by enforcing explicit observation and communication budgets.

I.1 Free-Form Legal Decisions and Evaluation

I.1.1 Construction and Data Splits

We construct CUAD-derived and MAUD-derived problem instances from the corresponding distributed CUAD and MAUD problem instances while preserving the PIU distribution and source material. For each input problem instance, GPT-5.5 receives the original decision task, the reasoning supporting its reference decision, the PIUs and their source text, and the PIU distribution. It produces a standalone legal question, a source-grounded reference decision, PIU-grounded answer components, a component requiring synthesis across all PIUs, material contradictions that disqualify a final decision, and problem-specific evaluation criteria. The question identifies the dimensions to resolve without exposing the original decision alternatives or their labels.

A separate GPT-5.5 call first checks the proposed question, reference decision, and evaluation criteria against the source evidence. The check rejects disguised multiple choice, reference-decision leakage, questions answerable from one PIU, unsupported content, incomplete PIU coverage, and criteria that would accept a partial or vague final decision. Deterministic checks then enforce the length and format requirements, require at least one answer component grounded in every PIU and one component synthesizing all PIUs, and verify that the PIU distribution is unchanged. If either stage fails, the proposal is revised using the resulting findings. Of 396 retained problem instances, 283 pass on the first draft; the remaining instances require up to seven revision rounds.

Across the 396 problem instances, the PIU count has median 2, mean 2.13, and range 2–6. Questions contain 47 words at the median, and reference decisions contain 100 words at the median. On the 62 CUAD-derived validation and MAUD-derived evaluation instances, the evaluator receives a median of six answer components: five grounded in individual PIUs and one requiring cross-PIU synthesis. All 62 reference decisions receive an LLM-judged score of 1.0. By contrast, 62 single-PIU partial decisions and 62 original incorrect decision alternatives have a mean score of 0.193. Policies never observe the reference decisions, answer-component annotations, PIU identifiers, construction checks, or evaluation criteria.

Table 42: **Free-form legal problem instances.** Only CUAD-derived training problem instances supply gradients. CUAD-derived validation contracts are absent from training, and MAUD-derived problem instances are opened only after CUAD-derived checkpoint selection.

Use	CUAD-derived	MAUD-derived	Total
Training	334	0	334
Validation	36	0	36
MAUD-derived evaluation	0	26	26

I.1.2 GPT-5.5 Evaluator

Free-form decisions admit multiple valid surface forms, so the evaluator compares a candidate final decision with the reference decision, PIUs, answer components, problem-specific evaluation criteria, and material contradictions. GPT-5.5 directly assigns an *LLM-judged score* in $[0, 1]$. The implementation clips the returned value to this interval and assigns zero when the score is unavailable. The arithmetic mean across problem instances is the reported LLM-judged score and the only semantic training reward. The evaluation criteria require the final decision to reach the reference conclusion, cover the required content, satisfy the applicable numeric and legal conditions, and contain no disqualifying error. A malformed or missing judge response receives a score of zero. All evaluations use the AWS Bedrock-hosted GPT-5.5 at temperature zero with the same prompt and normalization procedure.

I.2 Coordinator-selected speaker: Participant and Coordinator Policy Training

All episodes in this subsection use Multi-Agent Full-Context with Coordinator-selected speaker. A shared participant policy generates bounded messages from each selected participant's initial local observation, while a coordinator policy generates coordinator queries and the final decision. Each initial local observation has a 28,000-token budget; participant messages have a 48-token budget within a 64-token structured completion; coordinator queries have a 64-token budget; and the final decision has a 256-token budget within a 384-token structured completion. Participants do not generate private notes.

I.2.1 Optimization and Training Configurations

Each update samples eight shuffled CUAD-derived training problem instances and eight rollouts per problem instance, yielding 64 episode rollouts. GRPO standardizes the eight rewards within each problem instance and broadcasts the resulting episode advantage to the trainable actions. An invalid trainable action instead receives a local advantage of -0.1 and one retry. We update all parameters of each trainable policy with AdamW for one epoch per update using a PPO-style clipped surrogate, ratio clip 0.2, dual-clip bound 3.0 for negative advantages, sequence-mean/token-mean loss aggregation, learning rate 10^{-6}, weight decay 0.01, and a 10^{-3} KL loss relative to the base Qwen2.5-7B-Instruct policy. Training runs for 80 updates, validation occurs every 10 updates with greedy decoding, and each configuration uses four H200 GPUs.

Table 43: **Training configurations under Coordinator-selected speaker.** The participant-only stage and coordinator-only continuation form one two-stage policy-training experiment. Joint two-policy training begins independently from the base policy for both trainable policies.

Setting	Participant-only stage	Coordinator-only continuation	Joint two-policy training
Updated policy	Participant policy	Coordinator policy	Participant and coordinator policies
Fixed policy	Base coordinator policy	Trained participant policy	Neither

The LLM-judged score is the training reward in all configurations. Which participant the coordinator queries, query order, which PIUs appear in participant messages, and whether the coordinator queries every participant do not enter the reward.

I.2.2 Two-Stage Policy Training

The participant-only stage updates the participant policy while the base coordinator policy remains fixed. We evaluate the participant policy every 10 updates; the version after update 60 is the first to attain the highest mean LLM-judged score on CUAD-derived validation and is therefore used in the coordinator-only continuation. This continuation initializes the coordinator policy from the base Qwen2.5-7B-Instruct policy and holds the trained participant policy fixed. The coordinator policy selected after this stage is the update-80 checkpoint.

Table 44: **CUAD-derived validation results during two-stage policy training.** Values are percentages.

Training stage	LLM-judged score
Base policies	12.14
Participant-only stage	16.84
Coordinator-only continuation	**23.33**

The participant-policy comparison uses episodes generated by the base and trained participant policies under the same fixed coordinator policy and decoding configuration. Table 45 shows that training causes participant messages to include information needed for a correct final decision and changes which PIUs appear across participant messages. In the insurance-obligations example, the trained participant policy produces a different first message. Although the coordinator policy remains fixed, it consequently observes a different transcript and queries participant B instead of repeating participant A.

Table 45: **Illustrative changes after participant-only training.** Only the participant policy changes. Values are LLM-judged scores; the examples are not frequency estimates and do not enter training or checkpoint selection.

Case	Base participant policy	Trained participant policy	Observed change
Liability limits	Both participant messages deny that the agreement contains liability limits; the final decision receives 0.00.	One participant message includes the mutual exclusion of lost profits and indirect, special, consequential, and punitive damages, including the product context and even-if-advised condition; the final decision receives 0.33.	Participant message includes information needed for a correct final decision
Compliance audit and access	Participant messages leave the right holder unspecified and substitute unrelated consumer-credit services; the final decision receives 0.00.	The two participant messages separately include Equidata's at-any-time site-inspection right and its compliance audit at Equidata's expense; the final decision receives 0.33 but still merges the two rights.	Different PIUs appear across participant messages
Insurance obligations	The first participant message includes the $1M coverage, but the second coordinator query repeats participant A and obtains an unrelated fiber-rights clause; the final decision receives 0.40.	After the trained participant policy produces a different first message, the fixed coordinator observes the changed transcript and queries participant B, which supplies the cancellation, non-renewal, modification, and 30-day notice condition; the final decision receives 0.80.	Changed participant message leads to a different next coordinator query

The coordinator-policy comparison holds the trained participant policy fixed. Table 46 shows one example in which changed coordinator queries cause participant messages to include both the certificate-on-request clause and a reciprocal insurance obligation. In another, the trained coordinator policy produces the correct final decision from participant messages that are substantively unchanged. These examples separate changes arising from participant messages from changes arising through coordinator queries or the final decision.

On the 26 MAUD-derived problem instances held out from training and checkpoint selection, two-stage policy training does not improve the aggregate LLM-judged score.

Table 46: **Illustrative changes after the coordinator-only continuation.** Only the coordinator policy changes; the trained participant policy remains fixed. Values are LLM-judged scores; the examples are not frequency estimates and do not enter training or checkpoint selection.

Case	Base coordinator policy	Trained coordinator policy	Observed change
Distributor insurance	Participant messages include unrelated regulatory and committee-cost facts; the final decision receives 0.00.	Changed coordinator queries cause participant messages to include both the certificate-on-request clause and the reciprocal insurance obligation; the final decision receives 1.00.	Changed coordinator queries
Co-brand minimum commitment	Both participant messages contain the 5,000-unit initial order and the 10,000-unit/32-month guarantee, but the final decision confuses their roles; the final decision receives 0.33.	With substantively unchanged participant messages, the final decision correctly separates the initial order from the longer-term guarantee and receives 1.00.	Changed final decision from the same participant messages

I.2.3 Joint Two-Policy Training

Joint two-policy training removes weight sharing between the participant and coordinator policies. Both policies are initialized independently from Qwen2.5-7B-Instruct, maintain separate optimizer states and clipped-surrogate losses, and receive the same team payoff. Every update has a nonzero gradient norm for each policy, ruling out the failure mode in which one shared policy generates or updates both participant and coordinator actions.

The reported run uses random seed 29, 80 updates, and 5,120 episode rollouts. The update-80 pair attains the highest mean LLM-judged score among the scheduled CUAD-derived validation evaluations. The score increases from 11.97% to 22.38%. The coordinator queries every participant on 100% of validation problem instances by update 20 and maintains that rate through update 80.

The coordinator-query diagnostics use one greedy rollout per CUAD-derived validation problem instance. A first query copies the problem question only when the query and question match after normalization for case, punctuation, and whitespace. A later query is lexically reply-conditioned when it contains an explicit follow-up marker or at least two terms from a preceding participant message that are absent from the problem question. This diagnostic does not constitute a causal test of adaptation. The trained coordinator queries every participant, uses short queries, and does not copy the problem question in the first query; lexically reply-conditioned later queries remain infrequent. Thus, the learned coordinator-query behavior is distinguished by querying every participant with short queries, rather than by a high rate of lexically reply-conditioned later queries.

In one CUAD-derived validation example, the base participant and coordinator policies elicit only the parties and the Type I material label and produce a one-directional final decision with an LLM-judged score of 12.5%. The jointly trained policies instead elicit the IBM-to-Customer Type II permissions from one participant and the Customer-to-IBM downstream-authorization right for Type I materials from another. Their final decision represents both grant directions and receives 62.5%, although it remains incomplete. This example illustrates coordinated evidence decomposition but does not establish that the interaction between the two trained policies is necessary.

Under the jointly trained participant and coordinator policies, the coordinator queries every participant and every participant sends a public message on all 26 MAUD-derived problem instances; both behaviors occur on 20 of 26 instances under the base participant and coordinator policies. The LLM-judged score changes only slightly, from 5.12% to 5.58%. The query and messaging patterns therefore transfer more clearly than task-specific evidence use or final-decision quality.

I.3 Current-speaker selection: Shared Participant Policy Training

I.3.1 Problem Instances and Communication Structure

This experiment uses Multi-Agent PIU-Only with One PIU per participant under Current-speaker selection. The participant that ends communication makes the final decision; there is no separate final decision maker. Training contains 350 FinDVer problem instances with two to eight participants and 188 FinLongDocQA problem instances with two or three participants. Validation contains

39 FinDVer and 21 FinLongDocQA problem instances. The held-out test set contains 40 four-participant FinLongDocQA problem instances. Canonical task identifiers and source-material groups do not overlap across splits. Four-participant FinLongDocQA is absent from training, although four-participant FinDVer occurs in training.

Each participant receives exactly one PIU in its initial local observation. The initial common observation contains the decision objective, decision format, and a directory of participant identifiers and public participant descriptions. The descriptions are short role names generated from the broad subject of the assigned PIU. Deterministic checks and a separate leakage check reject names that expose a value, polarity, condition, conclusion, or decision implication. A prompt-construction check of all 638 normalized instances and 4,988 rendered prompts confirms that every directory contains the permitted participants and that no participant receives another participant's PIU.

In each round, the current speaker sends a public message and either selects the next participant or ends communication and makes the final decision. If the 15-round communication budget is exhausted, the last speaker makes the final decision. Public messages have a 600-token budget, the final-decision field has a 1,200-token budget, participant actions have 1,200 private reasoning tokens, and the final decision has 1,800 private reasoning tokens. The runtime also applies a 750-token structured-generation cap, a 16,384-token prompt cap, and a 24,576-token training-sequence cap.

I.3.2 Optimization and Checkpoint Selection

A single Qwen2.5-7B-Instruct participant policy is shared across participants. Participant messages, next-participant selection, decisions to end communication, and the final decision all receive participant-policy gradients. Each update samples eight problem instances and 16 rollouts per instance, yielding 128 rollouts per update and 10,240 rollouts across 80 updates. Rollouts use temperature 0.7 and top-$p = 0.95$; validation uses greedy decoding.

Training updates all parameters of the shared participant policy using FSDP in bfloat16 on four H200 GPUs. GRPO standardizes the 16 rewards within each problem instance. Invalid JSON, schema-invalid actions, and invalid next-participant identifiers receive a local advantage of -0.1 and one retry. Optimization uses AdamW with learning rate 10^{-6}, weight decay 0.01, gradient-norm clip 1.0, PPO ratio clip 0.2, dual-clip bound 3.0 for negative advantages, and sequence-mean/token-mean loss aggregation. This run does not use a KL loss.

Checkpoint selection uses the mean LLM-judged score on the 39 FinDVer validation problem instances, with the earlier update winning a tie. The 21 FinLongDocQA validation problem instances are evaluated on the same schedule but do not enter checkpoint selection. The selected update is update 70. Main-paper Table 6 reports the separate post-selection greedy evaluations of the base and trained participant policies.

I.3.3 Communication Diagnostics

Final-decision completion records whether an episode produces a final decision. "Every participant sends a message" records whether every participant appears at least once as a public-message sender. Participant coverage is the number of distinct public-message senders divided by team size, and the number of unique speakers is its unnormalized counterpart. Self-handoff rate is the fraction of valid handoffs in which the current speaker selects itself as the next participant. These diagnostics use one greedy rollout per problem instance and retain valid public actions produced before an episode failure. None enters the training reward.

I.3.4 Separating the Communication Trajectory from the Final-Decision Policy

The cross-replay analysis separates the communication trajectory from the final-decision policy. It uses the 40 validation problem instances for which both the base and trained communication trajectories produce a final decision. For each trajectory, we freeze the communication transcript, the identity and assigned PIU of the participant making the final decision, and the final-decision prompt, then regenerate the final decision with both the base and trained participant policies.

Let Y_{BB}, Y_{BT}, Y_{TB}, and Y_{TT} denote the scores from crossing the base and trained communication trajectories with the base and trained final-decision policies. We define the communication-trajectory

effect as

$$\frac{1}{2}\big[(Y_{TB} - Y_{BB}) + (Y_{TT} - Y_{BT})\big]$$

and the final-decision-policy effect as

$$\frac{1}{2}\big[(Y_{BT} - Y_{BB}) + (Y_{TT} - Y_{TB})\big].$$

Intervals are percentile 95% intervals from 100,000 paired problem-instance-bootstrap samples.

On validation, the communication-trajectory effect is 6.95 points with a 95% interval of $[0.49, 13.78]$. The final-decision-policy effect is -0.20 points with an interval of $[-0.99, 0.61]$, and is not statistically distinguishable from zero. The cross-replay cell means represent policy-conditioned regenerated decisions rather than deterministic reproductions of the original final-decision strings.

Illustrative trajectory pair. In one three-participant FinLongDocQA validation example, the base trajectory exhausts all 15 rounds, contains seven self-handoffs, and uses an incorrect flooring amount; both final-decision policies score 35% on that transcript. The trained trajectory contains no self-handoffs, voluntarily ends communication, and recovers the relevant amounts; both final-decision policies score 100%. This example illustrates how a changed communication trajectory can improve the decision while holding the final-decision policy fixed.

I.3.5 Additional Analysis of the Communication Trajectories

The completion-conditioned cross-replay excludes problem instances for which either original trajectory does not produce a final decision. We therefore run a second analysis on all 39 paired FinDVer validation problem instances and all 40 paired FinLongDocQA test problem instances. One GPT-5.5 final-decision policy receives the task and the valid communication transcript from each base and trained trajectory. It does not receive the original final decision, the participants' assigned PIUs, original score, or an indication of which policy produced the transcript. All incomplete episodes end after an invalid participant response; for these, the input ends with the last valid public message. We evaluate the resulting decision with the same LLM-judged score used in the original evaluation.

We separately show GPT-5.4 the task, reference answer, answer components, reference PIUs, and communication transcript, without the policy identity, original final decision, or original score. For every PIU, the evaluator records public messages that communicate it and separately records messages that communicate an incompatible value, unit, period, sign, or scope. It also records whether each answer component is fully supported by public messages. Exact repetition is computed directly after case-folding and whitespace normalization and does not use an evaluator.

Table 47: **Communication-trajectory analysis.** Values are means across problem instances. Percentages are shown except for public messages. The same 39 FinDVer validation and 40 held-out FinLongDocQA problem instances are paired across policies.

	FinDVer validation			FinLongDocQA test		
Measurement	Base	Trained	Δ	Base	Trained	Δ
Same final-decision policy: LLM-judged score	34.10	46.72	+12.62	49.60	44.82	−4.78
Public messages	7.28	12.46	+5.18	9.30	13.43	+4.12
PIUs appearing in the communication transcript (%)	38.0	53.9	+15.9	63.1	63.1	0.0
PIUs appearing in the communication transcript with no incompatible message (%)	23.3	26.3	+3.0	55.6	58.8	+3.1
Answer components fully supported (%)	23.5	33.0	+9.5	26.2	25.0	−1.2
PIUs communicated incompatibly at least once (%)	30.6	48.0	+17.4	26.9	25.6	−1.2
Exactly repeated public messages (%)	18.9	50.9	+31.9	23.1	48.6	+25.5

For each measurement, the reported change is the mean within-problem-instance trained-minus-base difference. Percentile 95% intervals use 100,000 paired problem-instance bootstrap samples with seed 20260922. Under the same final-decision policy, the score changes by 12.62 points on FinDVer ($[4.38, 21.59]$) and by -4.78 points on FinLongDocQA test ($[-17.75, 8.20]$). On FinDVer, the

increase in the proportion of PIUs appearing in the communication transcript is 15.9 points with an interval of $[4.4, 28.1]$, and the increase in fully supported answer components is 9.5 points with an interval of $[0.4, 19.2]$. On FinLongDocQA test, the corresponding changes are 0.0 points ($[-11.2, 11.2]$) and -1.2 points ($[-12.5, 10.4]$). Exact repetition increases by 31.9 points ($[21.7, 42.1]$) on FinDVer and 25.5 points ($[16.2, 34.3]$) on FinLongDocQA test.