The Interaction Tax: When Communication Erases Diversity in Multi-Agent Teams

Summer Eunhyung Ann Haokun Liu Chenhao Tan

Abstract

Does multi-agent LLM interaction help or hurt? Some work reports gains from debate (Du et al., 2024), critique loops (Chen et al., 2025), and mixture-of-agents synthesis (Wang et al., 2025), while other work finds that interaction adds cost without improving quality under equal budgets (Tran & Kiela, 2026; Xu et al., 2026; Jarrett et al., 2025), or that independent sampling already captures multi-agent gains (Li et al., 2024). We argue this contradiction reflects a missing distinction, because not all multi-agent communication is equal. Different model families find structurally different solutions, but when agents read each other's complete outputs, their proposals converge within one round, erasing the diversity that motivates using multiple models. We call this the *interaction tax*. We test 11 verifier-scored optimization tasks under matched budgets and find that full-solution interaction is a weak default. Independent proposal generation avoids this collapse. Full-solution interaction mainly makes agents stay close to the first solution they see instead of trying different approaches, and critique helps only if the violated rule is easy for the LLM to find and fix. These results suggest that multi-agent performance depends less on the number of agents than on the information they exchange, and interaction helps only when agents share the right information at the right time.

1. Introduction

Multi-agent LLM systems are often used because different models may find different solutions. This is especially useful for optimization tasks, where the goal is to find one strong solution from many possible candidates. A diverse team can cover more of the solution space, but only if those differences survive later interaction.

Recent work gives mixed evidence about whether interaction helps. Debate, critique loops, and mixture-of-agents systems report gains from agents reading, reviewing, or synthesizing each other's outputs (Du et al., 2024; Chen et al., 2025; Wang et al., 2025). Other work finds that multi-agent systems can add cost without improving quality under equal budgets (Tran & Kiela, 2026; Xu et al., 2026; Jarrett et al., 2025), or that independent sampling already captures much of the benefit of using multiple agents (Li et al., 2024). One reason these results can coexist is that some systems let agents read complete solutions before they answer again, while others keep solutions independent until selection or synthesis.

We study this on 11 verifier-scored optimization tasks. Each agent outputs a candidate solution, and a deterministic verifier returns a scalar score. Claude, GPT-4o, and Gemini each achieve the best score on at least one task, while each same-model team scores zero on at least one task. A diverse set of models can therefore find solutions that a same-model team may miss.

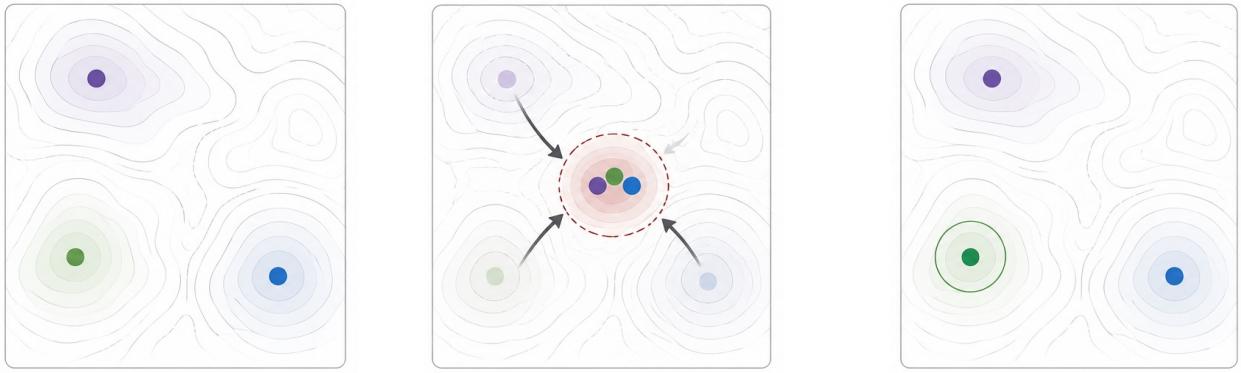
We then test what happens when agents exchange full candidate solutions. When agents read each other's complete outputs, their later solutions become more similar after one round, as shown in Figure 1. This can help same-model teams, where agents start from similar solutions, but it often hurts diverse-model teams, where the useful difference is erased. We call this loss the *interaction tax*.

The tax does not mean that all communication is harmful. MoA (Wang et al., 2025) avoids the collapse because agents produce candidate solutions independently before synthesis. Critique can also help when it shows a fault that the model can find and repair, as in Knapsack, but is less reliable when the fault is harder to locate, as in 3AP-Free.

Our main contribution is to show that full-solution interaction is a weak default. Idit can erase useful model diversity, while independent generation and repair-focused critique avoid this failure in different ways.

[1]Department of Computer Science, University of Chicago, Chicago, IL, USA. Correspondence to: Summer Eunhyung Ann <summereunann@uchicago.edu>.

Proceedings of the 43rd International Conference on Machine Learning, Seoul, South Korea. PMLR 306, 2026. Copyright 2026 by the author(s).



(a) Independent proposals *(b)* Full-solution exchange drives convergence *(c)* Selection preserves diversity

Figure 1. **The interaction tax on optimization tasks.** Different model families (Claude, GPT-4o, Gemini) find structurally different solutions. When agents exchange full candidate solutions, their proposals converge and diversity is lost. When agents generate independently and the best is selected, diversity is preserved.

2. Related Work

Single vs. multi-agent comparisons. Multi-agent debate improves factuality (Du et al., 2024), MAgICoRe raises math accuracy through solver-reviewer-refiner loops (Chen et al., 2025), and MoA synthesis outperforms individual models on language benchmarks (Wang et al., 2025). On the other side, Tran and Kiela (Tran & Kiela, 2026), Xu et al. (Xu et al., 2026), Li et al. (Li et al., 2024), and Jarrett et al. (Jarrett et al., 2025) each demonstrate multi-agent losses under various conditions, but all use same-model teams or do not control for model diversity. Our work isolates model diversity as a variable and tests when interaction preserves or erases the solution diversity that diverse models provide.

Evolutionary approaches. FunSearch (Romera-Paredes et al., 2024), Evolution through Large Models (Lehman et al., 2023), and AlphaEvolve (Georgiev et al., 2025) embed LLMs inside evolutionary loops for mathematical discovery. These approaches preserve population diversity through selection pressure rather than through agent interaction, sidestepping the convergence mechanism we study.

Ensemble diversity theory. The ensemble diversity decomposition (Krogh & Vedelsby, 1994; Hansen & Salamon, 1990; Lakshminarayanan et al., 2017) shows that ensemble error equals mean member error minus a diversity term. When members communicate and converge, diversity shrinks toward zero. The interaction tax is a structural instance of this effect.

3. Methods

Setup. Each task is scored by a deterministic evaluator. Each configuration is a workflow run under identical resource budgets (details in supplementary). Agents see a visible dev evaluator during search. Protocols use this score for refinement, selection, critique, and aggregation. After the run, the dev-selected artifact is evaluated once by a hidden evaluator. Hidden evaluators are stricter, more complete, or robustness-shifted checks designed to test whether a solution found using the visible scorer still holds up offline. We use hidden scores for MEG comparisons and visible feasibility for the two hard-validity analyses. Visible and hidden rankings diverge on three of the tasks analyzed with hidden scores.

We test ten configurations from a larger protocol pool (see supplementary) built from Claude Sonnet 4 (Anthropic, 2025), GPT-4o (OpenAI, 2024), and Gemini 2.5 Flash (Google DeepMind, 2025). Four are single-agent baselines (Single-Shot, Best-of-N sampling, Self-Refine, and Verifier-Guided Search). The remaining configurations are multi-agent workflows spanning sequential revision (Chain), solver-reviewer-refiner critique (MAgICoRe (Chen et al., 2025)), critique exchange (Debate (Du et al., 2024)), plan-and-execute (HPE (Wang et al., 2023)), and independent-proposer synthesis (MoA (Wang et al., 2025)). MoA is the only configuration where proposers never read each other's outputs. The benchmark contains 11 verifier-scored optimization tasks, with four adapted from the AlphaEvolve suite (Georgiev et al., 2025) (full configuration and task details in supplementary). All tasks share the same interface, where agents output JSON candidate solutions and deterministic verifiers return scalar scores. Tasks differ in feedback

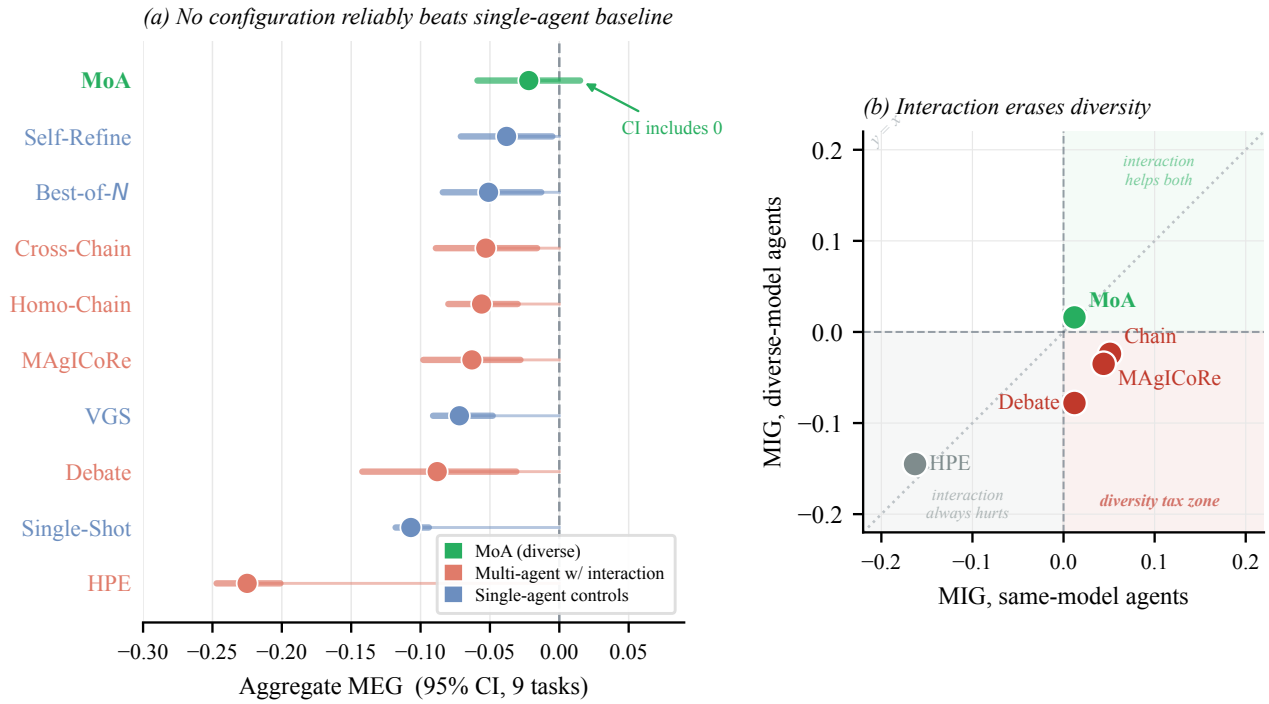


Figure 2. (a) Aggregate MEG for all ten configurations (95% CI, 9 tasks). MoA is the only configuration whose confidence interval includes zero. (b) MIG quadrant scatter (x-axis = same-model MIG, y-axis = diverse-model MIG). MAgICoRe, Chain, and Debate occupy the *interaction tax zone* (bottom-right quadrant), where full-solution interaction helps same-model agents but hurts once models are diverse. MoA escapes to the top-right because its proposers never see each other's outputs.

structure. On graded-feedback tasks (e.g., MaxCut, Circle Packing), almost every well-formed answer receives a meaningful score. On hard-validity tasks (e.g., Knapsack, 3AP-Free), a structural rule must be satisfied before the score is meaningful, and a small edit can violate the rule and collapse the score. Raw scores are normalized to $Q \in [0, 1]$. The main benchmark uses five seeds per cell, and the 2×2 factorial uses ten seeds on three tasks with sufficient score variance (task subsets noted per figure).

Metrics. Positive MEG means the configuration outperformed every single-agent baseline. Positive MIG means interaction improved over independent generation. The diverse parallel baseline is inherently stronger because it benefits from coverage, so negative diverse MIG is a conservative finding (formal definitions in supplementary).

4. Results

Diversity creates coverage. Every same-model team has at least one task with Q=0 (Table 1), so no single model covers all tasks. A diverse team (Claude + GPT-4o + Gemini) never scores zero on any task, even though its aggregate Q is comparable to the best same-model team (capability confound check in supplementary). To test whether this coverage translates into a measurable performance advantage, we run a controlled factorial experiment.

Table 1. Best-of-N Q-scores per model on six tasks where scores diverge. Bold = best model. Each model family wins on different tasks.

Task	Claude	GPT-4o	Gemini	Winner
Circle Packing	**1.000**	0.519	0.745	Claude
TSP-50	**0.141**	—	0.115	Claude
Erdős	0.131	**0.710**	0.142	GPT-4o
TSP-100	0.013	**0.021**	0.010	GPT-4o
DiffBases	0.209	0.264	**0.610**	Gemini
Flat Poly	0.000	0.007	**0.143**	Gemini

A controlled check on the diversity advantage. A 2×2 factorial crosses model diversity (same vs. diverse proposers) with synthesis (MoA synthesis vs. best-score selection) on three tasks with sufficient Q-score variance (N=120, task-stratified bootstrap). The diversity coefficient is +0.188 (CI [+0.073, +0.299], p<0.001, factorial table in supplementary). The synthesis coefficient is near zero. Leave-one-out analysis (supplementary) shows the finding is task-dependent (see Limitations).

The interaction tax. Same-model interaction can refine proposals. Chain, MAgICoRe, and Debate all produce posi-

tive same-model MIG (Table 2). With diverse models, all three turn negative (Chain -0.024, MAgICoRe -0.035, Debate -0.078), with bootstrap $P(\text{same}>\text{div}) \geq 88\%$. For MAgICoRe and Debate, only the models change across conditions, so the loss is consistent with the diversity erased by full-solution interaction. MoA's MIG remains positive from same-model to diverse ($+0.012 \rightarrow +0.016$), which suggests the damage comes from the full-solution interaction step. No configuration achieves positive aggregate MEG (Figure 2).

Table 2. Mean MIG per configuration family (8 tasks). $P(\text{same}>\text{div})$: bootstrap probability that same-model MIG exceeds diverse MIG.

Configuration	Same MIG	Div MIG	P
Chain	$+0.051$	-0.024	97%
MAgICoRe	$+0.044$	-0.035	99%
Debate	$+0.012$	-0.078	88%
HPE	-0.163	-0.145	38%
MoA	$+0.012$	$+0.016$	42%

How convergence happens. This loss is immediate because diversity collapses within a single round of full-solution interaction. On Erdős, diverse Debate achieves a strong intermediate score at round 2 but regresses at round 3 once agents read each other's full solutions (round-by-round scores in supplementary). Across protocols, mean pairwise distance between agents' solution representations falls from 0.315 before interaction to 0.229 after. The damage happens at the moment of full output exchange, not gradually over multiple rounds. The collapse happens because synthesis does not recombine proposals. On five of seven tasks, synthesis copies the best proposer's output $\geq$80% of the time rather than combining parts from each (classification table in supplementary). On Erdős, GPT-4o outputs the identical trivial constant on 100% of seeds (Q=0.710), so any interaction or synthesis step converges to that single point regardless of what Claude or Gemini proposed.

Critique helps only when the fault is easy to locate. In our experiments, the interaction tax reverses only when the shared information points to a concrete, repairable fault. In Knapsack-50, a failed solution usually violates a capacity constraint, and the critic can propose a direct repair by removing or swapping overweight items. Diverse Debate achieves 10/10 feasibility versus 2/10 for same-model Debate (Table 3). In contrast, 3AP-Free-100 also has an exact verifier, but locating the violated rule requires finding an arithmetic-progression triple among many combinations. Diverse Debate drops to 0/10 versus 6/10 for same-model. The same qualitative contrast appears under MAgICoRe, though the Knapsack effect is weaker. On graded-feedback tasks, the first critique round degrades solutions 57% of the time (17/30 runs), consistent with critique offering general guidance rather than a concrete repair (per-run breakdown in supplementary).

Table 3. Explicit feasibility rates on Knapsack and 3AP-Free (n=10 per cell). On Knapsack (fault easy to locate), diverse models help. On 3AP-Free (fault hard to locate), diverse models hurt.

Task	Config	Same	Diverse	Direction
Knapsack-50	MAgICoRe	1/10	2/10	$\approx$
	Debate	2/10	**10/10**	Div. helps
3AP-Free-100	MAgICoRe	6/10	3/10	Div. hurts
	Debate	6/10	**0/10**	Div. hurts

5. Discussion and Conclusion

Our experiments show that full-solution interaction can destroy the diversity it is meant to exploit. When agents read each other's complete outputs, solutions converge within a single round of full-solution interaction. MoA, where proposers never see each other's outputs, is the only configuration whose diverse MIG remains positive.

These results suggest that multi-agent performance depends less on the number of agents than on the information they exchange and when it is exposed. Full candidate solutions create strong convergence. Lower-bandwidth signals, such as scores, method descriptions, or failure causes, may preserve independent exploration while still supporting coordination. Evolutionary approaches like FunSearch (Romera-Paredes et al., 2024) and AlphaEvolve (Georgiev et al., 2025) already do this in effect, preserving population diversity through selection pressure rather than through agent interaction. Critique is most useful when it identifies a concrete repair, as in Knapsack, and can be harmful when the relevant error is difficult for the LLM to locate reliably, as in 3AP-Free.

Limitations and future work. The coverage finding is task-dependent (removing Erdős drops the diversity coefficient to $+0.014$ with CI crossing zero). The main benchmark uses only five seeds per condition, and the diversity factorial uses ten seeds on only three tasks. The explicit feasibility analysis rests on two hard-validity tasks, Knapsack and 3AP-Free, although other tasks such as TSP also have hard structural constraints. All tasks are verifier-scored optimization problems, so the interaction tax may not transfer unchanged to open-ended writing, planning, or dialogue settings. Our experiments identify full-solution interaction as a weak default, but they do not directly test which lower-bandwidth information channel is best. A natural next step is a sharing ablation over scores, method descriptions, failure causes, verified summaries, and late crossover. Code and data are available in the supplementary material.

References

Anthropic. The claude model card, 2025. URL https://docs.anthropic.com/en/docs/about-claude/models.

Bianchi, F., Kwon, M., and Zou, J. EinsteinArena: Harnessing the collective intelligence of agents in the wild to advance science. https://www.together.ai/blog/einsteinarena, 2025. Together AI.

Chen, J. C.-Y. et al. MAgICoRe: Multi-agent, iterative, coarse-to-fine refinement for reasoning. In *EMNLP*, 2025. arXiv:2409.12147.

Du, Y. et al. Improving factuality and reasoning in language models through multiagent debate. In *ICML*, 2024. arXiv:2305.14325.

Georgiev, B., Gómez-Serrano, J., Tao, T., and Wagner, A. Z. Mathematical exploration and discovery at scale, 2025. arXiv:2511.02864.

Google DeepMind. Gemini 2.5: Our most intelligent AI model, 2025. URL https://deepmind.google/technologies/gemini/.

Hansen, L. K. and Salamon, P. Neural network ensembles. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(10):993–1001, 1990.

Jarrett, D. et al. Multi-agent teams hold experts back, 2025. arXiv:2602.01011.

Krogh, A. and Vedelsby, J. Neural network ensembles, cross validation, and active learning. In *Advances in Neural Information Processing Systems*, volume 7, 1994.

Lakshminarayanan, B., Pritzel, A., and Blundell, C. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Lehman, J. et al. Evolution through large models, 2023. arXiv:2206.08896.

Li, J. et al. More agents is all you need, 2024. arXiv:2402.05120.

OpenAI. GPT-4o system card, 2024. arXiv:2410.21276.

Romera-Paredes, B. et al. Mathematical discoveries from program search with large language models. *Nature*, 625: 468–475, 2024.

Sung, J. JSAgent: An ai agent for mathematical discovery on EinsteinArena. https://github.com/jmsung/einstein, 2026.

Tran, D. and Kiela, D. Single-agent LLMs outperform multi-agent systems on multi-hop reasoning under equal thinking token budgets, 2026. arXiv:2604.02460.

Wang, J. et al. Mixture-of-agents enhances large language model capabilities. In *ICLR Spotlight*, 2025. arXiv:2406.04692.

Wang, L. et al. Plan-and-solve prompting: Improving zero-shot chain-of-thought reasoning by large language models. In *ACL*, 2023. arXiv:2305.04091.

Xu, J. et al. Rethinking the value of multi-agent workflow: A strong single agent baseline, 2026. arXiv:2601.12307.

Supplementary Material

This appendix provides workflow details, reproducibility notes, and extended tables and figures for the main paper.

A. Workflow Recipe

The paper's practical claim is a reusable workflow for AI-assisted math/CS/ML optimization.

1. Choose 2–3 diverse model families.
2. Generate candidate solutions independently with no cross-agent visibility.
3. Rank candidates with a visible deterministic checker or development evaluator.
4. If the task exposes local, checkable violations, allow critique-and-revise passes. Otherwise skip interaction and select among the independent candidates.
5. Verify the chosen artifact with a stricter held-out evaluator or deterministic check.

The reported gains motivating this rule are performance and correctness gains, not a claim of universal time savings. Backbone diversity contributes $+0.188$ in the 2×2 factorial (though this is driven by the Erdős task; see Section L), while critique raises feasibility from 0% to 47–73% only on verifiable constraint tasks.

B. Reproducibility and Verification

The artifact includes exact prompt templates for all configurations, pinned model identifiers, task instances, deterministic visible and hidden evaluators, saved JSON run traces with raw model outputs, and scripts that reproduce every table and figure from cached results. This makes the workflow inspectable without re-running API calls and re-runnable within a few hours with ordinary API access. Correctness is checked in three layers. Visible evaluators guide search. Hidden evaluators score the final artifact once. The analysis scripts recompute the reported statistics from saved raw outputs. This separation matters because visible and hidden rankings diverge on three of nine optimization tasks. Throughout this appendix, raw task scores follow each task's native direction. Normalized Q is always higher-is-better.

C. Quickstart Reproduction

The artifact runs on Node.js ≥ 20 and Python ≥ 3.10. No GPU is required.

```
# Install
npm install && pip install -r requirements.txt

# Verify saved scores (~3 min, no API key)
npx tsx tools/evaluate.ts \
  --results results/full-v2 \
  --task bench-difference-bases \
  --verifier both

# Paper tables from cached results
python analysis/analyze_bench.py
# -> results/analysis/meg_full.tsv

# Re-run one cell (needs OPENROUTER_API_KEY)
npx tsx src/runner/run-offline.ts \
  --task bench-difference-bases \
  --protocol moa --seeds 1,2,3
```

The verification step prints a table of recorded vs. recomputed scores; all rows should match exactly. Re-running one task with three seeds takes approximately 5 minutes and costs under $1 via OpenRouter. Results are written to `results/offline/`. The full 11-task, 10-protocol benchmark with 10 seeds takes roughly 4 hours. The artifact README contains the complete reproduction guide.

D. Minimal End-to-End Example

Below is a lightly condensed but faithful excerpt from a single MoA run on Molecule QED (seed 1, higher is better). Three proposers generate independently; the synthesizer sees all three outputs and their dev scores.

Proposer 1 (Claude, devScore $= 0.828$)
`{"smiles": "COc1ccc(Cc2cnc(N)nc2N)cc1"}`

Proposer 2 (GPT-4o, devScore $= 0.685$)
`{"smiles": "O=C(C)Oc1ccccc1C(=O)NCc2ccccc2"}`

Proposer 3 (Gemini, devScore $= 0.547$)
`{"smiles": "CC(=O)NCCC(=O)O"}`

Synthesizer (Claude). Receives all three proposals with scores. Verbatim reasoning:

> "Agent 1's molecule `COc1ccc(Cc2cnc(N)nc2N)cc1` contains: a methoxybenzene ring (drug-like aromatic system), a pyrimidine ring with amino groups (common in pharmaceuticals), connected by a simple methylene bridge... Agent 2's molecule has lower QED likely due to suboptimal LogP or molecular weight, while Agent 3's simple amino acid derivative lacks the structural complexity needed for high QED. Since Agent 1 already achieved an excellent QED score of 0.8279 with a synthetically accessible molecule that balances all drug-likeness parameters optimally, this represents the superior solution."

Final output: `{"smiles": "COc1ccc(Cc2cnc(N)nc2N)cc1"}` (devScore $= 0.828$, hiddenScore $= 0.738$).

The synthesizer correctly identified the best proposal using the interpretable molecular structure and large score gap

($0.83 \gg 0.55$). On tasks where one backbone produces a fixed high-scoring constant (e.g., Erdős), synthesis always converges to that constant, erasing the diversity of the other proposals (see the Erdős transcript in Section V below).

E. Artifact Contents

The repository at https://github.com/SummerAnn/interaction-tax contains: (i) all 11 task instances with embedded dev and hidden evaluators; (ii) verbatim prompt templates for all 10 configurations (src/config/prompts.ts); (iii) pinned model identifiers (src/config/models.ts); (iv) 1,556 saved JSON run traces with raw model outputs, dev scores, and hidden scores; (v) Python analysis scripts that reproduce every table and figure; and (vi) the offline runner for re-executing any cell from scratch via OpenRouter.

F. Full Per-Task MEG

Table 4. Per-task MEG for all ten configurations. MaxCut and LJ-n=41 omitted (all configurations score zero on both tasks).

Task	MoA	SR	BoN	XCh	HCh	MAgI	VGS	Deb	SS	HPE
Circle Pack	−0.24	−0.03	0.00	−0.03	0.00	0.00	0.00	−0.52	0.00	−1.00
DiffBases	+0.09	−0.11	0.00	−0.11	−0.20	−0.15	−0.13	+0.11	−0.20	−0.20
Flat Poly	+0.01	−0.06	−0.08	−0.08	−0.08	−0.08	0.00	−0.04	−0.08	−0.08
TSP-100	−0.02	−0.02	−0.01	−0.01	−0.02	−0.01	0.00	−0.02	−0.01	−0.02
Erdős	+0.02	0.00	−0.37	−0.13	−0.05	−0.15	−0.48	−0.33	−0.48	−0.48
MolQED	−0.01	−0.05	0.00	−0.10	−0.12	−0.14	−0.04	+0.02	−0.11	−0.24
TSP-50	−0.05	−0.08	0.00	−0.02	−0.04	−0.05	0.00	−0.02	−0.09	−0.02

G. Per-Backbone Coverage

Table 5. Per-backbone Best-of-N Q-scores on six tasks where backbone scores clearly diverge. Bold = best backbone. On Erdős, GPT-4o's Q=0.710 reflects a trivial constant (Section Y); Gemini averages Q=0.142 (4/5 seeds score Q=0, one seed matches GPT-4o).

Task	Claude	GPT-4o	Gemini	Winner
Circle Packing	**1.000**	0.519	0.745	Claude
TSP-50	**0.141**	—	0.115	Claude
Erdős	0.131	**0.710**	0.142	GPT-4o
TSP-100	0.013	**0.021**	0.010	GPT-4o
DiffBases	0.209	0.264	**0.610**	Gemini
Flat Poly	0.000	0.007	**0.143**	Gemini

H. Capability Confound Check

Table 6. MoA-no-synthesis aggregate Q-scores across nine optimization tasks (n=10 per condition). The diverse team and GPT-4o×3 have similar point estimates, but each same-model team catastrophically fails on at least one task (Q=0.000), while the diverse team never does.

Configuration	Aggregate Q	Min task Q
Diverse (Claude + GPT-4o + Gemini)	0.349	> 0
GPT-4o×3	0.215	0.000
Gemini×3	0.250	0.000

I. Critique Regression Rates

Table 7. First-round critique regression rate (MAgICoRe, Gemini backbone, n=15 per task). Constraint = Knapsack-50, 3AP-Free-100. Optimization = Erdős, Flat Polynomials.

Task type	Regress 0→1	Any regress	n
Constraint (2 tasks)	**0/30** (0%)	5/30 (17%)	30
Optimization (2 tasks)	17/30 (57%)	25/30 (83%)	30

Table 8. Feasibility success rates on two constraint tasks (Gemini backbone, n=15 each). Fisher exact p compares MAgICoRe to Best-of-N.

Task	BoN	Self-Ref	MAgICoRe	p
Knapsack-50	0%	13%	47%	0.003
3AP-Free-100	0%	7%	73%	<0.001

J. MIG per Configuration Family

Table 9. Mean MIG per configuration family (8 tasks). P(same>div): bootstrap probability that same-model MIG exceeds diverse MIG.

Configuration	Same MIG	Div MIG	P
Chain	+0.051	−0.024	97%
MAgICoRe	+0.044	−0.035	99%
Debate	+0.012	−0.078	88%
HPE	−0.163	−0.145	38%
MoA	+0.012	+0.016	42%

K. 2×2 Factorial Estimates

Table 10. 2×2 factorial estimates (N=120, 10,000 task-stratified bootstrap). Diversity is the only factor whose CI excludes zero.

Term	Estimate	95% CI
Intercept (same + selection)	+0.182	[+0.123, +0.246]
Diversity	+0.188	[+0.073, +0.299]
Synthesis	−0.010	[−0.111, +0.094]
Diversity × Synthesis	+0.046	[−0.132, +0.226]

L. Leave-One-Out Sensitivity

Table 11. Leave-one-out sensitivity of the diversity main effect. Only removing Erdős drops the coefficient to near zero.

Task withheld	$\hat{\beta}_{\text{div}}$	95% CI	p
None (N=120)	+0.188	[+0.073, +0.299]	<0.001
Erdős (n=80)	+0.014	[−0.109, +0.139]	0.84
DiffBases (n=80)	+0.265	[+0.142, +0.381]	<0.001
MolQED (n=80)	+0.286	[+0.125, +0.442]	<0.001

M. Convergence Mechanism

Step-level trajectories reveal when diversity dies. On Erdős, diverse Debate achieves a strong intermediate score at

round 2, when agents exchange only critiques, but collapses at round 3 once agents read each other's full solutions. The damage occurs at the moment of full output exchange, not gradually.

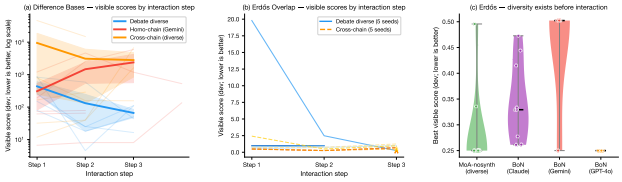


Figure 3. Per-round visible-score trajectories on two optimization tasks. (a) DiffBases: sequential interaction often improves at step 2 but loses that advantage by the final step. (b) Erdős: diverse interaction improves at round 2 then regresses at round 3 after agents exchange full outputs. (c) Erdős: diverse MoA-nosynth achieves broader coverage than same-backbone Best-of-N.

N. Diversity Collapse

Table 12. Mean pairwise solution distance. After synthesis, diverse MoA distance (0.315) falls to 0.229.

Configuration	Avg pairwise distance
MoA (no synthesis)	0.315
Best-of-N (same model)	0.312
Debate	0.309
MoA (diverse + synthesis)	0.229

O. Visible–Hidden Rank Agreement

Table 13. Spearman rank correlation between visible and hidden configuration rankings per task (n=10 configurations). Three tasks show $\rho < 1$.

Task	ρ	Top-3 overlap	Top-1 inv?
MaxCut	0.891	2/3	Yes
Circle Packing	1.000	3/3	No
DiffBases	0.806	2/3	No
Flat Poly	1.000	3/3	No
TSP-100	1.000	3/3	No
LJ-n=41	1.000	3/3	No
Erdős	1.000	3/3	No
MolQED	0.952	3/3	No
TSP-50	1.000	3/3	No

P. Per-Task MIG Breakdown

Table 14. MIG per task for five configuration families (eight optimization tasks). Bold marks Erdős sign flips. On tasks with no Q-score variance (MaxCut, LJ-n41), MIG is zero for all configurations.

Family	Cfg	MCut	CPack	DBases	FPoly	TSP1	LJ41	Erdős	MolQ
MAgICoRe	homo	0.00	0.00	+0.05	0.00	0.00	0.00	**+0.33**	−0.03
MAgICoRe	mixed	0.00	0.00	−0.07	0.00	−0.01	0.00	**−0.28**	+0.09
Debate	homo	0.00	−0.52	+0.31	+0.04	−0.01	0.00	+0.14	+0.13
Debate	mixed	0.00	−0.60	−0.14	+0.02	0.00	0.00	+0.14	−0.04
Chain	homo	0.00	0.00	0.00	0.00	−0.01	0.00	**+0.43**	−0.01
XChain	mixed	0.00	−0.03	−0.06	−0.03	+0.01	0.00	**−0.08**	+0.01
MoA	homo	0.00	−0.08	+0.07	0.00	−0.01	0.00	+0.07	+0.04
MoA	mixed	0.00	−0.24	+0.15	+0.06	−0.01	0.00	+0.07	+0.10

Q. Agent Configurations

Table 15. Agent configurations. All operate under the same budget vector (T=200K, W=600s, C=30s, K=25). HPE = Hierarchical Planner-Executor. Two MoA ablations (same-model proposers, no synthesis) are used in the 2×2 analysis.

Configuration	Role	Mechanism	Ref.
Single-Shot	Control	One model, one attempt, no iteration.	—
Best-of-N (N=8)	Control	8 independent samples, return visible-best.	—
Self-Refine	Control	Same model generates, critiques, revises for 3 rounds.	Madaan et al., 2023
VGS	Control	Evolutionary loop (pop. 4, gen. 5, elite 2) with visible scores as fitness.	Romera-Paredes et al., 2024
Homo-Chain	Multi	4 same-model agents in sequence, each revises prior output. Visible-best returned.	This work
Cross-Chain	Multi	3-agent chain (Claude → GPT-4o → Gemini). Visible-best returned.	This work
MAgICoRe	Multi	Solver → reviewer → refiner, 2 rounds.	Chen et al., 2025
Debate	Multi	2 agents propose, exchange critiques (2 rounds), synthesizer merges.	Du et al., 2024
HPE	Multi	Planner decomposes, executors solve subproblems, integrator assembles.	Wang et al., 2023; Hong et al., 2023
MoA	Multi	3 proposers generate independently, 1 synthesizer combines.	Wang et al., 2025

All proposer calls use temperature 0.7; critique and synthesis calls use 0.3–0.5, consistent across backbones.

R. Synthesis Decision Quality

Table 16. Synthesis output classified relative to the best-scoring proposer. On five of seven tasks synthesis copies the best proposal ≥80% of the time. On Difference Bases it degrades 50% of the time.

Task	Improved	Copied	Degraded
Circle Packing	60%	40%	0%
Molecule QED	20%	80%	0%
Erdős	10%	80%	10%
Flat Polynomials	0%	100%	0%
TSP-100	0%	100%	0%
TSP-50	0%	90%	10%
Difference Bases	7%	43%	50%

S. Within-Backbone Comparisons (n=15)

Table 17. Within-backbone comparisons (Gemini, n=15 per condition). Positive d = iterative configuration is worse.

Task	Config	Iter	BoN	d	p
Flat Poly	MAgI-G	2.087	1.899	+1.25	0.002
Flat Poly	SR-G	2.020	1.899	+1.02	0.009
TSP-100	MAgI-G	53,351	50,786	+1.42	0.001
TSP-100	SR-G	53,239	50,786	+1.46	0.001
TSP-50	MAgI-G	24,167	23,103	+1.23	0.002
MolQED	MAgI-G	0.863	0.859	+0.05	0.93

T. Task Descriptions

Table 18. Benchmark tasks. Four optimization tasks are adapted from AlphaEvolve (Georgiev et al., 2025) with smaller parameterizations. Two constraint tasks test interaction under checkable violations.

Task	Domain	Type	Objective
Circle Packing	Geometry	Optimization	Maximize radius of n non-overlapping circles in a unit square
Difference Bases	Combinatorics	Optimization	Find a set whose pairwise differences cover all residues mod n
Erdős Overlap	Analysis	Optimization	Minimize $\max_t \int \lvert f(x)f(x+t)\rvert\, dx$ over translates
Flat Polynomials	Algebra	Optimization	Minimize $\lVert p\rVert_\infty$ for a polynomial with ± 1 coefficients
TSP-50 / TSP-100	Routing	Optimization	Minimize tour length for 50 or 100 cities
MaxCut	Graph theory	Optimization	Maximize cut value in a fixed graph
LJ-n=41	Chemistry	Optimization	Minimize Lennard-Jones potential for 41 atoms
Molecule QED	Drug design	Optimization	Maximize quantitative estimate of drug-likeness (QED score)
Knapsack-50	Combinatorics	Constraint	Maximize value subject to weight capacity constraint
3AP-Free-100	Combinatorics	Constraint	Find the largest 3-AP-free subset of $\{1, \ldots, 100\}$

MaxCut and LJ-n=41 are omitted from most analyses because all configurations score Q=0 on both tasks under the given budget, providing no inter-configuration signal.

U. Representative Prompt Templates

Full prompt templates for all ten configurations are available in the release artifact. Below we show the structure of two representative prompts.

Proposer prompt (used in MoA, Best-of-N, Single-Shot).

```
System: You are a mathematical optimization
agent. Your goal is to [OBJECTIVE]. Return
ONLY executable Python code that prints the
solution as a JSON object.

User: [Full task description with parameters,
constraints, input data, and evaluation
function specification.]
Output format: a single Python code block.
```

Reviewer prompt (used in MAgICoRe, Debate).

```
System: You are a mathematical reviewer.
Analyze the proposed solution for correctness.
List specific errors, constraint violations,
or improvements.

User: Task: [OBJECTIVE]
Proposed solution (scored [VISIBLE_SCORE]):
[SOLUTION_CODE]

Provide a numbered list of issues and concrete
suggestions for improvement.
```

The proposer prompt is identical across configurations; only the orchestration (whether agents see prior outputs) differs. The reviewer prompt is used only in critique-based configurations. All prompts, including the synthesizer prompt for MoA and the planner/executor prompts for HPE, are included verbatim in the artifact.

V. Example Multi-Agent Exchange

Below is a condensed transcript from a Cross-Chain run on Erdős (seed 1). On Erdős, *lower* raw scores are better; after Q-normalization, higher Q is always better.

Example 1. Sequential interaction converges to a degenerate constant (Cross-Chain, Erdős, seed 1).

Step 1 (Claude, dev score $= 0.500$). Claude reasons about minimizing overlap with translates and produces a structured sparse function using binary patterns.

Step 2 (GPT-4o, dev score $= 0.250$). GPT-4o sees Claude's solution but outputs a trivial constant $f = 0.5$. GPT-4o defaults to near-constant functions on this task regardless of input context (Q=0.710, best single backbone on Erdős).

Step 3 (Gemini, dev score $= 0.694$). Gemini produces a structured binary pattern but scores worse than GPT-4o. The chain's visible-best selection picks GPT-4o's output (0.250, lowest hence best).

GPT-4o's trivial constant is genuinely the best-scoring solution (Q=0.710), but it contributes zero strategy diversity because every GPT-4o seed produces the identical output (Table 21). When a chain includes GPT-4o on Erdős, the non-trivial solutions from Claude and Gemini are discarded in favor of a degenerate constant. Under independent generation (MoA-nosynth), the same three models produce raw scores of 2.47 (Q=0), 0.34 (Q=0.710), and 0.50 (Q=0); the best (0.34, Q=0.710) is selected, outperforming any single-backbone run.

W. Per-Task 2×2 Cell Means

Table 19. Mean Q-scores for each arm of the 2×2 diversity $\times$ synthesis factorial, broken down by task (n=10 per cell). The diversity contrast shows that Erdős drives the aggregate diversity coefficient, DiffBases contributes moderately, and MolQED contributes near zero. The synthesis contrast is near zero on all three tasks.

Task	Same+Sel	Same+Syn	Div+Sel	Div+Syn	Div contr.	Syn contr.
DiffBases	0.202	0.071	0.237	0.287	+0.126	−0.041
Erdős	0.031	0.071	0.568	0.497	+0.482	−0.016
MolQED	0.314	0.377	0.307	0.437	+0.026	+0.097

The per-task breakdown is essential context for the leave-one-out result (Section L). Erdős contributes $+0.482$ to the diversity contrast, which is $2.5\times$ larger than DiffBases ($+0.126$) and $19\times$ larger than MolQED ($+0.026$). Removing Erdős therefore drops the aggregate coefficient from $+0.188$ to $+0.014$. The synthesis contrast is near zero or slightly negative on all three tasks, confirming the null synthesis finding is not an artifact of task selection.

X. Cross-Synthesizer Replication

Table 20. Diversity coefficient under three synthesizer backbones. The coefficient remains positive and significant for all three synthesizers. The synthesis coefficient remains near zero regardless of which model synthesizes.

Synthesizer	$\hat{\beta}_{\text{div}}$	95% CI
Claude Sonnet 4	$+0.234$	$[+0.106,\ +0.355]$
GPT-4o	$+0.176$	$[+0.049,\ +0.303]$
Gemini 2.5 Flash	$+0.170$	$[+0.044,\ +0.298]$

The diversity coefficient is stable across synthesizer choices. All three synthesizer backbones produce a positive, significant diversity coefficient. This rules out the possibility that the null synthesis effect (Table 10) is an artifact of a particular synthesizer being too weak or too strong to combine diverse proposals.

Y. Model Strategy Classification

Table 21. Strategy frequency per backbone on Erdős and DiffBases, classified from rawOutput keywords across all single-shot seeds. Categories are non-exclusive keyword tags. GPT-4o outputs a trivial constant ($\approx$0.005) on 100% of Erdős runs. Gemini also produces near-trivial constants (all zeros, raw score $\approx$0.503). Only Claude generates structurally varied, non-trivial solutions. On DiffBases, GPT-4o uses triangular numbers on every run.

Task	Strategy type	Claude	GPT-4o	Gemini
Erdős	Trivial constant	0/10	**5/5**	**5/5**
	Binary/block pattern	4/10	0/5	0/5
	Smooth/cosine-based	2/10	0/5	0/5
	Mathematical reasoning	6/10	0/5	0/5
DiffBases	Sidon sets (B_2 sets)	4/10	0/5	1/5
	Triangular numbers	1/10	**5/5**	1/5
	Arithmetic progression	1/10	0/5	2/5
	Greedy/algorithmic	2/10	0/5	0/5

On Erdős, GPT-4o and Gemini each produce a single stereotyped output across all seeds, contributing zero strategy variance to any diverse ensemble. A "three-backbone" team effectively has one contributor. On DiffBases, GPT-4o is similarly fixed (triangular numbers on 100% of runs), though Gemini and Claude each use multiple strategies. The effective diversity of an ensemble depends on whether each backbone explores different regions of the solution space, not merely on the number of distinct model families.

This has a direct consequence for the MIG results. When a diverse Debate or Cross-Chain team includes GPT-4o on Erdős, one agent always proposes the same trivial constant (Q=0.710). Although this constant scores highest, it contributes zero strategy variance, so the nominal three-backbone team has effectively one non-trivial contributor. The interaction step risks discarding Claude's or Gemini's structurally different solutions in favor of this single fixed output, producing the regression visible in the per-task MIG breakdown (Section P).

Z. Within-Model vs. Between-Model Variance

Table 22. Variance decomposition of single-shot Q-scores across three backbones (n=5 seeds each). Between-model variance (σ_B^2) measures how different the model means are. Within-model variance (σ_W^2) measures per-seed noise. A B/W ratio above 1 indicates that models are more different from each other than from themselves, the prerequisite for backbone diversity to help. Two tasks clearly meet this criterion ($B/W \geq 2$) and two are marginal ($B/W \approx 1.2$–1.3). LJ-n=41 and TSP-50 are omitted (LJ because all models score Q=0, TSP-50 for incomplete backbone coverage).

Task	σ_W^2	σ_B^2	B/W	Diversity useful?
Circle Packing	0.017	0.253	15.2	Yes
MolQED	0.015	0.030	2.0	Yes
Flat Poly	0.054	0.070	1.3	Marginal
Erdős	0.050	0.061	1.2	Marginal
TSP-100	0.020	0.010	0.5	No
MaxCut	0.0001	0.00002	0.3	No
DiffBases	0.095	0.016	0.2	No

Diversity adds information primarily on tasks where models make structurally different errors ($B/W \geq 2$). On tasks where within-model noise dominates ($B/W < 1$), adding a different backbone contributes no more than resampling the same one. This is consistent with the interaction tax theory. The tax only appears where there is genuine diversity to destroy. Note that B/W measures separation of model *means*, while the 2×2 diversity contrast measures the gap between the best draw from a diverse team and the best draw from a same-model team. On DiffBases (B/W=0.2, diversity contrast +0.126), within-model noise is high, but Gemini's mean (Q=0.610) far exceeds Claude's (Q=0.209), so a diverse team's best draw is still better even though individual seeds are noisy.

AA. Inter-Model Error Correlation

Table 23. Spearman rank correlations of per-seed hidden scores between model pairs on four tasks with nonzero cross-model Q-score variance. Low or negative ρ indicates uncorrelated errors (diversity helps). Positive ρ indicates correlated errors (diversity adds less). Gemini is constant on Erdős (all seeds produce Q=0.000), precluding correlation.

Task	Cl–GPT ρ	Cl–Gem ρ	GPT–Gem ρ	Error indep.
Erdős	+0.18	— (const)	— (const)	Yes
DiffBases	+0.05	+0.12	+0.17	Yes
MolQED	−0.45	−0.20	−0.33	Yes (anti-corr)
TSP-100	+0.38	+0.50	+0.41	No (correlated)

AB. Cross-Backbone Replication

The optimization loss pattern replicates on both tested backbones (Gemini and GPT-4o).

Table 24. MAgICoRe vs. Best-of-N on two optimization tasks, two backbones (n=15 each). The loss replicates on both tested backbones, with large effect sizes.

Task	Backbone	MAgICoRe	Best-of-N	d	p
Flat Poly	Gemini	2.087	1.899	+1.25	0.002
Flat Poly	GPT-4o	2.951	2.251	+1.30	0.001
TSP-100	Gemini	53,351	50,786	+1.42	0.001
TSP-100	GPT-4o	53,529	50,128	+2.64	<0.001

AC. Constraint Feasibility Gradient

Within the Gemini backbone (n=15 each), constraint feasibility generally increases with configuration sophistication. Critique-based configurations outperform evolutionary search on both tasks. Single-Shot outperforms Best-of-N on Knapsack (7% vs. 0%), breaking a strict monotonic gradient at the sampling end.

Table 25. Constraint feasibility rates by configuration type (Gemini backbone, n=15). Critique outperforms evolution on both tasks.

Task	S-Shot	BoN	VGS	MAgICoRe
Knapsack-50	7%	0%	27%	47%
3AP-Free-100	0%	0%	13%	73%

AD. Constraint MIG: Same-Model vs. Diverse Backbones

The interaction tax reverses on constraints with trivially checkable violations but persists when verification requires reasoning. On Knapsack-50, where a weight violation is an arithmetic check, diverse Debate achieves 10/10 feasibility versus 2/10 for same-model. On 3AP-Free-100, where verifying a three-term arithmetic progression requires reasoning, diverse Debate drops to 0/10 versus 6/10 for same-model. The pattern replicates across MAgICoRe.

Table 26. Constraint feasibility rates by backbone condition (n=10 per cell). On Knapsack (trivially checkable), diverse backbones help. On 3AP-Free (reasoning-required), diverse backbones hurt.

Task	Configuration	Same	Diverse	Direction
Knapsack-50	MAgICoRe	1/10	2/10	≈
	Debate	2/10	**10/10**	Div. helps
3AP-Free-100	MAgICoRe	6/10	3/10	Div. hurts
	Debate	6/10	**0/10**	Div. hurts

AE. Formal Metric Definitions

Raw scores are normalized to $Q(s) = (s - b)/(r - b)$, clipped to $[0, 1]$, where b is the trivial baseline and r the best known result.

Marginal Epistemic Gain (MEG) measures whether a configuration beats the best single-agent baseline:

$$\mathrm{MEG}(p,t) = Q_{\mathrm{hidden}}(p,t) - \max_{c \in \mathcal{C}} Q_{\mathrm{hidden}}(c,t),$$

where $\mathcal{C} = \{\text{Self-Refine}, \text{Best-of-}N, \text{VGS}\}$.

Marginal Interaction Gain (MIG) measures whether interaction helps relative to the same agents running independently:

$$\mathrm{MIG}(p,t) = Q_{\mathrm{hidden}}(p,t) - Q_{\mathrm{hidden}}(\mathrm{parallel}(A),t).$$

The 2×2 factorial uses N=120 runs total (ten per arm on each of the three tasks: Erdős, DiffBases, and MolQED).

AF. Step-Level Solution Distance

Table 27. Pairwise solution distance between consecutive steps within each run, averaged over seeds. On Debate diverse, distance drops sharply after agents exchange outputs (round 2→3), confirming diversity collapse in real time. Metrics: Jaccard distance for DiffBases (discrete sets), cosine distance for Erdős (continuous arrays), Levenshtein distance for MolQED (SMILES strings).

Task	Configuration	Step 1→2	Step 2→3	Pattern
DiffBases	Debate (same)	0.950	0.891	Slight conv.
	Debate (diverse)	0.956	**0.479**	Collapse r3
	MAgICoRe (div.)	0.953	0.815	Convergence
Erdős	Debate (same)	0.500	—	Single change
	Cross-Chain	0.128	0.199	Low throughout
	MAgICoRe (div.)	0.448	0.293	Shrinking
MolQED	Debate (same)	0.571	0.412	Converging
	Debate (diverse)	0.669	**0.378**	Fast conv.

AG. Per-Round Debate Scores

Table 28. Per-round visible scores for same-model and diverse Debate on two tasks (lower is better for both, bold = best round). Same-model Debate improves monotonically. Diverse Debate reverses direction at round 3 after agents have exchanged their full outputs.

Configuration	Task	Round 1	Round 2	Round 3
Debate (same)	DiffBases	436	132	**66**
Debate (diverse)	DiffBases	889	**72**	274
Debate (same)	Erdős	4.61	1.24	**0.62**
Debate (diverse)	Erdős	1.28	**0.30**	0.38

On both tasks, diverse Debate peaks at round 2 (when agents have exchanged critiques but not full solutions) and then regresses at round 3 after full output exchange. Same-model Debate improves monotonically because there is no structural diversity to destroy.

AH. MAgICoRe Step-by-Step Trajectories

Table 29. Mean devScore at each MAgICoRe step (step 0 = initial attempt, step 1 = after first critique, step 2 = after second critique), averaged over n=15 seeds. Direction arrows indicate whether lower (↓) or higher (↑) is better. *Only 2/15 Erdős seeds reach step 2.

Task	Dir	Step 0	Step 1	Step 2	Trend	Note
TSP-100	↓	54,054	53,686	53,638	Slight impr.	Still below BoN
Flat Poly	↓	2.290	2.297	2.345	Degrading	Values, not struct.
Mol. QED	↑	0.731	**0.830**	0.785	Peak revert	Step 1 lost
Erdős	↓	0.699	0.504	—*	Conv. trivial	12/15 → 0.250

On optimization tasks, critique frequently degrades solutions. Molecule QED peaks at step 1 ($+13.5\%$) but reverts at step 2. On Erdős, 12 of 15 seeds converge to the trivial constant function at raw score 0.250 (Q=0.710). The critic steers Gemini toward the simplest possible answer, which scores highest but is structurally degenerate. Flat Poly shows no structural change across steps. These trajectories are the per-run manifestation of the $0/30$ vs. $17/30$ first-round regression rate in Section I.

AI. Proposer Score Spread

Table 30. Mean pairwise score spread among MoA proposers within each run, measuring how different the proposals are before synthesis. Diverse proposers produce $2.2\times$ more score spread than same-model proposers on Erdős. On DiffBases, within-model variance dominates, so backbone diversity adds less spread.

Task	Diverse spread	Same spread	Ratio
Erdős	0.448	0.207	$2.2\times$
MolQED	0.222	0.152	$1.5\times$
DiffBases	1500	3045	$0.5\times$

Diversity is real at proposal time. On Erdős, diverse proposals are $2.2\times$ more spread out than same-model proposals, confirming that different backbones explore different regions of the solution space. On DiffBases, within-model noise dominates ($0.5\times$), consistent with the B/W ratio of 0.2 in Section Z. Synthesis erases this diversity (Section N).

AJ. Relation to Open-Ended Agent Systems

Open-ended platforms such as EinsteinArena (Bianchi et al., 2025) show that multi-agent systems can produce real scientific results. These systems operate in a richer regime than our controlled benchmark, with many attempts over long horizons, public verifiers, leaderboards, discussion, partial results, and accumulated findings. Public descriptions of EinsteinArena and agents such as JSAgent (Sung, 2026) emphasize selective information channels, including scores, notes, failure causes, and reusable findings, rather than only short-horizon peer-to-peer exchange of complete candidate solutions. Our controlled experiments suggest a possible

mechanism consistent with such architectures. Selective information may let agents compound useful knowledge without forcing every solver to condition on a complete peer solution early in generation.

AK. Additional Transcript Examples

Examples 2–5 complement Example 1 in Section V. Together they illustrate synthesis failure, synthesis success, homogenization-as-improvement, and preserved diversity.

Example 2. Synthesis selects a degenerate solution (MoA, Erdős, seed 1). Three proposers generate independently (lower scores are better on Erdős). Claude produces a structured binary-pattern function (devScore $= 0.490$, Q=0). GPT-4o outputs a trivial uniform array of constant 0.005 values (devScore $= 0.250$, Q=0.710). The synthesizer selects GPT-4o's output, constructing a mathematical justification:

> "Agent 2 uses a uniform distribution with constant value 0.005 across all points. This achieves exactly the theoretical lower bound of $1/4$ for uniform distributions... The uniform approach is mathematically optimal... Agent 2's solution is superior as it achieves the lower bound with perfect consistency."

The synthesizer correctly picked the higher-scoring proposal (Q=0.710 vs. Q=0), though the mathematical reasoning it gives is wrong. The problem is the loss of strategy diversity. GPT-4o produces this identical trivial constant on every seed (Table 21), so synthesis always converges to the same degenerate output regardless of what Claude or Gemini propose. Diversity existed at proposal time but was erased at synthesis.

Example 3. Synthesis selects the right proposal (MoA, MolQED, seed 1). Three proposers produce structurally distinct molecules (higher scores are better on MolQED). Claude produces a methoxybenzene-pyrimidine scaffold (devScore $= 0.828$). GPT-4o produces an aspirin derivative (devScore $= 0.685$). Gemini produces a small acetyl amino acid (devScore $= 0.547$). The synthesizer identifies Claude's molecule as containing "a methoxybenzene ring (drug-like aromatic system) and a pyrimidine ring with amino groups (common in pharmaceuticals)" and correctly selects it. Here the quality gap is large ($0.83 \gg 0.55$) and the solution format is chemically interpretable. Synthesis works when the best proposal's advantage is obvious.

Pattern. On Erdős, the synthesizer correctly selects the highest-scoring proposal, but that proposal is a structurally degenerate constant that GPT-4o produces on every seed. On MolQED, the synthesizer also selects the best proposal, but here all three proposals are structurally distinct. The difference is whether diversity survives. When one backbone dominates on score with a fixed output, synthesis erases diversity by always converging to that output. This is consistent with the null synthesis coefficient (Table 10). Across tasks, synthesis neither reliably helps nor reliably hurts.

Example 4. Homogenization improves score (Debate diverse, DiffBases, seed 1). Round 1 (devScore $= 597$). The model reasons creatively about fractal-like structures for a difference basis set. Round 2 (devScore $= 157$). After seeing the first model's approach, the agent shifts to a standard triangular-number construction. Hidden score is 70.2. This is a positive example of interaction. The solution improved by replacing a creative but poorly executed approach with a conventional mathematical construction. But the improvement came from homogenization. The solution got better *by becoming less creative*. This is exactly the mechanism the diversity coefficient captures at aggregate. Interaction rewards convergence toward standard approaches.

Example 5. Independent generation preserves diversity (MoA-nosynth, Erdős, seed 1). Three proposers generate independently (lower is better). Claude scores 2.47 (structured sparse function). GPT-4o scores 0.34 (compact near-constant solution). Gemini scores 0.50 (different interpolation). The configuration picks the best (GPT-4o's 0.34), which is better than any Claude-only attempt on the same seed (best of three is 0.392). The diverse ensemble finds a solution Claude does not naturally produce. The gain is real, but only because the agents generated without seeing each other.

AL. Protocol Implementation Details

All ten configurations share the budget vector (T=200K tokens, W=600s, C=30s, K=25) and the same task prompt. The selection rule is: the artifact with the best dev score under the task's score direction is submitted for hidden evaluation. Below we summarize implementation details from the release artifact. Full source code is available at `https://github.com/SummerAnn/interaction-tax`.

Backbone models. All models are accessed via OpenRouter: Claude Sonnet 4 (`anthropic/claude-sonnet-4`), GPT-4o (`openai/gpt-4o`), Gemini 2.5 Flash (`google/gemini-2.5-flash`). Model IDs are pinned in the codebase.

Best-of-N. Eight independent calls at temperatures $0.60, 0.65, \ldots, 0.95$. Temperature variation introduces sample diversity without changing the prompt.

VGS. FunSearch-style evolutionary loop. Initial population of four at varied temperature. Five generations, top-two elites retained, two new candidates per generation via mu-

tation call. Early termination if no improvement for two consecutive generations.

MAgICoRe roles. Solver generates (SINGLE_SHOT prompt). Reviewer critiques without producing code ("What are the weaknesses? Be concrete and technical."). Refiner receives critique and produces improved solution. Two rounds.

Debate. Debater A uses exploratory strategy (temperature 0.9): "Take an EXPLORATORY approach — try unconventional strategies." Debater B uses rigorous strategy (temperature 0.5): "Take a RIGOROUS approach — use well-established methods." Two critique rounds, then a neutral synthesizer merges.

MoA. Three proposers generate independently with no shared context. Aggregator receives all three proposals with dev scores and produces synthesized output. The aggregator prompt includes: "Multiple agents independently solved this problem. Synthesize their best ideas into a superior solution."

HPE. A planner decomposes the problem into sub-foci for parallel executors. Each executor works only on its assigned focus and produces a partial contribution (free-form text, not a complete solution). An integrator assembles the contributions into one schema-valid solution. Multiple rounds give the planner score feedback.

Retry behavior. All protocols share a common retry mechanism. If the LLM produces malformed JSON or a verifier-rejected solution, the error message is appended to the next call. Generation steps allow up to 5 retries; refinement steps allow up to 2 to conserve budget. A cell that exhausts retries at the first generation step is excluded from analysis.

Budget enforcement. A shared BudgetEnforcer tracks cumulative token usage, wall clock time, and evaluator call count. Any call exceeding a cap is silently dropped and the protocol terminates early.

AM. Extended Prompt Templates

Section U shows two representative prompts. Here we include additional templates verbatim from the release artifact.

MoA aggregator prompt.

```
System: You are an Aggregator. Multiple agents
independently solved this problem. Synthesize their
best ideas into a superior solution. Return ONLY
valid JSON.

User: # Problem
[PROBLEM]

# Schema
[SCHEMA]

# Agent Solutions
[CANDIDATES with dev scores]

Synthesize the best solution. Output ONLY JSON.
```

Debate synthesizer prompt.

```
System: You are a neutral Synthesizer. Two agents
debated this problem. Combine the best ideas from
both into an optimal solution. Return ONLY valid
JSON.

User: # Problem
[PROBLEM]

# Schema
[SCHEMA]

# Debater A (score: [SCORE_A])
[POSITION_A]

# Debater B (score: [SCORE_B])
[POSITION_B]

Synthesize the best solution. Output ONLY JSON.
```

MAgICoRe reviewer prompt.

```
System: You are a REVIEWER. Analyze the solution
below and provide specific, actionable critique.
What are the weaknesses? What strategies could
improve the score? Be concrete and technical.

User: # Problem
[PROBLEM]

# Current Solution (score: [SCORE], goal: [DIR])
[SOLUTION]

Provide your critique.
```

HPE planner prompt.

```
System: You are a PLANNER. Decompose this problem
into focused sub-tasks for [N] executor agents
working in parallel. Each executor works ONLY on
its assigned sub-focus, then a separate integrator
will combine the contributions into one final
solution.

Return ONLY valid JSON:
{"strategy": "one-sentence plan",
 "subtasks": [{"id": 1, "focus": "..."},
              {"id": 2, "focus": "..."}, ...]}

Make the foci genuinely complementary -- they should
NOT all attempt the whole problem.
```

All ten protocol prompt sets (including VGS mutation, Self-Refine iteration, Homo-Chain refinement, Cross-Chain refinement, HPE executor, and HPE integrator) are committed verbatim in the artifact at `src/config/prompts.ts`.