Supplementary Material

Beyond Division of Labor: How Roles Shape Feedback and Recovery in Multi-Agent Systems
Preprint

Chih-Hsuan Yang*†
Argonne National Laboratory
United States
bellayang@anl.gov

Xinyang Li*
University of Houston
United States

Rajeev Thakur
Argonne National Laboratory
United States

Feng Yan
University of Houston
United States

Project Website | Code | Dataset

What this document is

The main paper states every claim it makes and is complete without this file. What it cannot carry in eight pages is the full numeric backing behind some of its sentences, and that backing is here, in the order of the main paper's results: the realized organization (R1), recovery (R2), the submission gate (R3), and the second domain and structural enforcement (R4), followed by the pre-registered endpoints and the single-agent comparator. Every table is generated by a script from the manifested data (the frozen snapshot, or the separately manifested post-freeze runs) with the same scripts and outputs as the main paper's tables, except the single-agent comparator, which comes from an earlier, separately collected corpus. The four figures are generated from the same outputs.

Terms used throughout. A *trajectory* is one problem's run through the pipeline, at most six *cycles* of Planner, Executor and Reviewer. A cycle ends in a *submission* to the evaluator when the third agent approves the current answer, and in a *decline* otherwise; each submission returns a *verdict*, PASS or FAIL, and on a FAIL a repair hint. *First pass* is whether the first submitted answer is correct; a trajectory that never submits is graded once when its budget runs out, and that grade is its first pass. *Recovery* is the share of *first-pass-wrong* trajectories that finish correct. Recovery is reported on two populations that must not be read across: the *marginal* population is every first-pass-wrong trajectory of a condition, on that condition's own denominator; the *paired* population is only the problems first-pass wrong under both conditions being compared, on one shared denominator, and it is the population every test of recovery uses. Tests are exact McNemar tests on the problems that exactly one condition gets right; intervals are 95% bootstrap intervals that resample problems; where several tests form a family, p values are corrected with Holm's step-down procedure. Two *scorings* recur: the *pre-registered* scoring counts a trajectory that never produced a graded answer as incorrect, and the *as-graded* scoring drops it. The conditions are abbreviated RS (role-structured: Planner, Executor and Reviewer prompts), RF (role-free: one generic prompt for all three agents, with the evaluator's feedback shown), RF$^-$ (role-free with the feedback withheld) and SE (structurally enforced: the same roles imposed by typed message formats). The actors are the gemma-4-31B-it, gpt-oss-120b and Llama-3.3-70B-Instruct models, each playing all three agents.

*These two authors contributed equally to this work.
†Corresponding author: Chih-Hsuan Yang (bellayang@anl.gov).

Table 1: Functional collapse on the Omni-MATH benchmark under three measures, with the structurally enforced condition and a sensitivity analysis over answer markers.

actor	functional collapse	exact byte duplication	adjudicated no division
role-structured (RS)			
gemma-4-31B-it	0.000	0.000	0.000
gpt-oss-120b	0.120	0.000	0.000
Llama-3.3-70B-Instruct	0.000	0.000	0.000
role-free, feedback withheld (RF$^-$)			
gemma-4-31B-it	**0.976**	0.000	**0.857**
gpt-oss-120b	**0.846**	0.680†	**1.000**
Llama-3.3-70B-Instruct	**0.872**	0.008	**1.000**
role-free, with feedback (RF)			
gemma-4-31B-it	**0.974**	–	–
gpt-oss-120b	**0.883**	–	–
Llama-3.3-70B-Instruct	**0.900**	–	–
structurally enforced (SE)			
all actors	n/a	$\leq$ 0.001	0.071–0.188

RS: role-structured prompts; RF$^-$ and RF: role-free prompts with the evaluator's feedback withheld or shown (RF was run after the freeze); SE: structural enforcement. Functional collapse: share of first cycles in which all three agents assert their own final answer. Exact byte duplication: share of first cycles in which two or more of the three messages are identical character for character, a lower bound on collapse that is never used to compare actors. Adjudicated no division: share of a judged sample of first cycles in which a model judge with a fixed rubric finds no division of labor; the judge did not see the grade but could infer the condition. Collapse is not applicable under SE, whose Planner schema has no answer field; for RF only functional collapse was measured. *Sensitivity.* Under three answer-marker definitions the feedback-withheld role-free rate is 0.846–0.999; the role-structured rate is 0.000–0.238 under the two stricter markers and 0.328–0.892 under the loosest, which also counts any "therefore", "hence" or "=number" (as when a Planner restates givens), so read the role-structured level as "low," not a point estimate. The ordering holds in all 9 of 9 actor×marker cells.
†Not interpretable: 80% of this cell's planner messages are under 500 visible characters (the main paper's logging defect).

1 The realized organization (R1)

The main paper reports that deleting the role clauses makes every agent assert its own final answer. Table 1 is the full form of its Table 1, with two further measures and the enforced condition, and Figure 1 shows which agent changes.

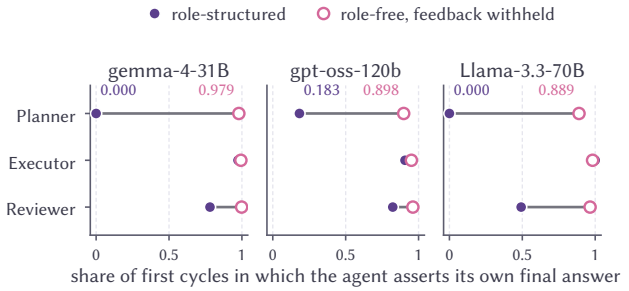

Figure 1: Which agent changes when the role clauses are deleted. For each actor, the share of first cycles in which the Planner, Executor and Reviewer assert their own final answer under role-structured prompts (filled) and under role-free prompts with the evaluator's feedback withheld (hollow), on the Omni-MATH benchmark. The Planner moves from almost never answering to almost always answering; the Executor answers under both; the Reviewer moves less.

The Planner is the agent that changes. By role, the share of first cycles in which an agent asserts its own final answer moves from RS to RF$^-$ as follows, for gemma-4-31B-it, gpt-oss-120b and Llama-3.3-70B-Instruct in turn: Planner 0.000, 0.183 and 0.000 to 0.979, 0.898 and 0.889; Executor 0.973, 0.907 and 1.000 to 0.993, 0.951 and 0.982; Reviewer 0.781, 0.823 and 0.492 to 0.998, 0.961 and 0.965. The Executor's job is to produce the answer, so it answers under both conditions. The Reviewer often states an answer even with roles, plausibly because approving an answer tends to restate it. The division of labor therefore rests on the Planner's single prohibition, and the role-free footer, which asks every agent for its current answer, lifts it.

Without roles, Planner and Executor compute the same quantities. The numbers in a Planner's first-cycle message and in the Executor's overlap more than Planner and Executor messages from different problems in the same condition do, by +0.449, +0.680 and +0.432 without roles against +0.159, +0.167 and +0.145 with them (Jaccard similarity of the two number sets). Under roles the Planner writes a plan and the Executor computes; without them both compute the same quantities. The gpt-oss-120b value without roles is not interpretable, because most of those Planner messages are only a few characters long (the logging defect below).

A second opinion from a model judge. A frontier model rated the first cycles of 171 sampled trajectories (42 RS, 56 RF$^-$ and the rest SE) against a rubric frozen before the first judgment, asking whether a division of labor was realized. Its verdicts agree with the functional-collapse measure (Table 1), and across the three rubric questions checked for stability its repeated judgments agree on 0.947 to 0.959 of the items. The judge did not see the grade, but it could infer the condition, because role words were renamed rather than equalized across conditions; the collapse claim therefore does not rest on it.

The logging defect, in detail. A message whose hidden reasoning was discarded is short but slow, so we mark messages with fewer than 10 visible characters per second of latency. They are 75.7% of gpt-oss-120b's role-free messages and 7.7% of its role-structured ones, and 93.4% and 22.7% of its first cycles contain at least one. The runner's own fallback flag marks only 5.8% and 2.7% of the same messages, so the flag undercounts the defect. For the other two actors at most 0.5% of messages are marked. The defect cannot have produced the collapse: on first cycles with no marked message, gpt-oss-120b's functional collapse is 0.821 (n = 106) without roles and 0.118 (n = 1,236) with roles.

Validating the computation-abstention measure. P1a, the pre-registered measure of whether a Planner abstains from computing, was checked by hand on 72 Planner messages against a definition frozen before the adjudication sheet was opened, each judgment with a recorded reason. It agrees with the hand labels on 0.903 of messages (Cohen's $\kappa = 0.806$; precision 0.943, recall 0.868), clearing the pre-set 0.80 threshold. Agreement fell from 0.967 at $n = 30$ as adjudication continued, and we report the final figure. Its residual errors point in opposite directions (symbolic results with no digits are missed, restated givens are flagged), so they do not push the measure one way.

2 Recovery (R2)

The main paper reports that, with the same access to feedback, the role-free team recovers more, and that against the feedback-withheld diagnostic the roles would look load-bearing (the main paper's R3). Table 2 gives that frozen feedback-withheld comparison on both populations, so the marginal rates and the paired test can be checked against each other; the comparison with feedback is the main paper's Table 2.

Where the diagnostic gap comes from. For gemma-4-31B-it the accounting closes exactly. The two conditions reach a correct first answer on the same number of problems (1,284 of 1,600), and a correct first answer is never lost because the loop halts on a PASS, so the gap in final accuracy, 54 problems in favor of roles ($p = 1.6 \times 10^{-5}$), is exactly the gap in recoveries, 70 against 16. Most of it arises on the 237 problems both teams first miss, where 38 are recovered with roles against 4 without; the problems that only one team first misses contribute a further 20 in the same direction. This accounting needs equal first-pass counts and does not transfer to the other two actors.

The result reproduces. The two gemma-4-31B-it cells of the RF condition were run twice, independently and about thirteen hours apart, with byte-identical runner and prompt code; every other RF cell was run once. Table 3 gives both runs against the frozen RS cell. The ordering and the final-accuracy contrasts reproduce in both domains; the SciBench paired recovery test, on 100 problems, is $p = 0.013$ in the first run and $p = 0.052$ in the second, so on that benchmark the recovery contrast is at the edge of resolution. Each run also repeated the drift check of the main text on the same 400 frozen problems: the RS re-run reaches final accuracy 0.818 in the first and 0.845 in the second, against 0.825 frozen (paired $p = 0.75$ and $p = 0.20$). The main text reports the first run throughout.

Recoveries are revisions, and the hints do not leak the answer. Of the 193 recoveries of the gemma-4-31B-it RF team on Omni-MATH,

Table 2: First-pass accuracy and recovery with roles and without them (feedback withheld) on the Omni-MATH benchmark, on both populations.

Marginal population: first pass paired; recovery on each condition's own denominator						
	first pass			recovery | first verdict wrong		
actor	RS	RF$^-$	difference	RS (n)	RF$^-$ (n)	difference
gemma-4-31B-it	0.802	0.802	+0.0000 [−0.0156, +0.0150]	0.222 (316)	0.051 (316)	−0.1709 [−0.2247, −0.1171]
gpt-oss-120b	0.713	0.763	+0.0503 [+0.0339, +0.0666]	0.324 (463)	0.247 (380)	−0.0766 [−0.1361, −0.0153]
Llama-3.3-70B-Instruct	0.308	0.316	+0.0077 [−0.0283, +0.0437]	0.014 (276)	0.015 (268)	+0.0004 [−0.0214, +0.0192]
Paired population: problems first-pass wrong under both conditions; one shared denominator						
actor	n_{both}	RS	RF$^-$	RS only	RF$^-$ only	exact p
gemma-4-31B-it	237	0.160	0.017	37	3	1.95×10^{-8}
gpt-oss-120b	327	0.242	0.202	52	39	0.208
Llama-3.3-70B-Instruct	245	0.004	0.016	1	4	0.375

RS: role-structured; RF$^-$: role-free with the evaluator's feedback withheld. First pass: whether a trajectory's first submitted answer is correct. Recovery: the share of first-pass-wrong trajectories that finish correct. Upper block: first pass over the problems graded under both conditions; recovery over each condition's own first-pass-wrong trajectories (n in parentheses, not a shared denominator); differences are RF$^-$ minus RS, with 95% bootstrap intervals over problems. Lower block: only the problems first-pass wrong under both conditions (n_{both}), which is the actual test; "RS only" and "RF$^-$ only" count problems recovered under one condition and not the other, with the exact McNemar p for the two counts. The two blocks have different denominators and must not be read across. Recovery conditions on a post-treatment variable and is descriptive rather than causal; no cell recovers before its first evaluator verdict.

Table 3: The two gemma-4-31B-it cells of the role-free condition with feedback, run twice, against the frozen role-structured cell.

	first pass	recovery	paired recovery	final	paired final
Omni-MATH					
role-structured (frozen)	0.802	70/316 = 0.222	–	0.846	–
role-free, with feedback, run 1	0.802	193/317 = 0.609	89 vs 4, $p = 6.2 \times 10^{-22}$	0.921	137 vs 17, $p = 1.7 \times 10^{-24}$
role-free, with feedback, run 2	0.811	177/302 = 0.586	95 vs 7, $p = 7.8 \times 10^{-21}$	0.922	139 vs 18, $p = 2.4 \times 10^{-24}$
SciBench					
role-structured (frozen)	0.794	30/118 = 0.254	–	0.847	–
role-free, with feedback, run 1	0.796	49/117 = 0.419	21 vs 7, $p = 0.013$	0.880	31 vs 12, $p = 0.0054$
role-free, with feedback, run 2	0.784	54/124 = 0.435	19 vs 8, $p = 0.052$	0.878	29 vs 11, $p = 0.0064$

Both runs use byte-identical runner and prompt code; the main text reports run 1. First pass: whether a trajectory's first submitted answer is correct. Recovery: first-pass-wrong trajectories that finish correct, over each condition's own first-pass-wrong set (count / denominator). Final: accuracy at the end over all problems, never-graded trajectories counted as incorrect. The paired columns compare each role-free run with the role-structured run on the problems first-pass wrong under both (recovery) or on all problems (final): problems recovered, or finally correct, only without roles versus only with them, with the exact McNemar p for the two counts.

192 pass with an answer different from the one first rejected (541 of 547 across the five RF cells), so re-grading the same answer explains almost none of them; identical evaluator prompts at temperature 0 return different verdicts in only 20 of 2,790 prompt groups (0.0072). The evaluator writes each hint with the gold answer in view, so a hint could hand the answer over. Hints contain it verbatim in at most 2.3% of failing verdicts in any cell, and dropping every gemma-4-31B-it Omni-MATH trajectory that saw such a hint leaves recovery at 0.599 for RF and 0.212 for RS, against 0.609 and 0.222 on all trajectories. We did not audit partial leakage.

3 The submission gate (R3)

The main paper's account of what roles do after feedback is that the role-structured Reviewer gates submission: the RS team asks for fewer verdicts, and matched on verdicts no significant difference remains. This section gives the evidence in three steps: where the declines happen (Table 4), how recovery tracks exposure to the evaluator (Figure 2), and how far the verdict-matched comparison can be pushed (Table 5, Figure 3a). The forced-submission run (Table 6) then removes the gate directly.

Where the Reviewer declines. On Omni-MATH the clause that forbids approval after a FAIL covers only a minority of the RS Reviewer's declines, between 4% and 37% depending on the actor, and in the free cycles, where no clause applies, it approves far less often than the RF team's third agent (Table 4). The gate is therefore mostly the Reviewer's disposition rather than the letter of its prompt; gpt-oss-120b's Reviewer is the most permissive of the three. On SciBench the clause covers more of the declines, and the RS team almost always submits at least once, which is why the recovery contrast is weaker there (Section 4).

Recovery tracks exposure. Figure 2 shows the mechanism directly. In every cell the RS team's share of trajectories still receiving verdicts falls steeply after the first one, while the RF team's falls slowly, and the cumulative recovery curves follow the exposure curves. Per

Table 4: Where the role-structured Reviewer declines, per cell, and what each team's allocation of the budget buys.

	RS declining cycles			clause	RS never	free-cycle approval		by 2nd verdict		per later verdict		approvals	
actor, domain	before	FAIL	free	share	submits	RS	RF	RS	RF	RS	RF	RS	RF
gemma-4-31B-it, Omni-MATH	2.24	0.81	1.20	0.19	84/316	0.25	0.96	0.206	0.438	0.41	0.28	1.27	3.20
gpt-oss-120b, Omni-MATH	1.33	1.17	0.71	0.37	51/463	0.53	0.68	0.246	0.296	0.27	0.22	2.10	3.53
Llama-3.3-70B-Instruct, Omni-MATH	4.49	0.25	0.88	0.04	205/276	0.01	0.82	0.014	0.092	0.25	0.08	0.32	2.71
gemma-4-31B-it, SciBench	0.55	1.38	1.48	0.40	3/118	0.35	0.99	0.237	0.239	0.26	0.16	1.95	3.62
gpt-oss-120b, SciBench	0.41	1.53	0.60	0.60	1/124	0.67	0.78	0.258	0.317	0.19	0.16	2.68	3.95

RS: role-structured; RF: role-free with the evaluator's feedback shown. Every column is over first-pass-wrong trajectories. A cycle is one pass of the three agents; it ends in a submission to the evaluator when the third agent approves and in a decline otherwise. RS declining cycles: per trajectory, split into those before the first submission (no feedback yet), those in a cycle that received a FAIL verdict (where the RS clause forbids approval) and the remaining "free" cycles; clause share is the FAIL column as a fraction of all three. RS never submits: trajectories with no approval in six cycles, graded once at the end. Free-cycle approval: the share of free cycles ending in approval. By 2nd verdict: the share recovered with at most one round of feedback. Per later verdict: recoveries per verdict after the first, a ratio over different trajectories in the two conditions (the verdict-matched comparison of the main paper's Table 2 is the like-for-like one). Approvals: submissions per trajectory.

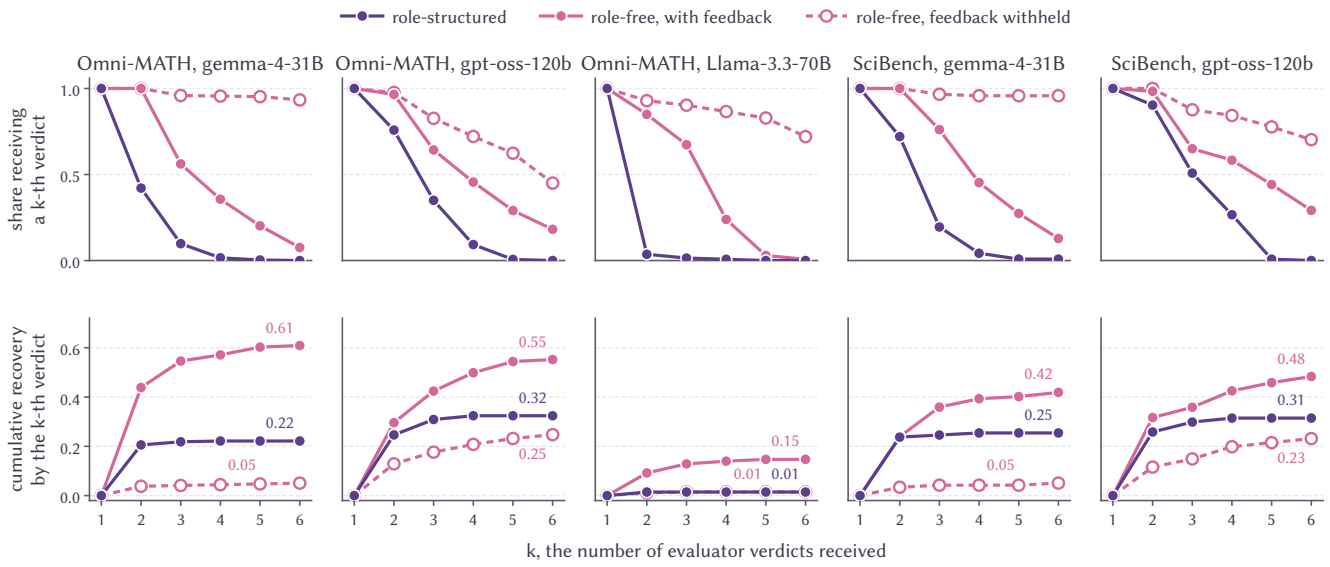

Figure 2: Recovery follows exposure to the evaluator. Top: the share of first-pass-wrong trajectories that receive a k-th evaluator verdict. Bottom: the cumulative share recovered by the k-th verdict, with the final value labeled. One panel per cell (actor on benchmark): role-structured (purple), role-free with feedback (rose) and role-free with the feedback withheld (hollow, dashed). The role-structured curve flattens where its trajectories stop receiving verdicts, not where they stop using them.

verdict received, neither team is consistently ahead: on the paired problems, pooled over the five cells, the share whose second verdict is a PASS is 0.267 for RS and 0.224 for RF, and at the third and fourth verdicts 0.083 against 0.127 and 0.105 against 0.089. Recoveries per verdict after the first are higher with roles in every cell (Table 4), but that ratio compares different trajectories, since most of the RF team's later verdicts are further attempts on problems it has already failed more than once; the like-for-like comparison is the verdict-matched one.

How far the verdict-matched comparison can be pushed. Matching on verdicts is coarse. When the RS team received a single verdict, because it never submitted and was graded at the end or because it submitted once and stopped, the RF trajectory cut to one verdict cannot recover either, and the pair ties at zero by construction. These forced ties are most of the gemma-4-31B-it pairs and nearly all of the Llama-3.3-70B-Instruct pairs (Table 5), so the pooled interval of the main paper is narrower than the information behind it. The cut is also one-sided, since the RS count stops at that team's own passing verdict, which can only favor it. Table 5 and Figure 3a therefore give the contrast under six definitions: the problems with at least two RS verdicts, both trajectories cut at the smaller count, and the scoring defect recoded, each with and without Llama-3.3-70B-Instruct. No variant puts the RF team significantly ahead: the two symmetric cuts favor it by 0.005 and 0.010 with intervals that include zero, the informative problems allow up to +0.065 in favor of roles (+0.073 with the recoding), and with the recoding on all problems the RS team leads by +0.029 with an interval that excludes zero. The defensible reading is the main paper's: matched on verdicts the RF advantage disappears, and any per-verdict advantage

of roles is small. Matching equalizes exposure to the evaluator, not calls; the RS team spends more cycles per verdict.

Across the freeze. The frozen RS cell and the RF team were run on opposite sides of the freeze, so we repeat the comparison against the post-freeze RS re-run of gemma-4-31B-it on 400 Omni-MATH problems. On the 65 problems both first answer wrongly, the RF team leads 22 to 3 unmatched ($p = 1.6 \times 10^{-4}$), but the RS team leads 4 to 1 verdict-matched ($p = 0.38$): the same pattern as in the frozen comparison.

Removing the gate alone. The forced-submission run tests the gate directly. It uses the RS prompts and runner unchanged except for one step: after the Reviewer's turn, the pipeline submits the team's current answer to the evaluator whatever route the Reviewer chose, and the verdict and hint reach the agents through the same role-specific slots as in the RS condition. An offline test in the supplementary material (no model calls) checks that the prompts are byte-identical to the RS ones up to the first verdict and that the evaluator is called in every cycle. It was run once, after the freeze, for gemma-4-31B-it on the 400 Omni-MATH problems of the RS re-run, with the same server, budget and evaluator. Table 6 compares it with that re-run, the frozen RS cell and the RF team, restricted to the same problems. Recovery rises to the RF team's level while first-pass accuracy does not move: once the Reviewer's approvals no longer decide anything, the role prompts recover as much as the generic ones.

4 The second domain (R4)

Table 7 gives the SciBench feedback-withheld comparison in full: the frozen RS and RF^{-} cells with their denominators, for the two actors run there. The main comparison, with feedback, is in the main paper's Figure 2 and Table 2.

What was and was not changed. The prompts are not byte-identical to Omni-MATH's, and the prompt files in the supplementary material make the difference checkable. Of the 16 prompts the two domains share, 11 are identical and 5 differ: four of the six RS prompts (the Planner, Executor and Reviewer preambles and the Reviewer's routing rules) and the evaluator's preamble, which must ask for scientific rather than mathematical equivalence. Both RF prompts are identical across domains, as are the six SE prompts (enforcement was not run on SciBench). The asymmetry deserves stating plainly: the RS prompts, whose team recovers less than the RF team, are the ones that were rewritten, while the RF prompts ran identical text in both domains. The rewrite is domain wording ("algebraic manipulation or numerical calculation" where Omni-MATH said "symbolic manipulation or arithmetic"), and the structural clauses are unchanged; a reader who suspects the adaptation is doing the work can check the files. Topology, budget, decoding, retry policy and the evaluator protocol are unchanged, so this is a replication of the mechanism under a matched protocol, not a rerun of identical prompts.

Why the answer does not lock here. On Omni-MATH the gemma-4-31B-it RS team often keeps the same answer across cycles (a single distinct answer in 28.8% of its failed trajectories); on SciBench it does not (7.6%, against 8.5% for the RF team). The RS team on SciBench also almost always submits at least once (Table 4), so it sees more feedback, and by the second verdict the two teams have recovered equally often. A looser gate is the simplest common explanation of both observations, though we did not test it.

5 Structural enforcement (R4)

The main paper reports that enforcement loosens the gate and raises recovery, but that attrition keeps the gain from reaching end-to-end accuracy, and that the apparent accuracy benefit under the as-graded scoring is survivorship. Table 8 and Figure 4 are the evidence for both halves: the manipulation check, the one-sided attrition, and the same accuracy difference under both scorings.

Why the columns do not compose. The enforced recovery and first-pass rates do not combine into the accuracy column, and cannot: trajectories that never receive a first verdict have no first pass, and those that die after one stay in the recovery denominator, so the three quantities do not share a denominator. The pre-registration fixed in advance that unparseable trajectories count as incorrect rather than being dropped, and that rule is what makes the as-graded reading survivorship rather than a finding.

The gate loosens, and brittleness follows the same order. Per failed trajectory, the SE Reviewer approves 2.02, 2.33 and 1.84 times for gemma-4-31B-it, gpt-oss-120b and Llama-3.3-70B-Instruct, against 1.27, 2.10 and 0.32 under RS prompts. Enforcement logs 838, 1,183 and 395 role-boundary rejections, that is 0.52, 0.74 and 0.99 per trajectory, in the same order as the attrition (8.0%, 9.9% and 17.0%): the actor that breaks the schema most often is the one that loses the most trajectories. For Llama-3.3-70B-Instruct, whose RS team rarely submits, enforced recovery comes close to the RF team's (0.139 against 0.147); on the 219 problems both first answer wrongly, 9 are recovered only under enforcement and 17 only by the RF team ($p = 0.17$).

The context-restored control. Enforcement also hides the shared history from the Executor, so part of its brittleness may be lost context rather than the typed interface. Restoring the shared history at fixed enforcement (gemma-4-31B-it and gpt-oss-120b, 400 problems each; attrition 4.5% and 9.8%) changes final accuracy relative to enforcement by +0.0450 [+0.0125, +0.0775] ($p = 0.0079$) for gemma-4-31B-it and by −0.0125 [−0.0475, +0.0200] ($p = 0.56$) for gpt-oss-120b under the pre-registered scoring, and by +0.0196 [−0.0028, +0.0419] ($p = 0.14$) and −0.0089 [−0.0357, +0.0149] ($p = 0.65$) as graded. For gemma-4-31B-it the gain is significant only under the scoring that penalizes attrition, so it comes mostly from fewer trajectories dying; the control is partial, because the attrition it removes is one-sided.

An early campaign stopped for a truncation confound. The main run was gated on a smoke test requiring non-zero boundary rejections with reasons logged, answer-bearing calls equal across conditions, and a pinned backend. An early campaign was stopped after 121 trajectories when 74% of enforcement rejections proved to be the model cut off at the token limit mid-JSON rather than violating a role boundary. Truncation was not condition-specific but its cost was: a prose condition tolerates a truncated message where the enforced one rejects unparseable JSON, so left alone it

Table 5: Robustness of the verdict-matched recovery contrast: forced ties, six definitions of the contrast, and per-verdict pass rates.

cell	paired n	forced ties	RS never submitted	problems left	held-answer recodes (RS / RF)
gemma-4-31B-it, Omni-MATH	236	141	66	95	12 / 3
gpt-oss-120b, Omni-MATH	319	84	36	235	4 / 0
Llama-3.3-70B-Instruct, Omni-MATH	250	242	187	8	2 / 1
gemma-4-31B-it, SciBench	100	29	2	71	2 / 0
gpt-oss-120b, SciBench	93	11	1	82	0 / 0

	all five cells			without Llama-3.3	
definition	n	difference [95% interval]	equivalent	n	difference [95% interval]
all paired problems (as in the main paper's Table 2)	998	+0.009 [−0.014, +0.033]	yes	748	+0.011 [−0.020, +0.043]
problems with two or more role-structured verdicts	491	+0.018 [−0.029, +0.065]	no	483	+0.017 [−0.031, +0.064]
both cut at the smaller verdict count	998	−0.005 [−0.029, +0.019]	yes	748	−0.008 [−0.040, +0.024]
both cut, two or more verdicts	491	−0.010 [−0.059, +0.039]	no	483	−0.012 [−0.062, +0.037]
scoring defect recoded	998	+0.029 [+0.004, +0.054]	no	748	+0.035 [+0.003, +0.067]
recoded, two or more verdicts	491	+0.026 [−0.020, +0.073]	no	483	+0.025 [−0.023, +0.072]

per-verdict pass rate	role-structured	role-free with feedback
verdict 2	131/491 = 0.267	212/946 = 0.224
verdict 3	19/228 = 0.083	89/700 = 0.127
verdict 4	8/76 = 0.105	39/439 = 0.089

RS: role-structured; RF: role-free with the evaluator's feedback shown; a cell is one actor on one benchmark. The contrast takes the problems both teams first answer wrongly and cuts the RF trajectory after as many evaluator verdicts as the RS trajectory received on that problem. Where RS received a single verdict, the cut RF trajectory cannot recover either, so the pair ties at zero by construction. Upper block: such forced ties per cell, how many of them never submitted, the problems left, and the trajectories whose held answer matched the key but were graded wrong (the scoring defect of the main paper's Limitations). Middle block: the pooled difference, RS minus RF, stratified by cell with 95% bootstrap intervals over problems, under six definitions, with and without Llama-3.3-70B-Instruct; equivalent means the 90% interval lies inside ±0.05. Recoding counts an RS trajectory with a correct held answer as recovered, and an RF one only if it was not cut. Lower block: the share of trajectories whose k-th verdict is PASS among those whose earlier verdicts all failed, pooled over the five cells.

Table 6: Removing the submission gate: gemma-4-31B-it on the same 400 Omni-MATH problems in every row.

condition	first pass	recovery	final accuracy	verdicts	Reviewer approvals
role-structured, every cycle submitted	0.768	57/93 = 0.613	0.910	4.20	0.98
role-structured, re-run after the freeze	0.773	18/91 = 0.198	0.818	1.44	1.13
role-structured, frozen snapshot	0.795	12/82 = 0.146	0.825	1.41	1.15
role-free with feedback	0.795	45/82 = 0.549	0.908	3.34	3.34

		recovery (both first wrong)		final accuracy	
pair	n	first / second only	p	first / second only	p
every cycle submitted vs. re-run	72	26 / 2	3.0×10^{-6}	42 / 5	2.5×10^{-8}
every cycle submitted vs. role-free with feedback	70	7 / 7	1	8 / 7	1
re-run vs. role-free with feedback	65	3 / 22	1.6×10^{-4}	6 / 42	1.0×10^{-7}

Role-structured: Planner, Executor and Reviewer prompts, in the frozen snapshot and in a re-run after the freeze; role-free with feedback: one generic prompt for all three agents, with the evaluator's feedback shown. The forced-submission run ("every cycle submitted") uses the role-structured prompts byte for byte, but the pipeline submits the team's answer in every cycle whatever the Reviewer decides; its approvals are logged but do not decide submission. First pass: the share of trajectories whose first submitted answer is correct. Recovery: first-pass-wrong trajectories that finish correct, over each row's own first-pass-wrong set (count / denominator). Final accuracy: over all problems, never-graded trajectories counted as incorrect. Verdicts and Reviewer approvals: per first-pass-wrong trajectory, including the first. The lower block pairs two conditions on the problems both first answer wrongly (recovery) and on all problems (final accuracy): problems solved only by the first / only by the second, with the exact McNemar p for the two counts.

would have produced a convincing "enforcement lowers accuracy" artifact. The main run uses twice that token limit, and the early rows were never pooled.

Accumulated context as a competing explanation. The RS team's recovery advantage over the RF$^-$ team could in principle come from the longer history it accumulates over its internal cycles rather than from the feedback it sees. Table 10 stratifies recovery by accumulated context. The comparison is directional only: quartile cut points are computed over both conditions pooled, and accumulated context is nearly a function of condition, because the RF$^-$ team stops changing its answer, so the strata are not balanced and context cannot be separated from condition. It is reported because it does not behave uniformly. The gap survives in the lower quartiles, but

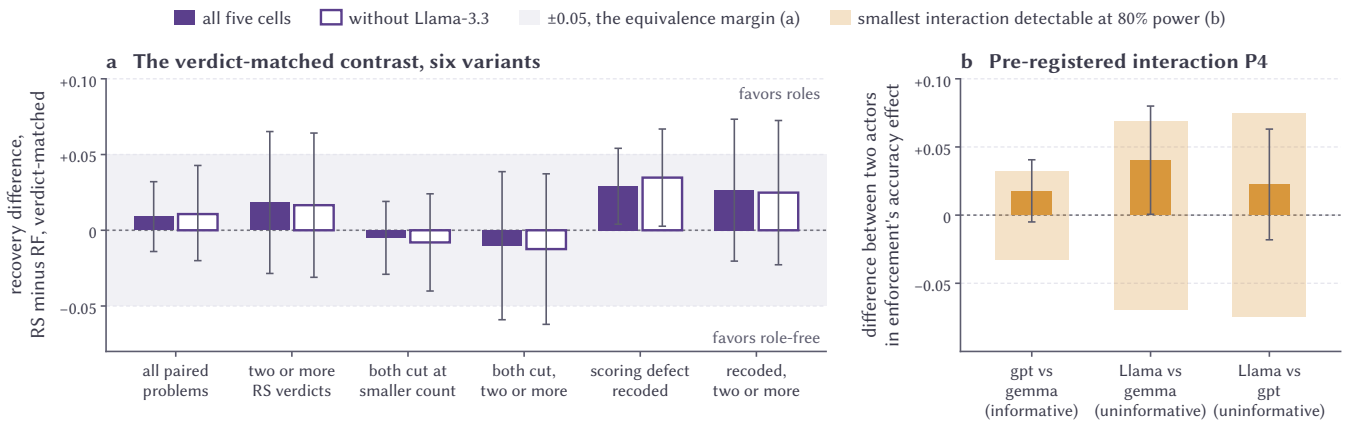

Figure 3: Two robustness checks. (a) The verdict-matched recovery difference, role-structured minus role-free with feedback, pooled over the five cells (filled) and without Llama-3.3-70B-Instruct (hollow), under the six definitions of Table 5, with 95% intervals; the band marks the ±0.05 equivalence margin. (b) The pre-registered interaction P4 (Section 6): each difference between two actors in enforcement's effect on final accuracy, with its 95% interval; the shaded bar behind it spans the smallest interaction detectable at 80% power. Where that bar is wider than any plausible effect, the null is uninformative.

Table 7: The SciBench benchmark: first pass and recovery with roles and without them (feedback withheld), two actors, frozen separately.

	first pass (paired, all problems)					recovery (paired on first-pass wrong under both)						
actor	n	RS	RF$^-$	diff.	exact p	n_{both}	RS	RF$^-$	diff.	RS only	RF$^-$ only	exact p
gemma-4-31B-it	574	0.794	0.794	+0.0000	1	99	0.212	0.020	−0.1919	21	2	6.60×10^{-5}
gpt-oss-120b	574	0.784	0.789	+0.0052	0.784	96	0.219	0.135	−0.0833	17	9	0.169

The complete 574-problem text-only slice of free-response college science, 2,296 trajectories. RS: role-structured; RF$^-$: role-free with the evaluator's feedback withheld; the contrast with feedback shown is in the main paper's Table 2. First pass is paired over all 574 problems, with the exact McNemar p for the problems right under only one condition. Recovery is paired over only the problems first-pass wrong under both conditions (n_{both}); "RS only" and "RF$^-$ only" count problems recovered under one condition and not the other. Differences are RF$^-$ minus RS under the pre-registered scoring (never-graded trajectories count as incorrect). gpt-oss-120b's recovery cell is underpowered at its n_{both}; its estimate points the same way as on Omni-MATH. Llama-3.3-70B-Instruct was not run in this domain.

in gpt-oss-120b's top quartile the ordering reverses, as it does on a handful of events in Llama-3.3-70B-Instruct's third, so more context is not simply more recovery.

6 The pre-registered endpoints in full

The main paper reports that the pre-registered primary endpoint P4, whether enforcement's effect on accuracy differs by actor, is null, and that the P1a manipulation objective failed. Table 9 is the complete endpoint table: all five endpoints as written before any intervention data existed, what each one returned, and the Holm-corrected verdict.

Two things in it should be read rather than skimmed. First, only one of the three P4 contrasts is an informative null (Figure 3b): the largest per-actor main effect is small enough that a true interaction is plausibly at most 0.040, only the gpt-oss-120b against gemma-4-31B-it contrast can detect an effect of that size, and the two contrasts involving Llama-3.3-70B-Instruct cannot detect an interaction of any plausible magnitude at their sample size. They are uninformative rather than evidence of absence. Second, the contrast pre-registered as secondary and exploratory was RS against RF$^-$; the contrast this paper is organized around, RS against RF, and the forced-submission run are outside the pre-registration; the RF condition's design and primary analysis were committed in writing before its data existed, but its results are exploratory, and the main paper labels them so.

7 The single-agent comparator

Table 11 is the single-agent reference under the same call cap. The main paper uses it as a guard against reading "collapse costs nothing" as "the system was one agent all along", and reports both directions. The result cuts both ways: a collapsed gpt-oss-120b team beats its own single agent, so three agents emitting one answer are not functionally one agent, while gemma-4-31B-it's team is significantly worse than its own single agent in both conditions. Neither "multi-agent systems are just single agents" nor "more agents help" survives it. The comparator comes from an earlier, separately collected corpus whose two actor arms were produced in separate runs, so each direction within an actor is a sound guard and the gap between actors is not.

Table 8: Structural enforcement against the role-structured baseline on the Omni-MATH benchmark: the manipulation check, attrition, and final accuracy under both scorings.

	Planner authors	locked answer	recovery		final accuracy, SE − RS	
actor	answer (RF^-)	(RF^- / RS)	(RS)	attrition	pre-registered	as graded
gemma-4-31B-it	0.000 (0.986)	0.119 (0.845/0.288)	0.314 (0.222)	8.0%	−0.0225 [-0.0387,-0.0063]	+0.0190 [+0.0054,+0.0333]
gpt-oss-120b	0.000 (0.916)	0.092 (0.208/0.052)	0.395 (0.324)	9.9%	−0.0050 [-0.0219,+0.0112]	+0.0361 [+0.0215,+0.0507]
Llama-3.3-70B-Instruct	0.000 (0.958)	0.072 (0.806/0.750)	0.139 (0.014)	17.0%	+0.0175 [-0.0200,+0.0550]	+0.0453 [+0.0030,+0.0876]

SE: structurally enforced; RS: role-structured; RF^-: role-free with the evaluator's feedback withheld. Planner authors answer: the share of Planner messages that state a final answer, under SE and (in parentheses) under RF^-; SE drives it to zero in all three actors, so the intervention engaged (the manipulation check). Locked answer: the share of first-pass-wrong trajectories whose answer string never changed, under SE and (in parentheses) RF^- / RS. Recovery: the share of first-pass-wrong trajectories that finish correct, under SE and (in parentheses) RS, on the denominator of Table 2. Attrition: the share of SE trajectories that never finish because a malformed output exhausted its regeneration budget. Final accuracy: the difference over all problems with 95% bootstrap intervals, under the pre-registered scoring (never-graded trajectories count as incorrect) and as graded (they are dropped); the two columns are different populations and the attrition column is why they disagree. Attrition elsewhere is negligible: 0.0–0.8% with roles, 0.0–2.5% without. The as-graded column turns every sign positive; the pre-registration fixed in advance that unparseable trajectories count incorrect, and **the benefit appears only after conditioning on successful parsing.**

Table 9: The five pre-registered endpoints and what each returned. P4, the primary, is null before and after Holm correction.

endpoint	disposition
P1 intervention → fidelity (also the manipulation check)	**failed as written; re-designated.** The interface forbids an *answer* field, not computation. Re-scoped to answer authorship, which it drives to 0.000 in all three actors (Table 8). Reported as a second dissociation, not as a pass.
P2 actor × interv. on fidelity	not estimable across the panel: the fidelity clauses are condition-specific, so no common scale exists. Reported as unresolved rather than substituted.

P3 intervention → accuracy (enforced − role-structured), Holm across three:

	raw p	Holm p	verdict
gemma-4-31B-it	0.0078	0.023	reject
gpt-oss-120b	0.60	0.85	retain
Llama-3.3-70B-Instruct	0.43	0.85	retain

P4 actor × intervention on accuracy (the pre-registered primary), Holm across three:

	difference-in-diff. [CI]	raw p	Holm p	$MDE_{80\%}$	null is
gpt-oss-120b − gemma-4-31B-it	+0.0175 [-0.0050,+0.0406]	0.145	0.29	0.0325	informative
Llama-3.3-70B-Instruct − gemma-4-31B-it	+0.0400 [+0.0006,+0.0800]	0.0553	0.166	0.0693	uninformative
Llama-3.3-70B-Instruct − gpt-oss-120b	+0.0225 [-0.0181,+0.0631]	0.286	0.29	0.0748	uninformative

P5 Δfidelity vs Δoutcome	association only, pre-declared as non-causal: both sides are outcomes of the same intervention. No p-value is reported and none is corrected.

The endpoints were fixed in writing before any intervention data existed. The intervention is structural enforcement against the role-structured baseline, per actor. Fidelity: whether the Planner abstains from computing (P1a). Accuracy: final accuracy over all problems, never-graded trajectories counted as incorrect. P3: exact McNemar test per actor, with the raw p, the p after Holm's step-down correction across the three actors, and the verdict at $\alpha = 0.05$. P4: each difference-in-differences is enforcement's effect in one actor minus its effect in another, with its 95% bootstrap interval over problems, the raw and Holm-corrected p, and $MDE_{80\%}$, the smallest interaction detectable with 80% power at the realized sample; a null is informative only where the MDE is below a plausible effect. P2 and P5 carry no p value, so the testable contrasts are P3's three and P4's three; Holm across all six changes no verdict. P4's intervals and p values come from two separately seeded bootstrap runs, so an interval can exclude zero while p is just above 0.05; the MDE is computed on problems shared by both actors while each difference-in-differences uses each actor's own problems, so neither can be derived from the other. **Only one of the three P4 nulls is informative.** The largest per-actor main effect is small enough that a true interaction is plausibly ≤0.040; only the gpt-oss/gemma-4 contrast can see an effect that size. The two Llama-3.3 contrasts cannot detect an interaction of any plausible magnitude and are uninformative rather than evidence of absence.

8 AI use statement

In this work, we used generative AI tools for three purposes: to edit the manuscript for readability and polish its wording; to search for and identify relevant literature and to format references; and to assist with data processing and analysis, including drafting and refining the code for these tasks and for the figures. All AI-generated text and code were reviewed by the authors, and LLM-generated code was verified and tested for correctness by the authors. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

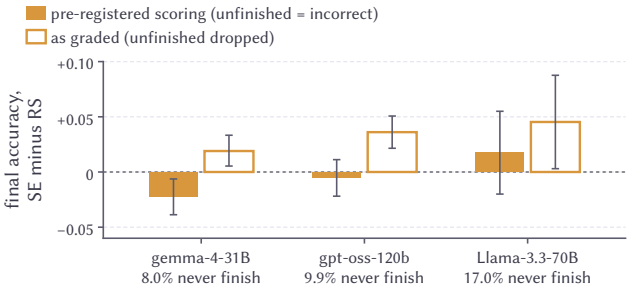

Figure 4: Enforcement's effect on final accuracy under the two scorings. Enforced minus role-structured final accuracy per actor with 95% intervals, under the pre-registered scoring (unfinished trajectories count as incorrect; filled) and as graded (they are dropped; hollow). The tick labels give the share of enforced trajectories that never finish, which is what separates the two bars. The sign flips between the scorings for two of the three actors, and the as-graded gains are ordered like the attrition.

Table 10: Recovery by accumulated-context quartile, with roles and without them (feedback withheld), on the Omni-MATH benchmark; directional only.

	RS		RF⁻		
context quartile	recovery	n	recovery	n	difference
gemma-4-31B-it					
Q1 (least context)	0.662	65	0.151	93	+0.511
Q2	0.280	50	0.009	108	+0.271
Q3	0.047	85	0.014	73	+0.033
Q4 (most)	0.078	116	0.000	42	+0.078
gpt-oss-120b					
Q1 (least context)	0.923	13	0.258	198	+0.665
Q2	0.656	128	0.293	82	+0.363
Q3	0.262	164	0.191	47	+0.071
Q4 (most)	0.070	158	0.189	53	−0.119
Llama-3.3-70B-Instruct					
Q1 (least context)	0.100	40	0.031	96	+0.069
Q2	0.000	86	0.000	50	+0.000
Q3	0.000	103	0.030	33	−0.030
Q4 (most)	0.000	47	0.000	89	+0.000

RS: role-structured; RF⁻: role-free with the evaluator's feedback withheld. Recovery: the share of first-pass-wrong trajectories that finish correct, with n the number of such trajectories in the quartile; the difference is RS minus RF⁻. Context is the total character count of all agent messages in the trajectory; quartile cut points are computed once per actor over both conditions pooled. Because accumulated context is nearly a function of condition (the RF⁻ team writes less because it stops changing its answer), the strata are not balanced and the comparison cannot separate context from condition.

Table 11: Each three-agent team against a single agent of the same actor under the same call cap, on the Omni-MATH benchmark.

organization	team	single agent	difference
gemma-4-31B-it			
role-structured	0.8462	0.8688	−0.0225
	$[-0.0400, -0.0056]$, $p = 0.0122$, $n = 1600$		
role-free	0.8125	0.8688	−0.0563
	$[-0.0737, -0.0394]$, $p = 1.22 \times 10^{-10}$, $n = 1600$		
gpt-oss-120b			
role-structured	0.8031	0.7900	+0.0131
	$[-0.0044, +0.0300]$, $p = 0.152$, $n = 1600$		
role-free	0.8194	0.7900	+0.0294
	$[+0.0125, +0.0456]$, $p = 0.000678$, $n = 1600$		

Realized call budget

actor	protocol	evaluator	actor
gemma-4-31B-it	single agent	1.66	3.16
gemma-4-31B-it	role-structured team	1.11	6.76
gemma-4-31B-it	role-free team	1.95	5.87
gpt-oss-120b	single agent	1.69	3.40
gpt-oss-120b	role-structured team	1.35	7.54
gpt-oss-120b	role-free team	1.85	6.25

Final accuracy over all problems (n) of the role-structured (RS) and role-free, feedback-withheld (RF⁻) teams against a single agent of the same actor solving the same problems under the same cap on model calls; the difference is team minus single agent, with its 95% bootstrap interval over problems and the exact McNemar p. The single-agent trajectories come from an earlier, separately collected trace corpus, joined by problem, and its two actor arms were produced in separate runs: each direction within an actor is a sound guard, the gap between actors is not. Realized call budget: calls per problem to the evaluator and to the actor.