FWCOLLAB: BENCHMARKING STRUCTURED MULTI-AGENT COLLABORATION WITH EXECUTABLE DEPENDENCY GRAPHS

Anonymous authors
Paper under double-blind review

ABSTRACT

Terminal success does not describe how a team establishes, maintains, and recovers the intermediate conditions required by its task. We introduce FWCOLLAB, a symbolic two-agent benchmark that operationalizes these cross-role dependencies as private, benchmark-authored executable graphs bound to authoritative environment state. A deterministic, replay-authenticated evaluator measures progress, handoffs, condition regressions, and structural blockers without a language-model judge. The construction library contains 72 Core tasks; evaluation uses a selected 24-task Core subset plus information, bounded-synchronization, and parallel-join tracks, totaling 52 tasks and 936 episodes across six models. Under the tested budgets, intermediate dependency satisfaction and environment completion can differ substantially; these differences characterize execution rather than establish general collaboration quality. A separate paired communication study is analyzed within its own fixed-budget protocol, with interface diagnostics and sensitivity to how controller releases are counted. Conformance and alternative-execution checks validate implementation against the declared specification, not human construct validity. A repair audit identifies historical credit for already-completed stages; requiring new progress removes clear evidence of either selector's superiority. FWCOLLAB provides a reproducible instrument for examining specified cross-role dependencies, with explicit boundaries between executable measurement, downstream utility, and capability comparison.

1 INTRODUCTION

Consider a team navigating a room in which Fire must remain on a pressure plate so that Water can cross a bridge. Reaching the far side does not end their coordination: Water then latches a lever that opens a door for Fire. The first dependency requires continuing support; the second persists after the supporter leaves. If Fire steps off too early, the bridge closes. The team may recover and eventually finish, but its execution differs from a team that maintained support throughout. Conversely, a team can establish the necessary intermediate conditions yet fail to reach the exits before its budget expires. A terminal success bit records neither distinction.

These are *task-required cross-agent dependencies*, not merely patterns of conversation or simultaneous activity. The plate and bridge are components; their control connection is a mechanism. A cross-agent dependency arises only when that mechanism links one role's support to another role's necessary progress. The same physical mechanism can serve different collaboration structures after assigning roles and routes. Figure 1 shows the component–mechanism–dependency chain in the author's original illustration; Appendix Figure 3 grounds the corresponding measurement idea in an actual model episode. This separation matters because a message count, a final-state check, and a dependency violation answer different questions even when they describe the same rollout.

Existing evaluation tools illuminate parts of this problem. Cooperative benchmarks measure team success, efficiency, and communication behavior; graph-based evaluators such as CRAB already gate intermediate checkpoints on predecessor completion (Carroll et al., 2019; Sun et al., 2025; Xu et al., 2025). Our aim is not to introduce DAG evaluation in general. It is to make the *cross-role meaning* of prerequisites explicit: who supplies a condition, who needs it, whether it must remain

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.

active, and which state transition demonstrates its use. Without these bindings, milestone completion cannot distinguish a persistent enabling action from a maintained obligation or explain which role-conditioned prerequisite remains unmet. Free-text explanations introduce a different measurement problem when their interpretation depends on a language-model judge.

FWCOLLAB operationalizes these distinctions in a symbolic two-agent environment. Each task pairs an executable map with a private, benchmark-authored dependency graph. Nodes bind to authoritative state or event predicates; adjacency, joins, and role ownership define when they count. Agents receive the environment rules and their permitted observations, never the graph or its witness. After execution, deterministic replay authenticates the trajectory before the evaluator measures progress, handoffs, controller regressions, and structural blockers. Because predicates refer to states and crossings rather than a prescribed action sequence, alternative valid routes can satisfy the same specification. The symbolic setting trades visual perception and motor complexity for inspectable semantics; it does not claim to represent all forms of real-world collaboration.

A concrete benefit is traceability from a reported metric back to an observed event. In the model case shown later, both branches and the join complete, a post-join support condition is released and restored, and the agents eventually reach their exits. Rendered frames expose these transitions beside their dependency states (Appendix F). They also expose the limits of the labels: a local preparation rollback is not necessarily interruption of already-established joint support, and an evaluator's first incomplete node is not a human-validated causal diagnosis. We therefore separate the measurement rule, its implementation checks, and evidence about downstream usefulness rather than treating them as interchangeable validation.

The construction library contains 72 Core instances and 367 audited ENABLE/MAINTAIN dependencies. The six-model evaluation covers 24 selected Core tasks plus information, bounded-synchronization, and parallel-join tracks: 52 tasks and 936 episodes. The other 48 Core tasks support construction and evaluator checks, not the six-model result table. Under the tested budgets, Qwen3.5-27B's Information dependency success is 55.6% while environment success is 2.8%; its Join environment success is 95.8% while violation-free execution is 62.5%. These differences describe execution, not a universal quality ranking. Information's tight budget, condition-dependent fallback errors in a separate communication study, and repair-credit sensitivity all constrain stronger conclusions.

Contributions.

- A role-grounded task specification and formal contraction from native evaluator graphs to executable cross-agent dependencies (Sections 3–4).
- A deterministic, replay-authenticated measurement instrument, validated against declared semantics and illustrated through rendered real trajectories (Section 5; Appendix F).
- Multi-track model results and explicit interface, maintenance, and repair-criterion audits that delimit what outcome–process differences establish (Section 7).

2 RELATED WORK

Interactive multi-agent benchmarks. Overcooked-AI established a two-player cooperative testbed for human-aware coordination with held-out proxy-human and human partners (Carroll et al., 2019); the Overcooked Generalisation Challenge evaluates unfamiliar partners and layouts through self-play, cross-play, and ad-hoc teamwork (Ruhdorfer et al., 2025). Collab-Overcooked enforces resource and knowledge partitioning and measures trajectory efficiency and initiation/response behavior (Sun et al., 2025). MultiAgentBench covers collaborative and competitive scenarios with milestone, communication, and planning scores; its agent graph describes communication relations rather than task dependencies (Zhu et al., 2025). TeamCraft studies multimodal embodied teams with temporal and resource dependencies (Long et al., 2024). These benchmarks offer richer perception, embodiment, or partner populations than FWCOLLAB; their primary protocols score collaboration through outcomes, milestones, or behavior statistics rather than through a benchmark-authored graph of cross-agent dependencies executed against the trajectory.

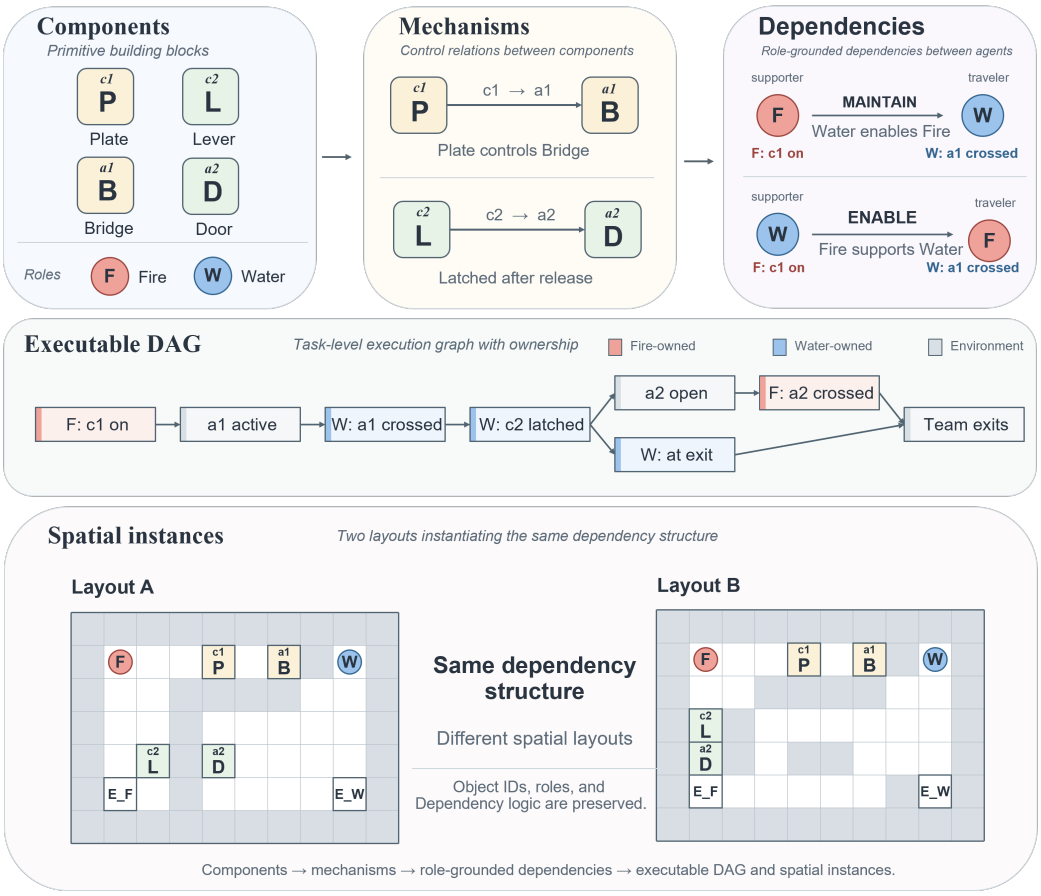


Figure 1: The author's original illustration, reproduced without redrawing: components, control mechanisms, role-grounded dependencies, an executable DAG, and spatial layouts. Interpret role directions using the F/W arrows: F holds c1 for W to cross a1 (MAINTAIN); W latches c2 for F to cross a2 (ENABLE). The dependency-panel explanatory text contains transposed role labels. The layouts are conceptual, not evaluated or certified no-bypass instances.

Structured evaluation and failure diagnosis. CRAB already uses a benchmark-owned evaluator graph with state checks, predecessor-gated checkpoints, and intermediate completion scoring for single-agent, cross-environment computer tasks (Xu et al., 2025). These generic DAG-evaluation ideas are not our novelty: the contribution here is role-grounded dependency semantics, volatile support and recovery, and their executable verification in two-agent tasks. VillagerAgent generates a method-side DAG to schedule multi-agent work (Dong et al., 2024). CoCoBench deterministically categorizes dependency, conflict, affordance, and skill-call failures from execution traces (Chen et al., 2026b), and TraceElephant evaluates attribution of failures to a responsible component and decisive step (Chen et al., 2026a).

Positioning. Each component above appears elsewhere. FWCOLLAB contributes their conjunction for two-agent collaboration: benchmark-authored executable cross-agent dependencies, authoritative state bindings, deterministic process evaluation over replay-authenticated traces, dependency-aware failure localization, and controlled collaboration semantics (ENABLE, MAINTAIN, information, bounded synchronization, parallel join) that can be varied independently. Appendix Table 23 provides the source-qualified comparison; it is not a priority claim.

3 BENCHMARK DESIGN

3.1 TASKS AND ROLLOUT

FWCOLLAB asks whether two agents satisfy cross-agent dependencies: one agent's progress depends on the other establishing, maintaining, communicating, or synchronizing a condition. A task instance is

$$\mathcal{B} = (\mathcal{M}, \mathcal{G}, \mathcal{R}, H), \tag{1}$$

where $\mathcal{M}$ is a symbolic map, $\mathcal{G}$ a private benchmark-authored executable dependency DAG, $\mathcal{R}$ the public transition system, and H the round budget. Roles F and W must both reach role-specific exits; an episode otherwise ends on a lethal state or timeout. Maps are grids with controllers (plates, levers, toggles), actuators (doors, bridges), hazards, movable objects, light paths, and exits. Actions are one-cell moves or `WAIT`. This abstraction removes motor-control confounds while retaining ordering, persistent and maintained state, role-conditioned coordination, and recovery.

At round t, role $i \in \{F, W\}$ receives an observation $o_t^i = O_i(s_t)$ of the authoritative world state s_t and decides concurrently from its own private history. In Core, Sync-v2, and Join, $O_F = O_W$ is the full public state (Sync-v2 adds the public charge and latch fields); in Information, O_i is a role-private projection that hides the partner's fact. The environment validates both intents against the pre-transition state, applies legal moves simultaneously, updates mechanisms, and writes s_{t+1} to the authoritative trace. A public message sent at t arrives at $t + 1$. Agents never see the DAG, node bindings, generation witness, or evaluator output; private reasoning is neither forwarded nor scored.

3.2 CONSTRUCTION AND ACCEPTANCE

Tasks follow the conceptual chain

$$\begin{aligned}\text{components} &\to \text{mechanisms} \to \text{role-grounded dependencies} \\ &\to \text{executable DAG + spatial scene} \to \text{deterministic evaluation.}\end{aligned} \tag{2}$$

Nineteen implemented components (e.g., a pressure plate or door) form 30 normalized mechanisms M01–M30 (e.g., a role-held plate controlling a door). A mechanism becomes a cross-agent dependency only after role grounding: a supporter establishes a controller or capability state, a different traveler must cross the resulting cut, and the manifest records evidence that the cut cannot be bypassed. Map and DAG are generated jointly from one stage specification. An instance is accepted only if a deterministic witness reaches team success, every node has an executable binding, and every edge has construction evidence (Appendix C.1). The witness is a solvability certificate, not a reference trajectory; evaluated policies need not reproduce it.

3.3 TASK SUITES

Core (Full-72). Seventy-two instances over difficulty levels L1–L7 collectively cover M01–M30. Canonicalization separates 72 typed executable templates, 65 normalized collaboration templates, and 48 pure dependency topologies (Table 1). Their contracted cross-agent dependencies (Section 4.2) are ENABLE or MAINTAIN. Diagnostic-24 is a structurally selected subset (21 topologies, 23 templates) chosen without model outcomes; a structural audit finds no consistent evidence that it is harder than Full-72 (all standardized mean differences $|d| < 0.4$; Appendix C.3).

Extension tracks. *Information-12* gives the roles complementary private facts (controller wiring versus required route) that only delayed messaging can join. *Sync-v2-8* requires both role-owned plates to be active for $K \in \{1, 2, 3\}$ consecutive rounds; the public observation exposes K, the current charge, and a latch that alone opens the synchronization gate, and exhaustive state-space search verifies that team success implies synchronization. *Join-8* requires independent F- and W-owned branches to complete before an `all`-join, in either order; in three tasks a role-held MAINTAIN stage follows the join. Each track passes witness, replay, and necessity checks (Appendix C.5).

Standard-52. The model-evaluation set combines 24 selected Core tasks (horizon 200), Information-12 (30), Sync-v2-8 (80), and Join-8 (80). All have replay-verified success certificates: the shortest available certificates take at most 164/27/36/34 rounds, respectively. These are

Table 1: Benchmark suites. For Full-72, "typed" preserves executable node/edge semantics up to a role swap, "normalized" removes layout and mechanism identity but keeps the collaboration-stage program, and "topology" keeps only directed adjacency and joins. Dependencies are contracted cross-agent dependencies (Section 4.2).

Suite	Tasks	Typed	Normalized	Topologies	Dependency semantics
Full-72	72	72	65	48	216 ENABLE, 151 MAINTAIN
Diagnostic-24	24	24	23	21	ENABLE, MAINTAIN
Information-12	12	–	–	–	private-information dependency
Sync-v2-8	8	–	–	–	bounded synchronization, $K \in \{1, 2, 3\}$
Join-8	8	–	–	–	role-owned parallel join
Standard-52	52	Diagnostic-24 (200 rounds) + Info-12 (30) + Sync-v2-8 (80) + Join-8 (80)			

upper bounds on required solution length, not optimal lower bounds or evidence that extra time would not help a policy. Information certificates take 23 rounds on one task and 27 on eleven, leaving only 7 or 3 rounds relative to these witnesses. Its reported outcome–dependency gap is therefore budget-conditioned. Full-72 denotes the construction library, not six-model coverage.

4 EXECUTABLE DEPENDENCY DAG

For each instance, $\mathcal{G} = (V, E)$ is an acyclic measurement decomposition. Each node has a kind (condition, state, event, goal), an owner (F, W, team, or environment), a join $j_v \in \{\texttt{all}, \texttt{any}\}$, and a predicate ϕ_v that manifest bindings compile into equality or role-boundary tests over authoritative state. An edge (u, v) makes u a predecessor of v.

4.1 STATE-GROUNDED COMPLETION

At snapshot s_t, nodes are scanned in a fixed topological order. Let $C_{t,<v}$ contain prior-snapshot completions and those added before visiting v in this scan (initially empty at $t = 0$). With predecessors $P(v)$,

$$\mathrm{ready}_t(v) = \begin{cases} 1, & P(v) = \varnothing, \\ \mathbb{1}[P(v) \subseteq C_{t,<v}], & j_v = \texttt{all}, \\ \mathbb{1}[P(v) \cap C_{t,<v} \neq \varnothing], & j_v = \texttt{any}. \end{cases} \tag{3}$$

A true, ready node completes immediately, so chains can complete within one snapshot; C_t is the set after the scan. Completion is monotone, while predicate regressions are recorded separately. Traversal predicates test a living responsible role beyond an actuator boundary, not a witness coordinate. When an `any` target completes, unfinished direct alternative predecessors are waived from that round onward, not recursively upstream. For non-start nodes V^+ and waivers effective by t, W_t, progress is

$$V_t^{\mathrm{req}} = V^+ \setminus W_t, \qquad p_t = \frac{|C_t \cap V_t^{\mathrm{req}}|}{|V_t^{\mathrm{req}}|}, \tag{4}$$

with $p_t = 1$ for an empty denominator. AUC integrates these time-local values, not values retrospectively normalized by the final required set. A waiver can itself increase progress; p_T is DAG Completion.

4.2 FROM NATIVE GRAPH TO CROSS-AGENT DEPENDENCIES

The native DAG contains agent-owned controller nodes, environment actuator nodes, and agent-owned progress nodes. Its edge labels (in Full-72: 500 `requires`, 822 `enables`, 72 `synchronizes`) are construction provenance; readiness depends only on adjacency and joins. The paper's dependency semantics are defined by a separate *contraction*.

Definition. A native path family $C \to e \to t$, where C is a set of controller nodes owned by supporter A, e an environment actuator, and t a progress node owned by traveler B, contracts to one semantic edge $(A, \phi_C) \xrightarrow{\tau} (B, \phi_t)$ iff (i) $A \neq B$; (ii) all path nodes are bound and each $c \to e$ link has Boolean-control evidence (necessity for `all`, sufficiency for `any`); and (iii) $e \to t$ is a verified

required cut (single-lane corridor or mandatory capsule). The label is $\tau = \textsc{Maintain}$ if ϕ_C contains a volatile role-held plate, so the source must stay true through the crossing, and $\tau = \textsc{Enable}$ if the established state persists without the supporter. A multi-controller stage yields one edge because its source is one Boolean expression.

Worked examples. In task V4-L1-003, F stands on plate `c1`, which raises platform `a1`, which W crosses: `n001` $\rightarrow$ `n002` $\rightarrow$ `n003` contracts to F:`c1_on` $\xrightarrow{\textsc{Maintain}}$ W:`a1_crossed`, because the plate is recomputed from occupancy and releasing it before W crosses closes the platform. In V4-L2-002, F latches lever `c2`, opening door `a2`; the lever persists after F leaves, so the path contracts to F:`c2_on` $\xrightarrow{\textsc{Enable}}$ W:`a2_crossed`.

Counts. Applying the contraction to Full-72 (Appendix C.2, Algorithm 1) yields 367 cross-agent dependencies: 322 from controller stages and 45 from collaborative capability capsules; 216 are ENABLE and 151 MAINTAIN. Handoff, mutual enabling, and role alternation are compositions of successive contracted edges; the native `synchronizes` edges only connect finished paths to the goal and impose no temporal window, so bounded synchronization is instantiated only in Sync-v2-8.

5 PROCESS-AWARE EVALUATION

Given a manifest record, private DAG, and trace, the evaluator first replays the action sequence in a fresh world and requires the reconstructed observations to match the stored ones; it then executes node bindings on every reconstructed snapshot. State-grounded process metrics are computed from these replayed states and actions. The Information track additionally checks recorded fact-transmission and receipt events: whether the sender's structured message and the partner's delivered inbox contain the required actuator identifier, a deterministic string match. No metric uses a language-model judge, scores model-private reasoning, or grades free-text message quality.

Progress. DAG Completion is p_T. For $T > 0$ executed rounds, Progress AUC is $\frac{1}{T}\sum_{t=1}^{T}(p_{t-1} + p_t)/2$, using the time-local denominator in Equation 4. Horizons, task lengths, and decompositions differ across tracks; AUC values are compared within a track and protocol, not as a cross-track quality scale.

Controller regressions and handoffs. A true-to-false `occupy`/`activate` predicate with a direct dependent still pending is a *pending-dependent regression* (legacy field: `harmful`). The separate stage-level `premature_release` rule records a controller release before crossing when the post-transition actuator is off. It can fire before the full Boolean support was ever established: partial preparation rollback is not necessarily interruption of partner support. Section 7.3 separates these cases without rewriting the frozen scores. A handoff succeeds when its crossing node completes and is *clean* if no recorded coordination violation belongs to that stage; this is a rule-defined indicator, not a validated general quality judgment. No opportunities means N/A.

Violation ledgers. The dependency ledger records premature predicate satisfaction and closed-actuator crossing attempts. The coordination ledger records closed-actuator attempts, stage controller release, and capability regression before crossing; it is not a superset of the dependency ledger. Premature predicate satisfaction is absent from coordination counts. Rates use executed rounds (Appendix A.6).

Failure stage. For an unsuccessful trajectory, the evaluator returns the first topologically ordered incomplete required node, its incomplete direct predecessors, responsible role, associated violations, and last progress round. This is structural localization under the frozen graph, not causal attribution (Appendix Figure 2).

Track-specific process indicators. Each extension track reports one indicator beside environment success (Env. SR, both agents at their exits). *Clean info. handoff* (Information) is an intermediate event: the fact is sent and delivered, the correct controller is activated no earlier than receipt, and no wrong controller is activated first; reaching the exits is not required. *Sync-v2 task success* requires

team success, the public synchronization latch, and full DAG completion; in the current suite it coincides with Env. SR, so one column is reported. *Violation-free* (Join) requires both branch nodes and the join node to complete with no dependency violation anywhere in the trace, including early release of the held condition in the MAINTAIN stage after the join; it does not itself require team success or completion of other downstream nodes. Field mappings: Appendix A.4.

6 EXPERIMENTAL SETUP

Two separate studies. Standard-52 measures four tracks under their feasible budgets (Section 3); a paired communication study uses Diagnostic-24 with a fixed 80-round budget. The latter is a protocol-specific intervention on communication, not a budget intervention or a replication of communication effects on Standard-52. Appendix A.9 compares the effective prompts, history windows, reasoning, output constraints, and model routes. In particular, GPT-5.5's differing Core results cannot be attributed to the budget alone.

Standard-52. Six homogeneous teams contribute 156 episodes each (52 tasks, three replicates; 936 total): GPT-5.5 (reasoning `none`), GPT-5-mini/GPT-5.4-mini (`low`), and Qwen3.5-9B/27B/122B-A10B-FP8. Roles have independent contexts, one-round message delay, no planning rounds, temperature 1.0, and an 800-output-token cap. Model-format failures receive one retry, then a scored `WAIT`. Qwen uses a shared vLLM JSON-schema route with `enable_thinking=false`; no model-specific output repair is applied. Reasoning and decoding conditions are not compute-matched. Provenance, including the 122B precision-label caveat and excluded Gemini routes, is in Appendices A.3 and E.

Paired communication study. GPT-5.5 and Gemini-3.7-Flash are evaluated in homogeneous self-play, matched No-Comm controls, and both heterogeneous role directions: $24 \times 6 \times 3 = 432$ episodes. In No-Comm, agents are explicitly told that messaging is disabled and must emit `message=null`; inboxes are empty and outgoing messages suppressed. Shared public state remains visible. Thus the treatment changes both channel availability and its disclosed instructions, not silent message loss. At 80 rounds, 9/24 tasks are provably infeasible; the ten certified-feasible tasks provide a separate sensitivity population. Appendix A.8 reports decision reliability rather than inferring it from episode validity.

Metrics and inference. We report SR, DAG completion, progress AUC, rule-defined clean handoff, and coordination violations per 100 executed rounds. Three repeats are averaged within task; aligned tasks are bootstrapped 10,000 times. Paired communication effects are $\Delta_m = \text{Self}_m - \text{NoComm}_m$, with direct interaction $\Delta_{\text{GPT}} - \Delta_{\text{Gemini}}$. For violations, $-\Delta_m$ is benefit-oriented. Endpoints are exploratory. New audits distinguish pooled decision rates from task-first performance effects; unavailable retry-attempt detail is not imputed as zero.

7 RESULTS

7.1 OUTCOME AND DEPENDENCY MEASURES ON STANDARD-52

Table 2 reports all 936 admitted episodes. Environment completion and intermediate dependency indicators answer different questions; neither alone measures general collaboration quality. Source gates and offline aggregation checks are distinguished in Appendix A.

Core. Despite certified feasibility within 200 rounds, GPT-5.5/GPT-5-mini/GPT-5.4-mini achieve 29.2/13.9/12.5% environment success and 58.9/41.0/31.3% DAG completion. Process measures for all six models are in Appendix Table 6.

Information. Dependency success exceeds environment success for each GPT model. In 20/108 GPT episodes, clean information transfer occurs without both agents reaching their exits within 30 rounds. This budget-conditioned observation does not establish that the gap persists at 45 or 60 rounds; the lack of reverse cases is not independent evidence of quality.

Table 2: Standard-52: six models in homogeneous self-play, three replicates per task. Closed models above the rule; open-weight models below. Task-first percentages; definitions in Appendix A.4, 95% CIs in Table 3. No overall score.

	Core-24	Information-12		Sync-v2-8	Join-8	
Model	Env. SR	Env. SR	Dependency SR	Sync task SR	Env. SR	Violation-free SR
GPT-5.5	29.2	8.3	33.3	100.0	100.0	100.0
GPT-5-mini	13.9	0.0	5.6	95.8	95.8	62.5
GPT-5.4-mini	12.5	5.6	30.6	100.0	100.0	66.7
Qwen3.5-9B	4.2	0.0	0.0	70.8	62.5	66.7
Qwen3.5-27B	6.9	2.8	55.6	87.5	95.8	62.5
Qwen3.5-122B-A10B-FP8	1.4	2.8	30.6	87.5	87.5	66.7

Synchronization. GPT teams complete 71/72 episodes; Qwen rates range from 70.8% to 87.5%. The present track offers limited discrimination near this ceiling despite its validated synchronization requirement.

Join. GPT environment success is 95.8–100%, but violation-free rates fall to 62.5% and 66.7% for the smaller GPT models. All 16 GPT success/violation discrepancies arise from one post-join premature release followed by recovery, confined to three maintenance-bearing tasks (Appendix A.4). This is evidence of a specific recoverable execution pattern, not broad join failure; the cause attribution does not extend to Qwen.

Open-weight results. Qwen3.5-9B/27B/122B-A10B-FP8 achieve Core success of 4.2/6.9/1.4%. For 27B, Information dependency success is 20/36 (55.6%) versus environment success 1/36 (2.8%); Join environment success is 23/24 (95.8%) versus violation-free satisfaction 15/24 (62.5%). Across all Standard-52 decisions, only 7/191,966 become fallback `WAIT`; all seven are closed-model decisions. However, 33,419/101,526 Qwen decision opportunities are parsed non-`WAIT` actions blocked by the environment. Low success is not explained by widespread terminal parse failures, but this does not isolate coordination from navigation or planning limits. Results remain descriptive, not a scaling law.

7.2 Communication effects under a fixed protocol

This fixed-80-round study has no matched Standard-52 No-Comm replication. Episode replay validity does not imply decision reliability: Gemini fallback decisions are 767/10,708 (7.163%) with communication versus 1090/10,834 (10.061%) without; GPT's are 241/10,344 (2.330%) versus 192/10,286 (1.867%). Thus the observed interaction combines coordination and condition-dependent output reliability; the latter competing explanation is *not excluded*. The frozen contrasts below remain total effects under this runtime, not isolated communication or family-capability effects (Appendix A.8).

Full condition estimates and marginal intervals are retained in Appendix Table 18.

Similar outcomes, different processes. GPT Self-play and No-Comm have near-equal SR (38.89%/40.28%; paired effect -1.39 pp), but messaging accompanies higher clean handoff (+4.20 pp) and fewer violations (+0.518/100 rounds). These are total runtime effects. Complete paired intervals, event counts, and cross-play results are retained in Appendix B.1.

Model×Communication. Under this runtime, Gemini's completion, AUC, and clean-handoff effects are positive; its SR effect is inconclusive. The direct interaction on 24 tasks excludes zero for SR, completion, AUC, and violations, not clean handoff (Appendix Table 19). This establishes a protocol-specific contrast, not an isolated coordination mechanism or a general model-family ordering.

7.3 Measurement boundaries and diagnostic scope

Full support versus partial preparation. Replay reproduces all 432 original episodes: 58 releases interrupt established support, while five roll back preparation. Excluding the latter does not change the communication-clean or P2 contrasts; the P0 violation interaction remains negative (Appendix A.7). The event meanings are still distinct.

Horizon feasibility. At 80 rounds, tasks split into ten certified-feasible, nine infeasible, and five unresolved. Removing the nine all-zero SR tasks only rescales SR contrasts by $24/15$, not independent robustness evidence. In the certified subset, several process intervals include zero and Gemini's violation point estimate changes sign (-0.035 to $+0.042$/100 rounds). Appendix Table 19 gives the direct interactions; we make no blanket sign-preservation claim.

Graph decomposition. Coarse and refined decompositions preserve condition ordering and effect signs on all 432 traces ($\rho \geq 0.975$ completion; 285/285 stage-localization agreement). This does not establish a unique or externally validated graph.

Evaluator conformance. Full-72 passes 555/555 applicable cases: all 267 invalid mutations detected and no false positives on 216 valid perturbations. Forty scripted alternative trajectories also match expected output; eight use an earlier synchronization prototype, not the current track. Appendix D separates this coverage from current synchronization's exhaustive validation. These tests establish specification fidelity, not human construct validity.

Diagnostic criteria and selectors. The old repair scores favor local-event, but incorrectly credit already-completed stages without new progress. Removing that credit leaves 138/285 useful local-event targets versus 185/285 Full-DAG targets; the task-first difference is inconclusive (-4.92 pp $[-25.88, +14.27]$). Appendix B.4 preserves all old/new counts and selection confounds; Figure 8 visualizes them. Neither this retrospective criterion check nor coarse collision groups establish an isolated edge gain, external blocker correctness, or general repair superiority.

8 Discussion and Limitations

Measurement is not a universal quality scale. The graphs operationalize specified task dependencies, not a unique decomposition or a human-validated notion of collaboration. Pending-dependent regressions can include partial preparation rollback; the narrower full-support sensitivity analysis must be read beside, not substituted for, the frozen metrics. Structural blockers localize an unmet specification, not a causal root cause. Human agreement has not been measured.

Utility and coverage. Repair conclusions depend on representation, selection, executor coverage, and the progress-credit rule; no isolated edge gain is established. Collision groups establish only the insufficiency of four coarse statistics. Six-model evaluation covers 24/72 Core tasks, with floor/ceiling effects across tracks. Process indicators describe additional events but do not establish sensitivity to method improvements. Navigation and planning remain possible failure sources even when terminal interface failures are rare.

Budgets and protocols. Solvability certificates do not make budgets nonbinding. Information needs an unperformed 45/60-round rerun before a budget-robust claim. The paired study has 9/24 infeasible tasks, differing runtime settings, and condition-dependent fallbacks; its communication effects are protocol-specific. Three Qwen variants do not establish family diversity, matched compute, or scaling. Results do not cover continuous control or larger teams.

9 Conclusion

FWCollab provides an executable, replay-verifiable specification of cross-role dependencies. Its evidence describes execution beyond terminal success under explicit budget and protocol limits; broader construct validity and superior diagnostic utility remain unestablished.

AI USE STATEMENT

Generative-AI tools, including Claude Code, assisted with manuscript drafting, language polishing and revision; literature search and reference discovery; research ideation and implementation; code navigation and analysis scripts; figure preparation; and consistency checks. AI assistance also supported synthetic benchmark task and map construction. These research-assistance uses are distinct from using language models as the evaluated agents: reported performance values are derived from recorded experimental trajectories and deterministic analyses, not generated estimates.

The benchmark's process scores are computed by a deterministic evaluator rather than an LLM judge. Automated replay, source-hash checks, and software tests are used to check the resulting artifacts; these checks do not establish human construct validity or replace author responsibility. The authors remain responsible for the research design, accuracy, originality, attribution, interpretation, and all AI-assisted content in the submission. Limitations and unavailable validation are disclosed in the paper rather than treated as completed work.

REFERENCES

Micah Carroll, Rohin Shah, Mark K. Ho, Thomas L. Griffiths, Sanjit A. Seshia, Pieter Abbeel, and Anca Dragan. On the utility of learning about humans for human-ai coordination. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/hash/f5b1b89d98b7286673128a5fb112cb9a-Abstract.html.

Mengzhuo Chen, Junjie Wang, Fangwen Mu, Yawen Wang, Zhe Liu, Huanxiang Feng, and Qing Wang. Seeing the whole elephant: A benchmark for failure attribution in LLM-based multi-agent systems. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 19888–19905, San Diego, California, United States, July 2026a. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.912. URL https://aclanthology.org/2026.acl-long.912/.

Yang Chen, Ye-Xin Xie, Lirong Che, Danyang Peng, Yuzhe Yang, Peiwen Lin, Xu Cao, Chuang Wang, Lei Yuan, Jian Su, and Lan-Zhe Guo. CoCoBench: A cooperative coordination benchmark for embodied multi-agent task planning, August 2026b. URL https://arxiv.org/abs/2608.28266.

Yubo Dong, Xukun Zhu, Zhengzhe Pan, Linchao Zhu, and Yi Yang. VillagerAgent: A graph-based multi-agent framework for coordinating complex task dependencies in Minecraft. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 16290–16314, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.964. URL https://aclanthology.org/2024.findings-acl.964/.

Qian Long, Zhi Li, Ran Gong, Ying Nian Wu, Demetri Terzopoulos, and Xiaofeng Gao. TeamCraft: A benchmark for multi-modal multi-agent systems in Minecraft, December 2024. URL https://arxiv.org/abs/2412.05255. arXiv:2412.05255v1.

Constantin Ruhdorfer, Matteo Bortoletto, Anna Penzkofer, and Andreas Bulling. The overcooked generalisation challenge: Evaluating cooperation with novel partners in unknown environments using unsupervised environment design. *Transactions on Machine Learning Research*, September 2025. URL https://openreview.net/forum?id=K2KtcMlW6j.

Haochen Sun, Shuwen Zhang, Lujie Niu, Lei Ren, Hao Xu, Hao Fu, Fangkun Zhao, Caixia Yuan, and Xiaojie Wang. Collab-overcooked: Benchmarking and evaluating large language models as collaborative agents. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 4922–4951, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.249. URL https://aclanthology.org/2025.emnlp-main.249/.

Tianqi Xu, Linyao Chen, Dai-Jie Wu, Yanjun Chen, Zecheng Zhang, Xiang Yao, Zhiqiang Xie, Yongchao Chen, Shilong Liu, Bochen Qian, Anjie Yang, Zhaoxuan Jin, Jianbo Deng, Philip Torr, Bernard Ghanem, and Guohao Li. CRAB: Cross-environment agent benchmark for multimodal language model agents. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 21607–21647, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-256-5. doi: 10.18653/v1/2025.findings-acl.1113. URL https://aclanthology.org/2025.findings-acl.1113/.

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 Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8580–8622, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.421. URL https://aclanthology.org/2025.acl-long.421/.

A Reproducibility Details

A.1 Frozen Assets and Privacy Boundary

Two kinds of freeze are used. Task and protocol freezes are written before data collection, and the runner verifies every listed SHA-256 before model calls. Result freezes are written after collection and seal the admitted traces and derived tables. The Phase II snapshot covers Diagnostic-24, its DAGs/maps, evaluator/runtime dependencies, schemas, protocol, and runner; a separate Benchmark v1 manifest covers all 72 map/DAG pairs. The corrected closed-model results are sealed by `eval_private/leaderboard_v2/closed3_result_freeze.json` (SHA-256 `7abff68c9b9f9a00...aa3aae`), which records the model matrix, Standard-52 run manifest, unified evaluator, Sync-v2 world, per-episode results, and integrity report. Existing freezes are never edited; a revised suite or protocol (e.g., Sync-v2, the corrected Standard-52) receives a new freeze, and the superseded one remains immutable. Acting policies receive only the public rules and agent observation; DAGs, bindings, witnesses, and evaluator output are excluded from prompts.

A.2 Replay Authentication and Publication Gates

The source analyzers check key uniqueness, trace parseability, condition metadata, requested and resolved model provenance, role assignment, communication setting, seed, map identity, replicate completeness, and absence of episode-level `runner_error`. During source finalization, each action sequence is replayed in a fresh world; stored observations must match the reconstruction, which is then passed to the frozen evaluator. Phase II admitted 432/432 episodes (Table 5). Corrected Closed-3 admitted 468/468 entries (264 re-validated under identical Core/Information/Join protocols and 204 new), with 468 replay passes, 468 evaluator recomputations, and zero mismatches.

Each of the three open-weight return packages reports 156/156 valid episodes, replay passes, evaluator recomputations, and schema passes, with zero runtime/evaluator mismatches. Before merging, all 516 files listed across the three package manifests were checked for size and SHA-256, all 468 trajectory hashes were matched against episode and gate records, and the task/replicate sets were matched to Standard-52. The two evaluation representations agree after aligning their outer model/track/replicate fields. The offline merge admits 936 unique episodes, 312 model–task rows, and 24 model–track rows. These are source-gate evidence and local integrity checks, not a fresh 936-episode replay or evaluator rerun.

Tables 2, 3, and 6 are generated by `tools/make_standard52_tables.py` from a hash-pinned aggregate snapshot. The manuscript's `evidence/standard52_final6/manifest.json` records the source hashes, the existing Closed-3 freeze, the three archive hashes, and the merger configuration; generated provenance records the input and table hashes. Frozen means and intervals are rendered without changing the evaluator or resampling with the older Closed-3 table generator's seed.

A.3 Open-weight Serving Provenance

The return-package identifiers are Qwen3.5-9B, Qwen3.5-27B, and `Qwen-Qwen3.5-122B-A10B-FP8`; the table drops the redundant first `Qwen-` prefix from the last display name only. All three use vLLM 0.19.1, tensor parallelism 4, the same Chat Completions JSON-schema contract, strict structured output, `enable_thinking=false`, temperature 1.0, and an 800-token response cap. The prompt contract, Standard-52 task manifest, evaluator, and effective action-schema hashes coincide across the three packages. No model-specific output repair is applied. All three per-replicate server-identity gate records match for each model. Closed-model reasoning settings and open-weight constrained decoding are not compute-matched treatments.

The 122B identifier contains an `FP8` suffix, whereas its run metadata reports `dtype=bfloat16` and `quantization=none`. We retain the recorded model identifier but do not infer the actual weight-quantization configuration from the filename or draw precision-related conclusions. The three archive SHA-256 identifiers, in 9B/27B/122B order, are `5858893ad2c0503b...99ccc05`, `62217ebff638cbc9...6a17162`, and `af5e2d07a16658e8...194ac0a0`; full digests are in the manuscript evidence manifest.

Table 3: Standard-52 outcomes with frozen pointwise 95% task-bootstrap CIs (10,000 draws, seed 20260922; three replicates averaged within task). All entries are percentages. Definitions are in Appendix A.4.

(a) Core and Information

Model	Core Env. SR	Info Env. SR	Info Dependency SR
GPT-5.5	29.2 [15.3, 44.4]	8.3 [0.0, 25.0]	33.3 [8.3, 58.3]
GPT-5-mini	13.9 [2.8, 26.4]	0.0 [0.0, 0.0]	5.6 [0.0, 13.9]
GPT-5.4-mini	12.5 [2.8, 25.0]	5.6 [0.0, 16.7]	30.6 [8.3, 55.6]
Qwen3.5-9B	4.2 [0.0, 12.5]	0.0 [0.0, 0.0]	0.0 [0.0, 0.0]
Qwen3.5-27B	6.9 [0.0, 15.3]	2.8 [0.0, 8.3]	55.6 [36.1, 72.2]
Qwen3.5-122B-A10B-FP8	1.4 [0.0, 4.2]	2.8 [0.0, 8.3]	30.6 [8.3, 52.8]

(b) Synchronization and Join

Model	Sync-v2 task SR	Join Env. SR	Join Violation-free SR
GPT-5.5	100.0 [100.0, 100.0]	100.0 [100.0, 100.0]	100.0 [100.0, 100.0]
GPT-5-mini	95.8 [87.5, 100.0]	95.8 [87.5, 100.0]	62.5 [25.0, 87.5]
GPT-5.4-mini	100.0 [100.0, 100.0]	100.0 [100.0, 100.0]	66.7 [37.5, 91.7]
Qwen3.5-9B	70.8 [58.3, 83.3]	62.5 [37.5, 87.5]	66.7 [37.5, 100.0]
Qwen3.5-27B	87.5 [62.5, 100.0]	95.8 [87.5, 100.0]	62.5 [25.0, 100.0]
Qwen3.5-122B-A10B-FP8	87.5 [62.5, 100.0]	87.5 [70.8, 100.0]	66.7 [33.3, 91.7]

A.4 Track Metric Definitions

All track indicators originate from the frozen evaluator applied to replayed traces (states, actions, and, for Information, recorded messages and inboxes). The earlier paper-side audit reproduced every Closed-3 value; the open-weight packages supply their own replay/recomputation gates as described above. *Env. SR* is `environment_team_success`: both agents at their exits within the horizon; Core semantic success is identical. *Information dependency SR* is `information_dependency_success`: a clean information handoff with no recorded dependency violation. W's structured message must contain the required actuator identifier, F's delivered inbox must contain it, and the correct controller must be activated no earlier than receipt, with no wrong controller activated first. It does not require reaching the exits (`c5.evaluate_c5_trace` and the unified evaluator). *Sync-v2 task SR* requires team success, the public synchronization latch, and full DAG completion; the suite invariant makes it coincide with Env. SR, so one column is shown. *Join violation-free SR*, exported as `join_satisfaction`, requires both role-owned branch nodes and the join node to complete and no dependency violation anywhere in the trace. It does not

Table 4: Frozen Phase II conditions. In Phase II (Diagnostic-24), both roles receive the same public environment state.

Condition	F policy	W policy	Messages
GPT self-play	GPT-5.5	GPT-5.5	on
Gemini self-play	Gemini-3.7-Flash	Gemini-3.7-Flash	on
GPT no-comm	GPT-5.5	GPT-5.5	off
Gemini no-comm	Gemini-3.7-Flash	Gemini-3.7-Flash	off
Cross-play A	GPT-5.5	Gemini-3.7-Flash	on
Cross-play B	Gemini-3.7-Flash	GPT-5.5	on

Table 5: Final Phase II publication-gate accounting. "Analyzer calls" counts provider calls made by analysis, not acting-model calls during data collection.

Check	Final value
Publication status	Final; gate passed
Episodes	432/432
Complete paired tasks	24
Valid traces	432/432
Missing/unexpected/invalid traces	0/0/0
Outcomes	147 success; 285 timeout
Episode-level `runner_error`	0
Observations authenticated by replay	Yes
Frozen files verified	61
Analyzer provider calls	0
Analyzer input mutation	No

itself require team success or completion of other downstream nodes; violations include a predicate rising before its prerequisites and early release of any held (`maintains`) condition, including the MAINTAIN stage following the join in three tasks. In all 16 Closed-3 episodes with Env. SR $= 1$ and violation-free SR $= 0$, both branches and the join completed and the only violation was one early release of that post-join condition. This cause attribution is restricted to the audited Closed-3 subset; the Qwen table reports the same indicator without extending that trace-level diagnosis.

A.5 BOOTSTRAP PROCEDURE

For episode metric $x_{c,r,i}$ with condition c, replicate r, and task i, analysis uses $\bar{x}_{c,i} = \frac{1}{3}\sum_r x_{c,r,i}$. Standard-52 resamples tasks within each model–track cell for 10,000 draws (seed 20260922), reporting pointwise percentile 95% intervals; it does not treat repeated episodes as independent tasks or test pairwise model differences. Its six-model CIs replace the earlier Closed-3 table's bootstrap realization, while Closed-3 point estimates are unchanged. Degenerate intervals at 0 or 100% reflect the observed task sample, not certainty about a broader population. Phase II instead resamples aligned task identifiers to aggregate task-paired differences, also using 10,000 draws and the 2.5/97.5 percentiles. Missing handoff opportunities remain missing. Rounds-to-Success uses the within-task median over successful replicates and resamples contributing tasks; SR and the task count accompany it.

A.6 VIOLATION UNITS AND TAXONOMY

The headline rate is $k/\max(T, 1)$ per executed round (reported per 100 rounds); the native per-agent-action rate uses $\max(2T, 1)$. The dependency ledger contains premature predicate satisfaction and closed-actuator attempts. The separate coordination ledger contains closed-actuator attempts, stage controller releases, and capability regressions; it does not include every dependency event. The generic node-level `harmful` flag only means a direct dependent remains pending. It is distinct from the stage release events used in coordination violations and is not a causal harm label. A crossing completed in the same transition is already complete when release harmfulness is checked, so a legal simultaneous crossing/release is excluded.

Table 6: Standard-52 process metrics: final DAG completion and Progress AUC, as task-first percentages with frozen pointwise 95% task-bootstrap CIs (10,000 draws, seed 20260922). Three replicates are averaged within each task. No overall score.

Core-24

Model	DAG completion	Progress AUC
GPT-5.5	58.9 [48.1, 70.0]	35.5 [29.9, 40.8]
GPT-5-mini	41.0 [30.2, 53.0]	27.7 [21.1, 34.7]
GPT-5.4-mini	31.3 [20.9, 42.6]	20.4 [14.7, 26.6]
Qwen3.5-9B	13.8 [8.0, 22.8]	9.3 [6.5, 12.6]
Qwen3.5-27B	31.2 [23.3, 40.7]	19.9 [15.6, 24.6]
Qwen3.5-122B-A10B-FP8	16.8 [12.5, 21.9]	12.3 [9.5, 15.2]

Information-12

Model	DAG completion	Progress AUC
GPT-5.5	60.4 [52.1, 70.8]	55.2 [50.9, 59.9]
GPT-5-mini	51.4 [50.0, 53.5]	49.8 [49.6, 50.1]
GPT-5.4-mini	59.7 [52.4, 68.4]	54.1 [50.6, 58.1]
Qwen3.5-9B	49.3 [47.9, 50.0]	47.0 [45.5, 48.1]
Qwen3.5-27B	66.0 [60.8, 70.8]	57.7 [54.7, 60.8]
Qwen3.5-122B-A10B-FP8	59.4 [52.8, 67.4]	54.1 [50.7, 58.0]

Sync-v2-8

Model	DAG completion	Progress AUC
GPT-5.5	100.0 [100.0, 100.0]	47.8 [44.9, 51.0]
GPT-5-mini	97.1 [91.2, 100.0]	47.0 [43.7, 50.7]
GPT-5.4-mini	100.0 [100.0, 100.0]	47.9 [45.0, 51.3]
Qwen3.5-9B	91.4 [84.8, 97.2]	46.1 [43.6, 49.0]
Qwen3.5-27B	91.2 [73.8, 100.0]	46.6 [40.3, 51.8]
Qwen3.5-122B-A10B-FP8	91.2 [73.8, 100.0]	45.1 [39.7, 49.8]

Join-8

Model	DAG completion	Progress AUC
GPT-5.5	100.0 [100.0, 100.0]	55.9 [53.7, 58.0]
GPT-5-mini	99.5 [98.6, 100.0]	60.8 [58.8, 64.1]
GPT-5.4-mini	100.0 [100.0, 100.0]	55.7 [52.8, 58.2]
Qwen3.5-9B	87.7 [78.9, 95.4]	56.5 [51.4, 61.4]
Qwen3.5-27B	99.5 [98.6, 100.0]	60.9 [57.1, 65.0]
Qwen3.5-122B-A10B-FP8	97.7 [94.4, 100.0]	59.6 [56.6, 62.8]

A.7 Preparation versus Established-support Sensitivity

The original evaluator is not changed. For each recorded stage `premature_release`, the offline audit reads the authoritative actuator immediately before and after release. A true-to-false actuator transition is an *established-support interruption*; an actuator that is false in both snapshots marks *partial preparation rollback*. Earlier activation is recorded separately and does not turn an already-inactive pre-state into established support. This uses the full stage's Boolean outcome, not the release of an individual conjunct. Under an OR stage whose other branch keeps the actuator active, no full-support interruption occurs. A just-completed crossing remains excluded under the original simultaneous-transition rule.

The narrow sensitivity definition removes only partial-preparation release events from coordination violations. All other event types remain. Clean handoff is recomputed by stage: crossing must still

Table 7: Task-paired effects with pointwise 95% task-bootstrap confidence intervals. Proportion metrics are percentage-point differences; violations are fewer per 100 rounds. Positive values favor the left-hand condition. †Exploratory post-hoc comparison.

Comparison	ΔSR	ΔDAG	ΔAUC
GPT: Self-play vs No-Comm	-1.39 [-4.17, 0.00]	-1.53 [-3.26, -0.10]	-0.66 [-1.50, +0.07]
Gemini: Self-play vs No-Comm	+8.33 [0.00, +18.06]	+4.22 [+2.08, +6.50]	+1.50 [+0.69, +2.36]
Pooled: Self-play vs Cross-play	+0.69 [-2.78, +4.86]	-0.83 [-1.92, +0.22]	-0.26 [-0.73, +0.20]
Cross-play A vs B†	0.00 [-4.17, +4.17]	-1.63 [-3.64, +0.20]	-0.71 [-1.61, +0.12]

Comparison	ΔClean	Fewer violations/100 rounds
GPT: Self-play vs No-Comm	+4.20 [+0.24, +8.08]	+0.518 [+0.292, +0.767]
Gemini: Self-play vs No-Comm	+3.99 [+1.22, +6.86]	+0.035 [-0.052, +0.139]
Pooled: Self-play vs Cross-play	+0.36 [-1.32, +2.00]	+0.048 [-0.047, +0.143]
Cross-play A vs B†	-0.52 [-3.94, +2.73]	+0.078 [-0.050, +0.209]

Table 8: Success Rate and survivor-conditioned Rounds-to-Success. Rounds are the median across contributing task-level medians over successful replicates; failures are not imputed with the 80-round horizon. Brackets are pointwise 95% task-bootstrap confidence intervals.

Condition	SR (%)	Rounds-to-Success	Contributing tasks
GPT Self-play	38.89 [20.83, 58.33]	61.00 [50.00, 70.75]	10
GPT No-Comm	40.28 [20.83, 59.72]	61.00 [48.50, 70.00]	10
Gemini Self-play	31.94 [15.28, 50.00]	73.00 [40.00, 78.00]	9
Gemini No-Comm	23.61 [9.72, 38.89]	67.50 [40.00, 75.00]	7
Cross-play A: GPT(F)+Gemini(W)	34.72 [16.67, 54.17]	60.00 [40.00, 76.00]	9
Cross-play B: Gemini(F)+GPT(W)	34.72 [18.06, 52.78]	65.25 [55.00, 72.00]	10

succeed and no retained stage-associated violation may remain. It is not obtained by subtracting events from a success count. Event multiplicity and stage counts are kept distinct; metrics with no handoff opportunities remain undefined. Original and narrow definitions use identical task-paired bootstrap draws, including the ten certified-feasible tasks. These post-hoc sensitivity results test dependence on an interpretation choice, not a replacement benchmark or human construct validity.

All 63 original events in this paired corpus are stage releases: 58 established-support interruptions and five partial-preparation rollbacks, affecting five episodes and three two-controller ALL tasks. None of the latter had established support earlier. Removing five events adds only three clean handoffs (1558 to 1561 out of 2178 opportunities): one excluded event's crossing never completes, and another stage retains a different violation. Crossing successes remain 1616. The paired clean-handoff *communication* effects are unchanged because the relevant changes cancel within task; P2 communication results are unchanged. This does not extend to every contrast: Cross-A minus Cross-B clean handoff shifts from -0.522 to -0.349 pp. These distinctions are automatically derived from states, not human annotations.

The offline replay audit reproduces all 432 original episodes. Of 63 stage-release violations, 58 interrupt immediately established support and five roll back preparation while the actuator is false before and after release. Excluding only the latter changes the P0 violation interaction from -0.484 to -0.449 per 100 rounds (95% CI $[-0.674, -0.251]$); clean-handoff communication contrasts and all P2 communication contrasts are unchanged (Appendix A.7). The numerical contrast is not driven by these five events, but the two event meanings remain distinct.

A.8 Decision-level Interface Audit

We audit the 432 paired-study and 936 Standard-52 episodes separately. One executed round has two role-decision opportunities. A schema/parse failure, a retry, a forced fallback `WAIT`, a voluntary parsed `WAIT`, and a syntactically valid but environment-blocked move are different events. We report both affected decisions and rounds containing a fallback, rather than silently equating their

Table 9: Phase-II event audit (P0). Partial is excluded only in the narrow sensitivity; established-support interruptions and other coordination violations remain. Clean counts include capability handoffs and retain any other violation on that handoff. Multiple controller releases in one transition remain separate events.

Condition	Original	Partial	Estab.	Other	Clean (orig.→narrow)
GPT Self	6	0	6	0	281 → 281/363
GPT NoComm	35	1	34	0	264 → 264/363
Gemini Self	3	2	1	0	250 → 251/363
Gemini NoComm	5	1	4	0	231 → 232/363
Cross A	5	1	4	0	265 → 266/363
Cross B	9	0	9	0	267 → 267/363

Table 10: Maintain-event sensitivity on frozen Phase II. Raw Self−NoComm effects and the direct GPT−Gemini interaction, with paired task-bootstrap 95% intervals (10,000 draws; three replicates averaged first). The narrow definition excludes only controller releases while the authoritative actuator was false both immediately before and after the transition; all other violations, crossing completions, and denominators are unchanged. Clean effects are percentage points. This descriptive sensitivity does not replace the original metrics.

(a) Violations / 100 rounds

Population	Raw contrast	Original [95% CI]	Narrow [95% CI]
P0 (24)	GPT Self−NoComm	-0.52 [-0.77, -0.29]	-0.50 [-0.74, -0.28]
P0 (24)	Gemini Self−NoComm	-0.03 [-0.14, $+0.05$]	-0.05 [-0.16, $+0.03$]
P0 (24)	Interaction	-0.48 [-0.71, -0.28]	-0.45 [-0.67, -0.25]
P2 (10)	GPT Self−NoComm	-0.16 [-0.32, $+0.00$]	-0.16 [-0.32, $+0.00$]
P2 (10)	Gemini Self−NoComm	$+0.04$ [$+0.00$, $+0.12$]	$+0.04$ [$+0.00$, $+0.12$]
P2 (10)	Interaction	-0.20 [-0.36, -0.05]	-0.20 [-0.36, -0.05]

(b) Clean Handoff (pp)

Population	Raw contrast	Original [95% CI]	Narrow [95% CI]
P0 (24)	GPT Self−NoComm	$+4.20$ [$+0.24$, $+8.08$]	$+4.20$ [$+0.24$, $+8.08$]
P0 (24)	Gemini Self−NoComm	$+3.99$ [$+1.22$, $+6.86$]	$+3.99$ [$+1.22$, $+6.86$]
P0 (24)	Interaction	$+0.21$ [-4.50, $+4.54$]	$+0.21$ [-4.50, $+4.54$]
P2 (10)	GPT Self−NoComm	$+3.33$ [$+0.00$, $+6.67$]	$+3.33$ [$+0.00$, $+6.67$]
P2 (10)	Gemini Self−NoComm	$+1.00$ [-3.33, $+5.67$]	$+1.00$ [-3.33, $+5.67$]
P2 (10)	Interaction	$+2.33$ [-3.56, $+7.56$]	$+2.33$ [-3.56, $+7.56$]

denominators. Retry-attempt counts and terminal failures are not interchangeable: recovered failures can lack their original output/type, and missing attempt metadata is explicitly retained as unavailable. Pooled operational rates describe the collected calls; they are not task-bootstrap estimates of model capability.

The paired study contains 3045 fallback decisions among 63,172 opportunities, affecting 2951 of 31,586 rounds. Gemini's No-Comm fallback rate exceeds its self-play rate (10.061% vs. 7.163%), so interface reliability remains a competing mechanism for the communication contrast. GPT's corresponding rates are 1.867% and 2.330%. Across Standard-52, seven of 191,966 opportunities become fallback decisions; none is in the Qwen runs. Nonetheless, parsed blocked actions are common and remain distinct from schema errors. These audits neither repair the policies nor estimate their counterfactual performance without failures.

Phase II retains 6839 successfully retried decisions without the failed attempt's subtype; one connection-reset fallback has no usable attempt count. Its final errors include 2572 `JSONDecodeError` records whose provider-envelope versus model-text layer cannot be distinguished. Standard-52 has 144 recovered retries; its seven terminal fallback records are four schema-invalid and three malformed-output decisions. Open-weight `model_behavior_errors=0` in the original package reports counts terminal failures, not every failed attempt before recovery. Consequently, neither missing metadata nor a zero terminal-error counter is reported as a zero all-attempt parse-failure rate.

Table 11: Offline reliability audit. $D = 2R$ role opportunities for R executed rounds. Fallback percentages use D; affected-round percentages use R. Blocked moves are parsed non-WAIT actions rejected by environment rules, not schema errors. Standard-52 is split into Closed3/Open3 below (not additional episodes).

Dataset	Ep.	R	D	Fallback / D	Affected rounds / R	Blocked	Hazards
PhaseII	432	31,586	63,172	3,045 (4.820%)	2,951 (9.343%)	1,451	0
Standard-52	936	95,983	191,966	7 (0.004%)	7 (0.007%)	45,893	7
Closed3	468	45,220	90,440	7 (0.008%)	7 (0.015%)	12,474	5
Open3	468	50,763	101,526	0 (0.000%)	0 (0.000%)	33,419	2

Table 12: Executed decisions and environment outcomes by condition/model. FW is fallback WAIT; RW counts rounds with at least one fallback. VW is a parsed model-selected WAIT. B is environment-blocked parsed non-WAIT; H is a hazard event. FW/D and RW/R are distinct denominators. Full per-role and per-track data accompany the audit.

Condition/model	Ep.	R	D	FW	FW/D	RW	RW/R	VW	B	H
crossplay_geminiF_gptW	72	5246	10492	360	3.431%	357	6.805%	1662	263	0
crossplay_gptF_geminiW	72	5254	10508	395	3.759%	387	7.366%	1709	275	0
gemini_no_comm	72	5417	10834	1090	10.061%	1046	19.310%	2218	312	0
gemini_selfplay	72	5354	10708	767	7.163%	743	13.877%	1860	375	0
gpt_no_comm	72	5143	10286	192	1.867%	185	3.597%	1350	118	0
gpt_selfplay	72	5172	10344	241	2.330%	233	4.505%	1569	108	0
gpt-5-mini	156	15752	31504	4	0.013%	4	0.025%	14868	4742	2
gpt-5.4-mini	156	15249	30498	3	0.010%	3	0.020%	4492	4923	0
gpt-5.5	156	14219	28438	0	0.000%	0	0.000%	10057	2809	3
Qwen-Qwen3.5-122B-A10B-FP8	156	16548	33096	0	0.000%	0	0.000%	16254	10041	2
Qwen3.5-27B	156	16386	32772	0	0.000%	0	0.000%	7426	10343	0
Qwen3.5-9B	156	17829	35658	0	0.000%	0	0.000%	12779	13035	0

Table 13: PhaseII: role-model attribution by condition/track (F and W combined only when the same requested model is used). D is model-attributed role opportunities; A and X are observed provider attempts and retries; M is opportunities with missing attempt counts. A/X are lower bounds when M is nonzero. Intermediate error subtypes are unavailable; no exact attempt error rate is claimed. FW/VW/B/H follow Table 12.

Requested model	Condition/track	D	A	X	M	FW	VW	B	H
gemini-3.7-flash	crossplay_geminiF_gptW	5246	6300	1054	0	260	652	200	0
gemini-3.7-flash	crossplay_gptF_geminiW	5254	6450	1196	0	286	767	205	0
gemini-3.7-flash	gemini_no_comm	10834	13801	2967	0	1090	2218	312	0
gemini-3.7-flash	gemini_selfplay	10708	13202	2494	0	767	1860	375	0
gpt-5.5	crossplay_geminiF_gptW	5246	5605	359	0	100	1010	63	0
gpt-5.5	crossplay_gptF_geminiW	5254	5637	383	0	109	942	70	0
gpt-5.5	gpt_no_comm	10286	10990	705	1	192	1350	118	0
gpt-5.5	gpt_selfplay	10344	11069	725	0	241	1569	108	0

A.9 Effective Protocol Comparison

The paired study and Standard-52 are different runs, not matched horizon interventions (Table 17). The frozen prompt-builder hash and eight-record own-history/eight-message windows match; the visible clock changes from 80 to 200 Core rounds, GPT-5.5 reasoning changes from `low` to `none`, and retry handling changes. Qwen adds JSON-schema constrained decoding and shorter field limits. The actual Phase II backend deployment identity was not retained; the current corrected-runner file does not match the older matrix-pinned hash, so byte-identical historical runner provenance is not asserted. No private endpoint or credential is released. No-Comm explicitly discloses unavailable messaging; it is not covert message loss. These differences preclude attributing GPT-5.5's between-study results to extra rounds alone.

Table 14: Standard-52: role-model attribution by condition/track (F and W combined only when the same requested model is used). D is model-attributed role opportunities; A and X are observed provider attempts and retries; M is opportunities with missing attempt counts. A/X are lower bounds when M is nonzero. Intermediate error subtypes are unavailable; no exact attempt error rate is claimed. FW/VW/B/H follow Table 12.

Requested model	Condition/track	D	A	X	M	FW	VW	B	H
gpt-5-mini	Core-72	26892	26935	43	0	1	12835	4538	2
gpt-5-mini	Information-12	2160	2170	10	0	1	1504	202	0
gpt-5-mini	Join-8	1356	1366	10	0	0	264	0	0
gpt-5-mini	Sync-8-v2	1096	1104	8	0	2	265	2	0
gpt-5.4-mini	Core-72	26550	26574	24	0	1	3741	4818	0
gpt-5.4-mini	Information-12	2132	2139	7	0	2	642	102	0
gpt-5.4-mini	Join-8	898	902	4	0	0	23	1	0
gpt-5.4-mini	Sync-8-v2	918	921	3	0	0	86	2	0
gpt-5.5	Core-72	24494	24505	11	0	0	9208	2627	3
gpt-5.5	Information-12	2124	2124	0	0	0	724	181	0
gpt-5.5	Join-8	896	897	1	0	0	32	0	0
gpt-5.5	Sync-8-v2	924	924	0	0	0	93	1	0
Qwen-Qwen3.5-122B-A10B-FP8	Core-72	28290	28291	1	0	0	14846	9161	2
Qwen-Qwen3.5-122B-A10B-FP8	Information-12	2144	2144	0	0	0	801	512	0
Qwen-Qwen3.5-122B-A10B-FP8	Join-8	1394	1394	0	0	0	307	153	0
Qwen-Qwen3.5-122B-A10B-FP8	Sync-8-v2	1268	1268	0	0	0	300	215	0
Qwen3.5-27B	Core-72	28014	28014	0	0	0	6479	9382	0
Qwen3.5-27B	Information-12	2156	2156	0	0	0	381	738	0
Qwen3.5-27B	Join-8	1268	1268	0	0	0	238	10	0
Qwen3.5-27B	Sync-8-v2	1334	1334	0	0	0	328	213	0
Qwen3.5-9B	Core-72	28218	28238	20	0	0	9855	11181	0
Qwen3.5-9B	Information-12	2160	2162	2	0	0	585	862	0
Qwen3.5-9B	Join-8	2628	2633	5	0	0	898	666	0
Qwen3.5-9B	Sync-8-v2	2652	2654	2	0	0	1441	326	0

Table 15: Provider accounting over executed episodes. A/X: observed attempts/retries (lower bounds if M, missing opportunities, is nonzero). Rec: opportunities recovered after retry; their intermediate error subtypes were not saved. Err: final runner exceptions. B/I/S/FW are adapter-reported behavior, infrastructure, systematic-interface and substituted-WAIT counters; – means unavailable, not zero. B is not a complete error-attempt tally: the adapter omits recovered parsing errors and increments B only on final behavior failure. An exact attempt error rate is therefore unknown even where A is complete.

Dataset	A	X	M	Rec	Err	B	I	S	FW
PhaseII	73054	9883	1	6839	3045	–	–	–	–
Standard-52	192117	151	0	144	0	14	14	0	7

Table 16: Phase II persisted final diagnostics, aggregated across roles and conditions. JSON is a JSONDecodeError at an unknown parsing layer (provider envelope or model output), not necessarily a schema violation. No text means no model text in the provider envelope. No object means no JSON object in model output. These are final decision failures, not all failed provider attempts; earlier/recovered failure subtypes are unavailable.

Requested model	Fallbacks	JSON	No text	No object	HTTP	Timeout	Reset
gemini-3.7-flash	2403	2368	0	32	2	1	0
gpt-5.5	642	204	421	13	2	1	1

B Additional Phase II Results

B.1 Paired Communication and Cross-play

GPT's paired SR effect is -1.39 pp [-4.17, 0.00], clean handoff +4.20 pp [+0.24, +8.08], and the violation reduction +0.518/100 rounds [+0.292, +0.767]. Absolute violation counts are 6 over 5,172 rounds versus 35 over 5,143; all 24 leave-one-task-out reduction estimates stay positive (0.45–0.54). These original results are unchanged, and remain subject to the reliability confound documented above.

Table 17: Observed Phase II versus corrected Core protocol, from freezes, runner/configuration and trace records rather than manuscript descriptions. Shared task layouts do not imply an identical inference protocol. Hashes are abbreviated here; complete hashes and sanitized route/revision metadata are in the audit JSON. Phase II backend deployment identity is unknown; the current corrected-runner file differs from the matrix-pinned hash, so byte-identical historical runtime provenance is not asserted.

Field	Phase II	Corrected Core (Closed3 + Open3)
Version / scope	phase2_experiment.v1; Diagnostic-24 x 6 conditions	corrected protocol v2; 24 selected Core tasks (key Core-72)
Round cap	80	200
User prompt / system message	build_agent_prompt; no separate system message	same frozen builder; no separate system message
Prompt source SHA-256	a6d06b75e3c8...	a6d06b75e3c8...
Prompt contract SHA-256	not separately frozen; source hash above	862349ef54d6...
Own history / public message history	last 8 compact own records / last 8 delivered messages	same 8 / 8; not full transcript
Reasoning / output-token cap	low (GPT Responses; Gemini Chat Completions) / 800	GPT-5.5 none; GPT-5-mini and GPT-5.4-mini low; Qwen omit plus enable_thinking=false / 800
Provider route	OpenAI-compatible adapter; Responses (GPT), Chat Completions (Gemini); actual endpoint identity not persisted	closed: local Copilot gateway/Responses; open: pinned vLLM/Chat Completions
Resolved identity	GPT: gpt-5.5-2026-04-23; Gemini: gemini-3.7-flash (non-fallback records); gpt-5.5 fallback alias is not resolved-identity evidence	closed: gpt-5.5, gpt-5-mini, gpt-5.4-mini; open: served Qwen aliases plus pinned revisions (JSON)
Remaining budget	visible: clock.max_rounds, rounds_used, rounds_remaining	visible in same clock fields; Core cap 200, not 80
Communication / No-Comm	one-round delay; No-Comm explicitly says disabled, hides inbox and forces message=null	self-play communication enabled; no No-Comm arm
Structured output	prompted JSON; local parsing	closed: prompted JSON; open: JSON-schema-constrained decoding plus shorter text-field limits
Retry / final failure	one generic retry with repair note; final exception -¿ runner WAIT	one retry, unchanged prompt; final model-output failure -¿ WAIT; infrastructure exhaustion unscored; interface gate stops entry
Temperature / seeds	1.0; requested 2026091401–03, not guaranteed	1.0; 2026091801–03 requested where supported; GPT-5.4-mini and Qwen omit seed
Planning / movement	0 planning rounds; simultaneous frozen observations; one cell/role/round	same

Table 18: Phase II task-first condition estimates with pointwise 95% task-bootstrap confidence intervals. SR, DAG completion, Progress AUC, and Clean Handoff are percentages; violations are per 100 executed rounds.

Condition	SR	DAG completion	Progress AUC	Clean Handoff	Viol./100 rounds
GPT Self-play	38.89 [20.83, 58.33]	85.48 [79.50, 90.99]	46.88 [43.07, 50.55]	83.60 [76.50, 90.59]	0.104 [0.035, 0.191]
GPT No-Comm	40.28 [20.83, 59.72]	87.01 [82.02, 91.83]	47.54 [44.10, 51.00]	79.39 [71.55, 87.06]	0.622 [0.397, 0.865]
Gemini Self-play	31.94 [15.28, 50.00]	79.62 [73.20, 86.04]	44.41 [40.29, 48.66]	76.76 [68.72, 84.69]	0.052 [0.000, 0.122]
Gemini No-Comm	23.61 [9.72, 38.89]	75.40 [68.14, 82.72]	42.90 [38.37, 47.48]	72.77 [63.11, 81.97]	0.087 [0.017, 0.174]
Cross-play A: GPT(F)+Gemini(W)	34.72 [16.67, 54.17]	82.56 [76.17, 88.80]	45.55 [41.55, 49.41]	79.56 [72.26, 86.67]	0.087 [0.018, 0.158]
Cross-play B: Gemini(F)+GPT(W)	34.72 [18.06, 52.78]	84.20 [78.23, 89.89]	46.26 [42.50, 50.08]	80.08 [72.50, 87.34]	0.165 [0.063, 0.281]

Table 19: Direct Model×Communication interaction on raw metrics, $(\text{GPT Self} - \text{GPT NoComm}) - (\text{Gemini Self} - \text{Gemini NoComm})$, on all 24 tasks (P0) and on the 10 certified-feasible tasks (P2). SR, completion, AUC, and Clean Handoff in pp; violations per 100 rounds (raw, so negative means messaging reduces GPT's violations more; the benefit-oriented value is the negation, e.g. $+0.48$ $[+0.28, +0.71]$ in P0).

Metric (raw contrast)	P0 (24 tasks)	P2 (10 tasks)
SR	-9.72 $[-20.83, -1.39]$	-23.33 $[-46.67, -3.33]$
DAG Completion	-5.74 $[-8.68, -3.02]$	-3.50 $[-7.80, -0.18]$
Progress AUC	-2.17 $[-3.33, -1.06]$	-0.71 $[-2.20, +0.37]$
Clean Handoff	$+0.21$ $[-4.50, +4.54]$	$+2.33$ $[-3.56, +7.56]$
Viol./100 rounds	-0.48 $[-0.71, -0.28]$	-0.20 $[-0.36, -0.05]$

Holding GPT in role F and replacing its GPT(W) partner with Gemini(W) yields effects favoring GPT Self-play on completion (+2.92 pp [+1.45, +4.57]), AUC (+1.33 pp [+0.61, +2.11]), and Clean Handoff (+4.04 pp [+1.83, +6.41]). Holding Gemini in F and replacing Gemini(W) with GPT(W) yields -4.58, -1.85, and -3.32 pp on the same measures (positive would favor Gemini Self-play). The direct Cross-play A-versus-B SR difference is 0.00 pp [-4.17, +4.17]; completion -1.63 [-3.64, +0.20],

AUC -0.71 [-1.61, +0.12], Clean Handoff -0.52 [-3.94, +2.73], and violations +0.078 [-0.050, +0.209] per 100 rounds all include zero, so no role-assignment effect is established.

Message behavior. GPT sends 62.4 messages per 100 rounds versus Gemini's 13.9, 41.3% of them repeated confirmations (Gemini 2.7%), and its partner moves 2.00 versus 1.41 rounds after a READY/HOLDING message. These are descriptive associations with execution behavior, not evidence that message content causes improved coordination; condition-dependent output reliability remains a competing mechanism.

B.2 Conditional Speed

Paired Rounds-to-Success effects are inconclusive for GPT (-0.75 rounds [-2.00, +0.50], 10 tasks), Gemini (0.00 [-1.00, +5.00], 7 tasks), pooled cross-play (-0.25 [-1.50, +2.00]), and direct A versus B (0.00 [-4.00, +5.00]). Failures are never recoded as 80-round successes.

Spatial layout. On 12 held-out layouts with unchanged DAGs, seen-minus-held-out SR is -2.8 pp $[-8.3, 0.0]$ (GPT) and $+2.8$ pp $[-5.6, +11.1]$ (Gemini). These intervals do not establish equivalence.

B.3 Horizon Feasibility Partition

The certified-feasible subset widens intervals to include zero for GPT completion, Gemini completion/AUC/clean handoff, and the AUC interaction. GPT's violation reduction remains non-negative (+0.16 [0.00, +0.32]/100 rounds), and the direct interactions on SR, completion, and violations exclude zero. Gemini's raw violation effect changes sign from -0.035 to $+0.042$/100 rounds. These are protocol-specific sensitivity results, not universal sign stability.

A Core task is *provably infeasible* at 80 rounds if the maximum of F's and W's shortest-path distances, computed with every non-wall cell passable and all coordination removed, exceeds 80; it is *certified feasible* if some existing evaluator-valid successful trajectory finishes within 80. Diagnostic-24 splits into 10 certified, 9 provably infeasible, and 5 unresolved tasks. Observed SR on the infeasible tasks is zero in every condition. On certified tasks only, SR is 93.3% (GPT Self), 96.7% (GPT No-Comm), 76.7% (Gemini Self), 56.7% (Gemini No-Comm), and 83.3% (both cross-play directions); these are descriptive feasible-task rates.

B.4 Diagnostic Repair Ablation

Each of the 285 timeouts is restored at its last evaluator-progress checkpoint. A shared controller-stage repair executor attempts each selector's top-1 target. The historical usefulness rule credits blocker removal, increased required progress, or the selected stage being complete afterward; it does not require that last condition to be newly achieved. We retain that result as historical accounting and audit the criterion separately below. Abstention and unsupported targets count as not useful. Table 20 gives the exact selectors rather than treating their names as controlled information ablations.

All methods share the task's controller-stage mapping and executor. Full DAG may select a capability blocker that this executor cannot repair; predicate-only skips non-stage-mappable nodes. Crucially, predicate-only reads *completion flags from the full edge-aware evaluator*, in its topological output order; it does not re-evaluate raw predicates with edges removed. Local-event first uses the last coordination violation's handoff identifier, otherwise the latest door/platform change. Thus both its event summary and predicate-only's flags can contain information derived from the original graph. No candidate-matched, edge-only toggle was run, and these numbers cannot isolate the causal contribution of dependency edges.

Support of the Full-DAG selector. Full DAG selects an executable controller stage on 185 trajectories. On the other 100, its own selection is a capability blocker (50) or a terminal/unmapped blocker for which it abstains (50); the controller-stage executor cannot act on these. This does not mean no repair exists: every one of the 285 trajectories has at least one useful controller-stage target, and on these 100 trajectories predicate-only finds one 19 times and local-event 57 times. The subset is defined only by the Full-DAG selector's coverage (its target is an executor-supported controller stage); usefulness is not a selection criterion. On this selector-defined supported subset, useful rates are 0%,

Table 20: Selector comparison on the 285 frozen Phase-II timeout traces. The selector input and target mapping are taken from `analyze_dag_utility_ablation.py`; all methods use the same deterministic controller-stage repair executor. "Stage" is an executor-supported `stage_i` target; capability and terminal/unmapped selections are not executable by this executor. This is a selector ablation, not an isolated DAG-edge toggle.

Method	Selector input	Target mapping / candidate restriction	Stage	Cap.	Abstain	Useful
Terminal-only	No target; selector returns `None`	No stage candidate	0	0	285	0
Predicate-only	First incomplete evaluator `node_results` item with a stage-mappable predicate	`stage_i` only; evaluator completion still uses the full DAG	221	0	64	139
Local-event	Last violation `handoff_id`; otherwise last door/platform state-change event	Actuator-to-stage map; `stage_i` only	243	0	42	240
Full DAG	Failure `handoff_id`, else first incomplete required node	`stage_i`, capability, or abstain; only `stage_i` is executable	185	50	50	185

Implementation note. `stage_i` denotes the record-stage index. `stage_of` recognizes controller `<id>_on`, actuator `<id>_on`, and crossing `<id>_crossed`; the audit enumerates these stage candidates in record order, but each selector receives credit only for its selected top-1 target under the shared executor. Useful is the existing intervention criterion, not a causal edge-effect or human-validity measure.

64.9%, 98.9%, and 100%: all 185 Full-DAG targets were observed to be useful. This conditional, exploratory result, computed in the same analysis run as the primary results, is not an accuracy over all failures and does not replace the 285-trajectory comparison.

Strata and collision signature. The historical MAINTAIN stratum means that *any incomplete maintenance handoff remains*, possibly downstream; it is not the selected blocker's type. Twenty-nine executable Full-DAG targets in that stratum are actually ENABLE stages. We therefore do not interpret its percentages as Maintain-blocker performance. Local-event exploits regular staged task structure. Collision groups use only (outcome, executed rounds, completed-predicate count, local-violation count), not full predicate identities or histories.

Table 21: Historical top-1 repair usefulness over the 285 Phase II timeouts (22 tasks); this original criterion can credit a stage already complete before repair. The new-progress sensitivity below separates that artifact. *Pooled*: fraction of trajectories; *task-first*: mean of per-task rates. Paired effects in the text are task-first (10,000-draw task bootstrap). Abstentions and targets the controller-stage executor cannot run count as not useful.

Method	Pooled	Task-first	Executable target	Unsupported / abstain	Useful
Terminal-only	0.0%	0.0%	0	0 / 285	0
Predicate-only	48.8%	38.0%	221	0 / 64	139
Local-event	84.2%	79.1%	243	0 / 42	240
Full DAG	64.9%	46.8%	185	50 / 50	185

New-progress criterion sensitivity. The original condition `target_completed` only examines post-intervention completion. In 102 of 240 credited local-event repairs and 25 of 139 predicate-only repairs, the selected stage was already complete at the restored checkpoint, and neither blocker removal nor positive completion change occurs. No Full-DAG target receives this kind of credit. We re-score only existing intervention records, requiring blocker removal, positive progress, or a selected stage newly complete after repair. The narrower criterion retains the original executor-validity gate; absent raw validity fields are not invented. Checkpoint completion is recovered from hash-verified canonical completion timestamps.

The resulting counts are 0/114/138/185 out of 285 for terminal/predicate/local/Full DAG. Task-first Full-minus-predicate is $+15.60$ pp $[-4.16, +35.67]$; Full-minus-local is -4.92 pp $[-25.88, +14.27]$. Thus the strong historical local-event advantage is criterion-dependent. This retrospective check neither creates new interventions nor establishes a fair edge-only comparison, equivalence, or human-validated usefulness. Original scores remain visible to prevent a silent change of evaluation.

B.5 FAILURE-LOCALIZATION EXAMPLE

Figure 2 shows a complete evaluator record for one failed development-corpus trajectory.

Table 22: Retrospective repair-criterion sensitivity on the same 285 timeouts (22 tasks), without new interventions. Counts use all 285 traces; task-first rates average within task before averaging tasks. New-progress credit retains the original executor-validity gate and requires recorded blocker removal, positive DAG-completion delta, or selected-stage completion newly achieved after the restored checkpoint. Removed credit is for already-completed stages without blocker removal or progress. The original graph-based outcomes, selected targets, and executor remain confounded; neither criterion is human-validated usefulness or a fair edge ablation.

Method	Original useful	New-progress	Removed	Original task-first	New task-first
Terminal-only	0/285	0/285	0	0.0%	0.0%
Predicate-only	139/285	114/285	25	38.0%	31.2%
Local-event	240/285	138/285	102	79.1%	51.8%
Full DAG	185/285	185/285	0	46.8%	46.8%
Task-paired Full-DAG differences (pp), 95% intervals; 10,000 task-bootstrap draws					
Full − Predicate-only	$+8.8\,[-10.1, +27.8]$			$+15.6\,[-4.2, +35.7]$	
Full − Local-event	$-32.3\,[-49.8, -16.6]$			$-4.9\,[-25.9, +14.3]$	

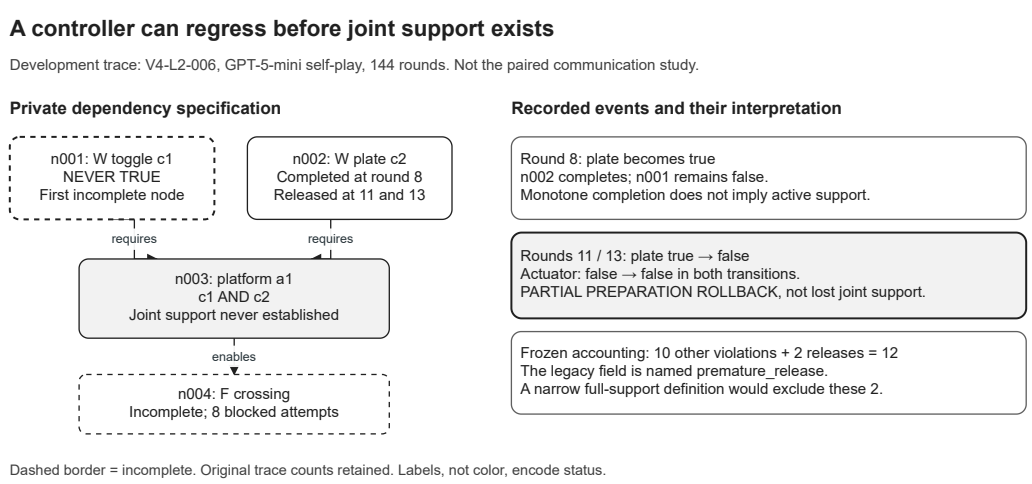


Figure 2: Evaluator record for a development-corpus trajectory (V4-L2-006, not the paired study). Node n002 completes at round 8, but n001 never becomes true, so their joint support at n003 never holds. Plate releases at rounds 11 and 13 are *partial preparation regressions*, not interruptions of established joint support. The legacy `premature_release` field counts them with ten other dependency violations, giving twelve coordination violations. This illustrates the rule's semantic boundary, not human-validated harmfulness; the graph localizes n001.

C Benchmark Details

C.1 Generation and Binding Checks

Construction emits the map, wiring, role/stage metadata, private DAG, and node-to-observation bindings together. An instance is admitted only when a deterministic unit-step witness reaches team success, every node has an executable binding, and every DAG edge has witness evidence. For all 72 tasks, witnesses succeed and replay exactly; every controller stage has a verified cross-agent no-bypass cut; permanent `WAIT` by either role prevents success; and 68 tasks require assistance in both role directions.

C.2 Contraction Algorithm

Algorithm 1 is implemented by `scripts/audit_dependency_edges.py`; its guards require 72 tasks, exactly 367 edges, cross-role endpoints, and verified bindings and evidence for every row.

Table 23: Capability differences among representative collaboration, executable-structure, and failure-diagnosis benchmarks. "Not established" means that the inspected primary source does not establish the property; it is not evidence of absence.

Benchmark / work	Interactive	Multi-agent	Cross-agent dependency	Process metrics	Executable task graph	Failure localization	Cross-play
Overcooked-AI	Yes	Yes	Partial	Not established	Not established	Not established	Yes
Collab-Overcooked	Yes	Yes	Yes	Yes	Not established	Partial	Not established
MultiAgentBench	Yes	Yes	Yes	Yes	Not established	Not established	Not established
TeamCraft	Yes	Yes	Yes	Yes	Not established	Partial	Not established
CRAB	Yes	Yes	Not established	Yes	Yes	Partial	Not established
VillagerBench / VillagerAgent	Yes	Yes	Yes	Yes	Partial	Not established	Not established
CoCoBench	Yes	Yes	Yes	Yes	Not established	Yes	Not established
TraceElephant	Partial	Partial	Yes	Yes	Not established	Yes	Not established
FWCOLLAB	Yes	Yes	Yes	Yes	Yes	Partial	Partial

Definitions and qualifiers. An executable task graph is a machine-readable node–edge dependency structure whose predicates and edges are directly consumed by a scheduler or deterministic evaluator, including post-rollout execution against an authoritative trajectory; a plan, action sequence, search graph, or agent-communication graph alone does not qualify. "Partial" covers layout-specific dependence (Overcooked-AI), manual or checkpoint-level diagnosis (Collab-Overcooked, TeamCraft, and CRAB), a method-generated rather than benchmark-owned DAG (VillagerAgent), and TraceElephant's mixed static/dynamic and single-/multi-agent corpus. Overcooked-AI's cross-play entry is qualified to held-out proxy-human and human partners. FWCOLLAB cross-play uses fixed independently developed pretrained models in both role directions; it is not a held-out policy-learning split. Its failure localization identifies structural blockage, not causal attribution (hence Partial). FWCOLLAB is an interactive symbolic grid; it does not test perception, continuous control, or GUI interaction.

Algorithm 1 Contract native DAG to cross-agent dependencies

for each task (by id): validate DAG and binding coverage
  for each controller stage: $C \leftarrow$ controller nodes; $e \leftarrow$ actuator node; $t \leftarrow$ crossing node
    require supporter $\neq$ traveler; owners(C)=supporter; owner(e)=environment; owner(t)=traveler
    require verified bindings; Boolean evidence for each $c \rightarrow e$; no-bypass evidence for $e \rightarrow t$
    label $\leftarrow$ MAINTAIN if any controller is a plate, else ENABLE; emit one edge
  if collaborative capability capsule: require distinct roles and mandatory cut; emit ENABLE edge
require every emitted edge crosses roles and is fully verified; **return** edges

All 45 capability edges are ENABLE; the 322 controller-stage edges split into 171 ENABLE and 151 MAINTAIN.

C.3 DIAGNOSTIC-24 SELECTION

The structural selector allocates 2,3,3,4,4,4,4 tasks to L1–L7 and maximizes join, controller-pattern, and dependency-structure coverage without reading model outcomes. Compared with Full-72, Diagnostic-24 has nearly identical mean DAG depth (16.1 vs. 16.4), node count (18.7 vs. 18.9), handoff count, movement lower bound (73.2 vs. 74.0), and best certified rounds (95.1 vs. 95.3); it contains slightly more MAINTAIN and fewer ENABLE dependencies per task (SMD $+0.34$ and -0.38).

C.4 LEGACY COLLABORATION TAGS

Earlier versions reported derived tags C1 (maintained plate stage), C2 (persistent lever/toggle stage), C3 (stages in both supporter directions), and C6 (at least three stages and two role alternations). They are diagnostic summaries over Full-72, not same-level primitives, and are superseded by the contracted ENABLE/MAINTAIN semantics and the extension tracks.

C.5 EXTENSION-TRACK VALIDATION

Sync-v2-8. Sync-v2 keeps the structural variants and K sequence (1,1,1,2,3,2,2,3) of an earlier Sync-8 prototype. The prototype enforced the duration requirement only inside the evaluator, not

in the public rules; it is superseded and used for no reported model result. Each Sync-v2 task has a certificate with separately counted checks, each 8/8: successful witness (30–36 rounds, within the 80-round budget), delayed-arrival success, wrong-duration negative, interrupted-overlap negative, deterministic replay of these four traces, evaluator validation, public-semantics and gate-reads-latch checks, corridor no-bypass, and necessity of both roles. An exhaustive finite-state search explores 3,707 reachable states and 92,475 transitions and finds 0 states with team success but no synchronization. The conformance mutation suite and the scripted alternative trajectories were run on the prototype, not on Sync-v2.

Join-8 and Information-12. Join-8: 8/8 witnesses, replays, bindings, permanent-`WAIT` failures, and no-bypass checks; conformance 59/59 applicable cases. Information-12: four wiring permutations crossed with three routes; wrong-first controller witnesses cannot finish within 30 rounds.

Certificate-based reference. A deterministic reference replays, per task, the shortest existing successful certificate (authored witnesses or audited successful traces). It succeeds on 24/24 Core, 12/12 Information, and 8/8 Join tasks at the corrected budgets. It was run before Sync-v2 existed; for Sync-v2 the certified witnesses above play the same role. It shows that legal solutions exist within budget; it is neither an independent planner nor an efficiency-optimal bound.

D Evaluator Conformance

Over the Full-72 witnesses, eight case families are generated per task: successful witness, irrelevant `WAIT`, irrelevant reversible move, parent–child order swap, hold release, hold deletion, closed-gate crossing, and truncation at a node. Of 576 generated cases, 555 are applicable (a mutation is applicable only if it changes authoritative state as intended) and all pass: node-completion accuracy 100% over 1,260 events, equality-binding round accuracy 100% over 894, failure-stage accuracy 72/72, and MAINTAIN-violation detection 130/130. Join conformance passes 59/59 applicable cases; the corresponding 37/37 synchronization suite was run on the superseded Sync-8 prototype. Of the 40 scripted alternative trajectories in Section 7.3, 15 Core trajectories replay each authored witness's per-role non-`WAIT` action queue under a different cross-role schedule; one Core trajectory takes an unused `any` branch via a scripted route; 16 Join trajectories are built by the extension route constructor with W-first order or a post-join detour; and 8 synchronization trajectories lose and recover overlap on the Sync-8 prototype. They differ from the witnesses but are not generated independently of them.

E Interface Admission and Excluded Runs

The timeline is as follows. (1) In the original leaderboard collection, the Gemini-3.7-Flash and Gemini-3.5-Flash route (a local OpenAI-compatible gateway, reasoning `none`) yielded schema-valid decisions in 36.95% and 1.41% of cases; most failures were JSON truncated mid-string, and a reference parser recovered none. (2) A post-hoc offline audit could not attribute the truncation to the model or the gateway, because provider envelopes had not been retained. (3) Before any corrected rerun, a route-selection protocol was frozen and applied as a 12-request compatibility check per candidate cell; no route was stable for both models. (4) A 100-request, four-stratum stress protocol was then frozen and applied: Gemini-3.7-Flash produced 99/100 valid responses with reasoning `none` and 73/100 with `low`, both failing the all-valid criterion, while GPT-5.4-mini passed 100/100 and took the open slot. These gates were written after the original failure was diagnosed and before the corrected collection, not before all historical experiments. The exclusion applies to the tested serving configuration and carries no capability interpretation. Phase II trajectories remain replay-valid, but their runtime record does not establish the same or a different backend deployment. The new decision audit finds materially nonzero, condition-dependent fallback rates there (Appendix A.8); its communication effects must therefore include this reliability mechanism rather than assuming interface equivalence.

F VISUAL GUIDE AND A RENDERED MODEL TRAJECTORY

From the original illustration to an actual episode. The author's complete original artwork, including both spatial-layout panels, is reproduced in Figure 1. Those panels illustrate the construction idea; the independently authenticated JOIN-06 episode below supplies the real spatial execution evidence.

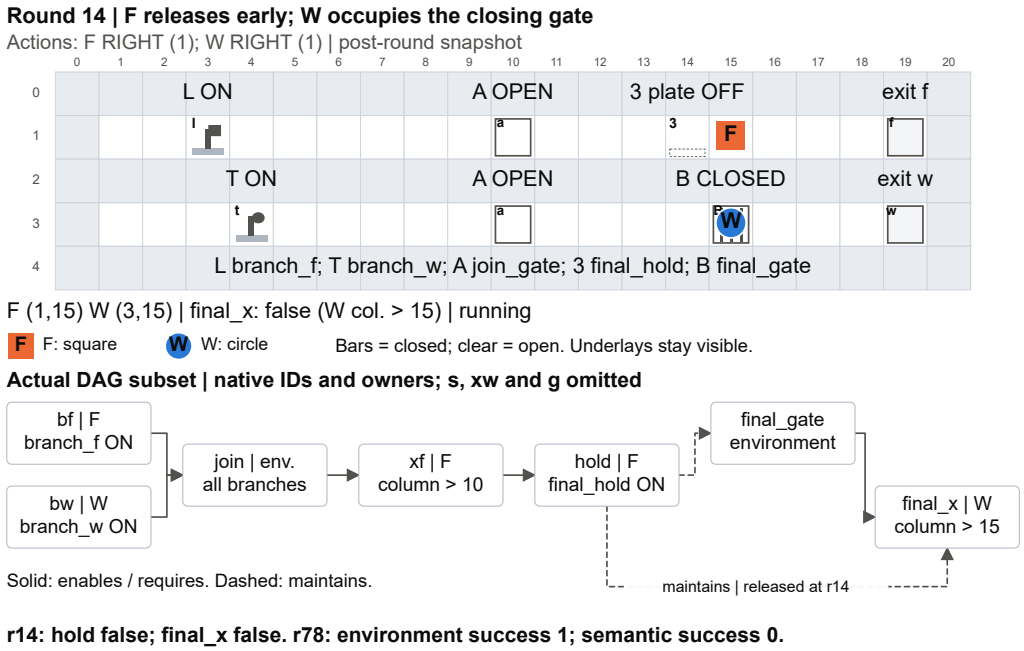


Figure 3: Actual post-round-14 JOIN-06 scene and native DAG subset (GPT-5-mini, replicate 1). F releases the plate before W completes its crossing; the same trace later reaches both exits. This is an authenticated model trajectory, not one of the conceptual layouts in the original illustration.

F.1 REAL EPISODE: COMPLETION AFTER A SUPPORT VIOLATION

The case is GPT-5-mini homogeneous self-play, replicate 1, JOIN-06, from the frozen Standard-52 data: 78 executed rounds, environment success, and one post-join `persistent_condition_released_early` violation. It is selected to illustrate the already-audited outcome/violation distinction, not as a representative success-rate sample. The model does not observe the private DAG shown in the figure. Full map views use one scale and the authoritative terrain, object layout, and dynamic states; no positions or transitions are interpolated.

A deterministic replay verifies all transition hashes and the terminal state before drawing. In round 14, F leaves the holding plate while W enters the gate using the open pre-transition state. The post-transition gate closes, and W has not yet crossed its far boundary; the recorded violation is therefore not the legal simultaneous completed-crossing exception. At round 15 support is restored and W crosses. Both roles reach their exits only at round 78. The final DAG completion of 1.0 does not erase the earlier violation.

The keyframe table reports executed actions and predicate events; the supplementary action CSV contains all 78 rounds, and the rendered sequence contains the initial state plus all 78 post-states. These are graphical renderings of a real symbolic trajectory, not screenshots of a commercial game or new model-generated execution. Only public action/state information is exported, not private reasoning or raw provider responses.

Round	F action	W action	F col.	W col.	Hold	Gate	Status
0	–	–	1	1	0	0	running
2	RIGHT	RIGHT	3	3	0	0	running
3	RIGHT	RIGHT	4	4	0	0	running
13	RIGHT	RIGHT	14	14	1	1	running
14	RIGHT	RIGHT	15	15	0	0	running
15	LEFT	RIGHT	14	16	1	1	running
77	RIGHT	WAIT	18	19	0	0	running
78	RIGHT	WAIT	19	19	0	0	success

Table 24: Selected submitted actions from the replay-verified JOIN-06 trace (GPT-5-mini, replicate 1). States are post-round; snapshot 0 is initial. Coordinates are zero-based; F stays in row 1 and W in row 3. Hold and gate are Boolean states. The complete 78-row action ledger is `evidence/revision9_case/actions.csv`. Environment success at round 78 does not erase the round-14 maintenance violation.

G Visual Summaries of the Frozen Results

The following figures complement the numerical tables rather than introduce new experiments. Information and Join compare different outcome/semantic indicators with the frozen marginal task-bootstrap intervals; those intervals do not test their paired difference. The reliability panels use explicit decision-opportunity denominators and keep environmental blocking separate from invalid responses. The repair plot changes only the credit criterion on the same stored interventions. Each figure is accompanied by a source-value CSV and hash manifest.

H Planned Controlled Experiments

Depth and length. Depth correlates with completion ($\rho \approx -0.90$ to -0.95) and movement lower bound ($\rho = 0.994$); controlling the latter reduces its partial correlation to $[-0.27, +0.11]$. Depth and execution length remain entangled; the matched counterfactual suite is not evaluated (Appendix H).

Counterfactual specificity suite. Eight matched pairs (16 tasks): four pairs in which a dependent twin requires the partner for the target crossing while a control twin binds the gate to the traveler, with identical geometry, mechanism inventory, and movement lower bound (maximum mismatch 0); and four pairs that keep an ENABLE, MAINTAIN, synchronization, or join dependency while lengthening corridors by 24 moves. All 16 tasks pass certificate, replay, and evaluator checks. No model has been evaluated on this suite.

Centralized baseline. A single-controller protocol that issues both roles' actions is frozen but not run.

Budget sensitivity and independent validity. Information-12 reruns at 45/60 rounds have not been performed; truncated existing trajectories cannot establish their counterfactual outcome. Independent, blinded human annotation of collision-group blockers is also absent. Such annotations would need an explicit rubric, adjudication, and inter-annotator agreement before comparison with evaluator labels. Neither automatic conformance nor an LLM-generated annotation substitutes for that evidence. A controlled edge ablation with matched selector inputs and executor-supported target coverage remains unperformed.

I Data Availability

All task, evaluator, protocol, and analysis assets are enumerated by versioned SHA-256 manifests. An anonymized benchmark and code release will accompany submission. Provider responses are retained as auditable traces subject to provider terms; credentials are never included.

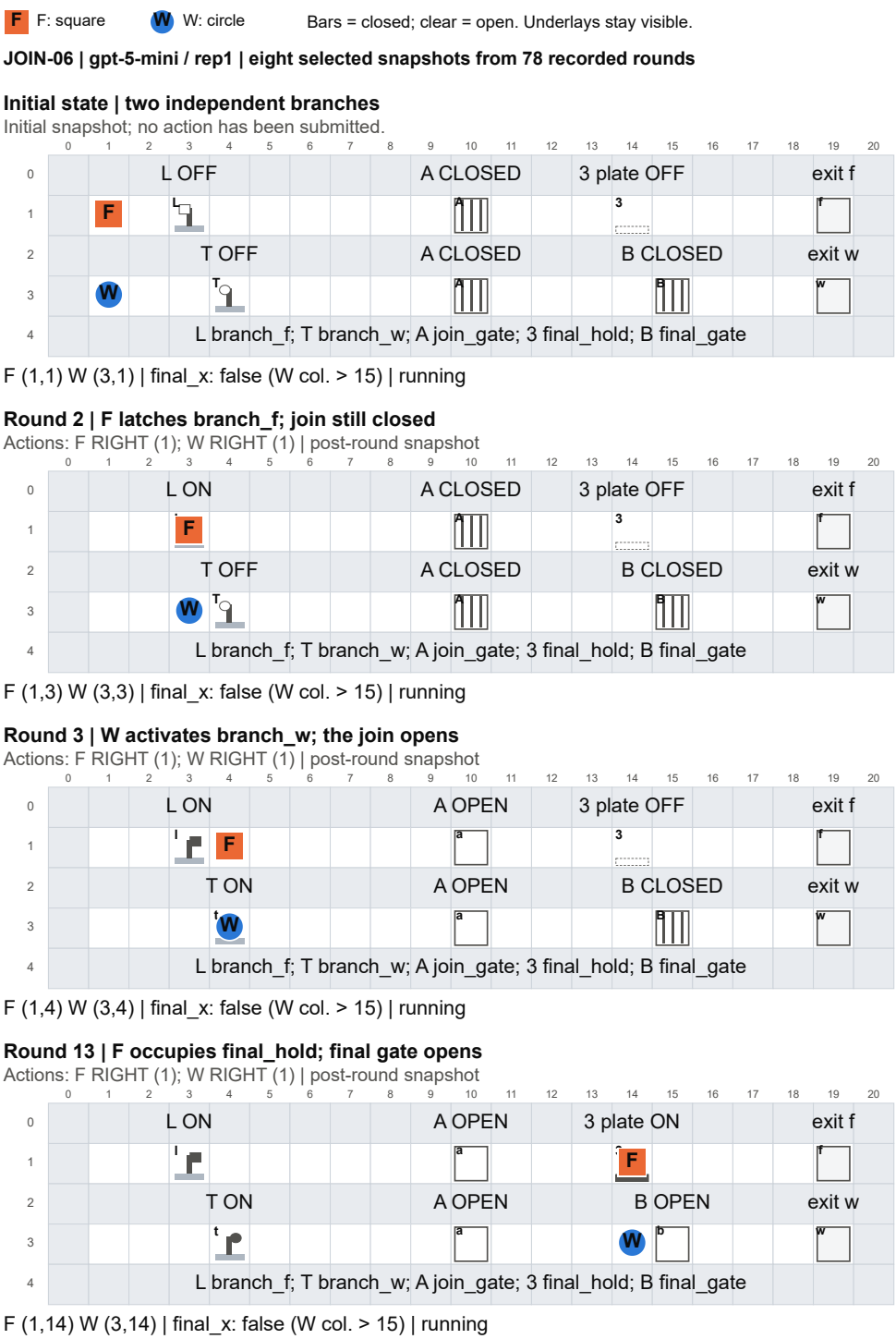


Figure 4: JOIN-06 model trajectory, part 1/2: initial state and post-rounds 2, 3 and 13. F and W establish their branches; at round 13 F holds the final plate and opens W's gate. All frames use the same complete map and scale.

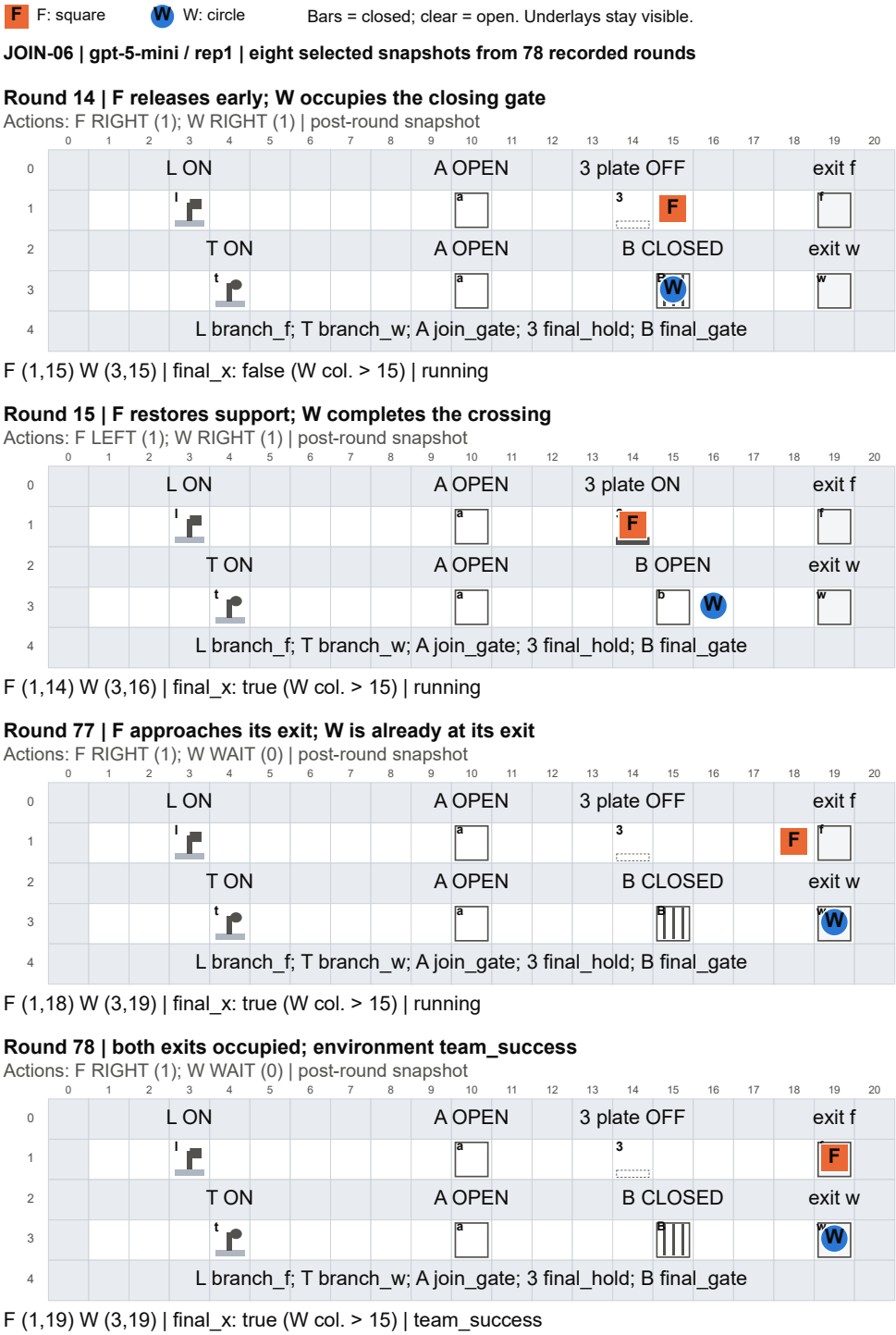


Figure 5: Same trajectory, part 2/2: post-rounds 14, 15, 77 and 78. At 14 F releases the plate while W is at column 15, before crossing; support returns and W reaches column 16 at 15. Both exits are reached at 78, but the recorded maintenance violation remains. All 79 states and 78 joint actions are available.

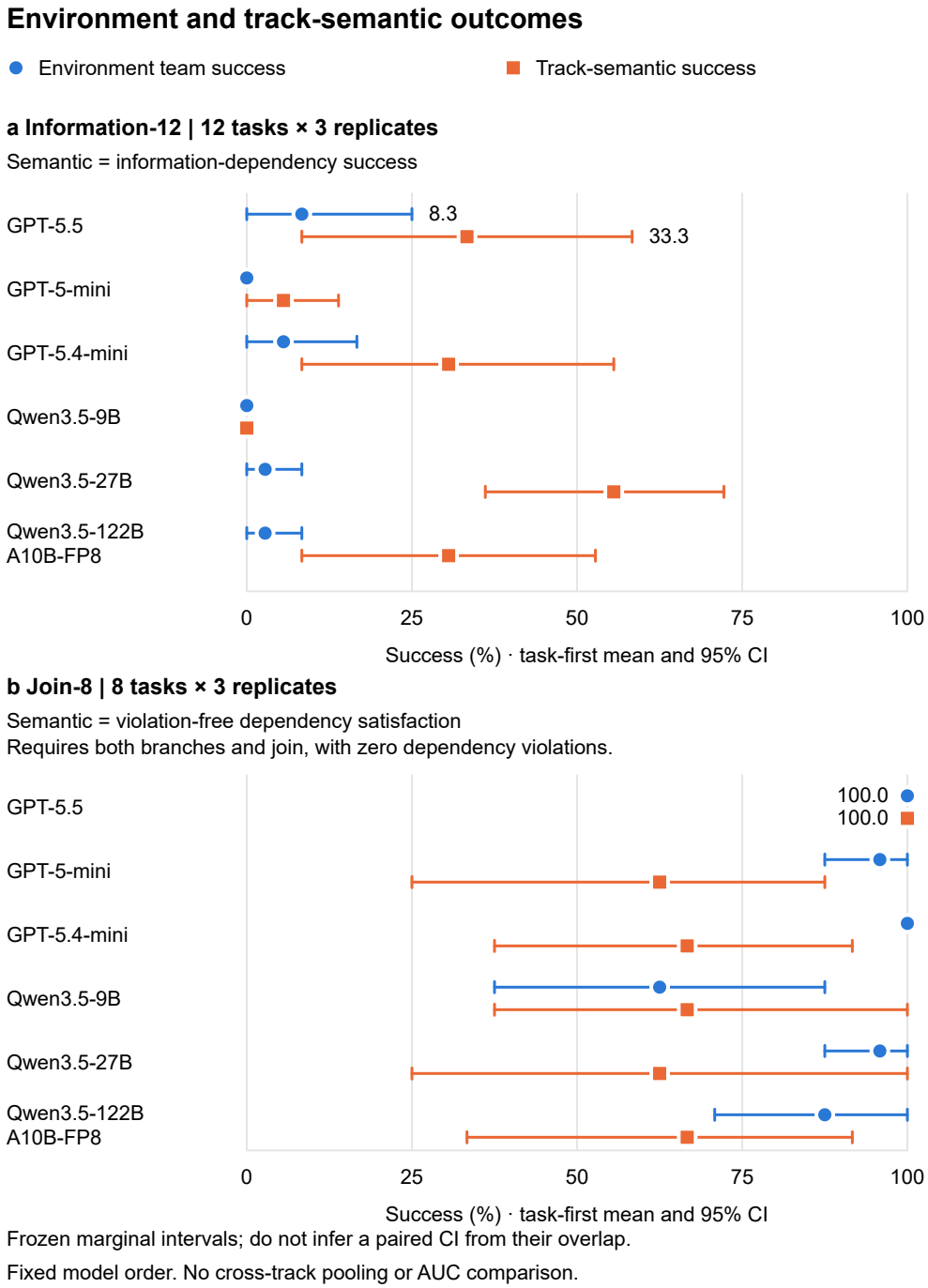
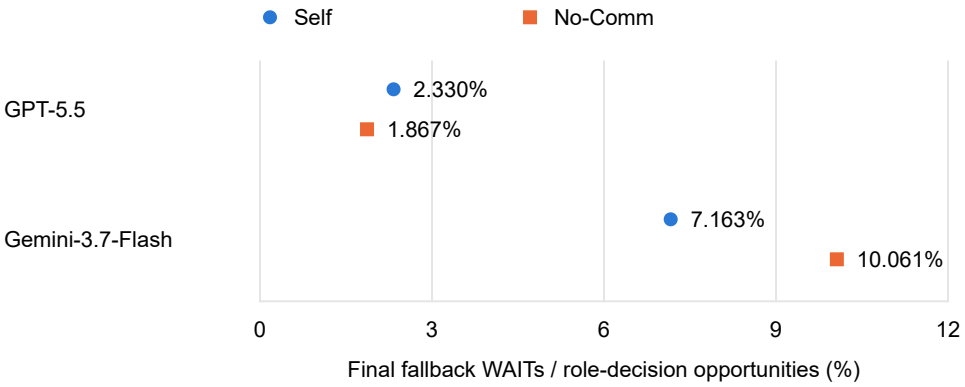


Figure 6: Environment versus track-semantic success. Join semantics require both branches, the join, and zero dependency violations. Whiskers are frozen marginal task-bootstrap intervals, not paired-difference intervals; model order is not a ranking.

Two distinct reliability diagnostics

a Phase II: final fallback WAITs per role decision



GPT: Self 241/10,344; No-Comm 192/10,286

Gemini: Self 767/10,708; No-Comm 1,090/10,834

Observed failures; interface/communication confounding remains.

b Standard-52: blocked parsed non-WAIT actions

Percent of all role-decision opportunities; not invalid outputs.

	Core-24	Info-12	Sync-v2-8	Join-8
GPT-5.5	10.73	8.52	0.11	0.00
GPT-5-mini	16.87	9.35	0.18	0.00
GPT-5.4-mini	18.15	4.78	0.22	0.11
Qwen3.5-9B	39.62	39.91	12.29	25.34
Qwen3.5-27B	33.49	34.23	15.97	0.79
Qwen3.5-122B A10B-FP8	32.38	23.88	16.96	10.98

Blocking scale (%): 0 | 10 | 20 | 30 | 40

Separate fallback counts: 7 / 191,966 = 0.00365%

GPT-5.5	0/28,438	Qwen3.5-9B	0/35,658
GPT-5-mini	4/31,504	Qwen3.5-27B	0/32,772
GPT-5.4-mini	3/30,498	Qwen3.5-122B A10B-FP8	0/33,096

Denominator: role decisions (two per round), not environment rounds.

Recovered retries are not final fallbacks. Descriptive rates; no CIs.

Figure 7: Separate operational diagnostics: Phase II final fallback WAITs and Standard-52 blocked parsed moves, each per role-decision opportunity. Blocking is not schema failure. Rare fallbacks are listed separately; interface/communication confounding remains.

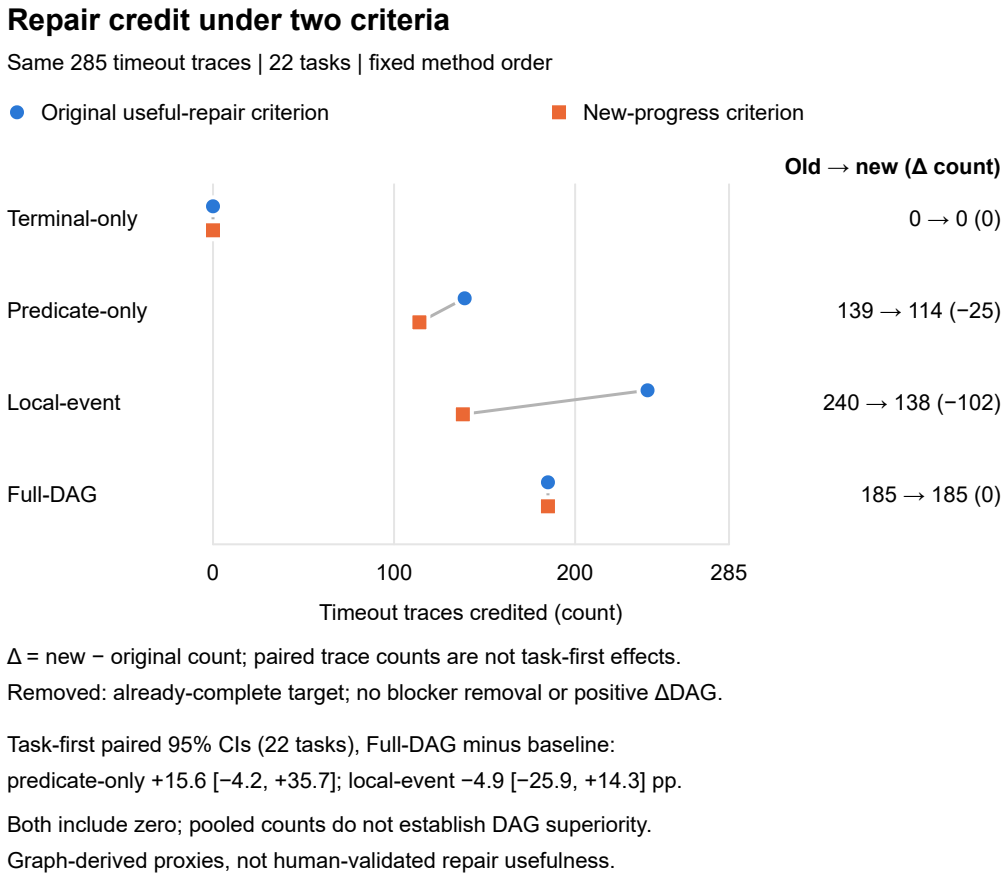


Figure 8: Original and new-progress repair credit on the same 285 timeouts. Counts are paired; reported paired intervals use 22 task means, not pooled-rate subtraction. Both Full-DAG contrast intervals include zero; no general repair superiority is established.

J Ethics Declaration

The benchmark uses synthetic symbolic environments and involves no human participants or personal data. Evaluation uses commercial model APIs, so cost, availability, and provider policies may affect reproducibility; provider interface failures are reported rather than hidden.