COMPOSITIONAL RL: EVALUATING AND TRAINING LLMS FOR COLLABORATIVE REASONING

Anonymous authors
Paper under double-blind review

ABSTRACT

Language models are often evaluated alone but increasingly deployed together, routed between, made to debate, ensembled, or collaboratively trained. For a model inside a system, what matters most is not how well it performs on its own, but how much the system gains from its presence. We call this a model's *compositional strength* and formalize it using the Shapley value of a cooperative game over coalitions of a model pool, and use a model's leave-one-out contribution to the full coalition as a tractable proxy, requiring $n + 1$ rather than 2^n system evaluations. Evaluating individual vs. compositional strengths across five model pools, twelve collaboration methods, and twelve datasets, we find them largely uncorrelated: in 12 of 15 (pool, method-family) settings, individual strength does not translate into compositional advantages, and the coupling is weakest among general-purpose models that practitioners are most likely to deploy. Standalone benchmarks are therefore often poor proxies for models' value in collaborative systems. We then propose *compositional reinforcement learning* (C-RL), which trains a policy inside a live multi-model system and rewards the outcome produced by the system as a whole, rather than the policy's correctness in isolation. The leave-one-out baseline in Shapley values cancels under a group policy gradient, so this costs no more than ordinary RL and never requires running a coalition without the policy. Across three collaboration protocols and eight tasks spanning reasoning, coding, and scientific discovery, C-RL produces the strongest collaborator in 21 of 24 settings, lifting collaborative performance by $+4.0$ to $+5.4$ pts on average ($\sim$3x the gain of baselines), up to $+13.1$ pts. C-RL is the only method that never leaves the system worse off than the untrained policy, and carries over to unseen datasets, unseen collaboration protocols, and pools of frontier proprietary models.

1 INTRODUCTION

Language models are almost always evaluated alone. A model answers a benchmark, is scored, and lands on a leaderboard, all by itself (Liang et al., 2023; Chiang et al., 2024). A growing share of deployed systems, however, do not use models alone. They route queries among several models (Ong et al., 2025), have models debate and revise (Du et al., 2023), rank and fuse competing answers (Jiang et al., 2023), or fine-tune a set of models to work together (Subramaniam et al., 2025; Jiang et al., 2025). For a single participant in such a system, the quantity that matters is not how well it does the task on its own, but how much it contributes to the system. Standard evaluation does not capture this, and neither does standard training: typical post-training pipelines (Ouyang et al., 2022; Guo et al., 2025) optimize the same solo correctness that leaderboards reward, so a model's usefulness inside a system is neither measured nor optimized for. We call that quantity a model's *compositional strength*, and this paper asks how to evaluate it and how to train models for it.

Evaluating LLMs by compositional strength. Inspired by cooperative game theory (Shapley et al., 1953), we treat a collaboration method on a task as a game whose coalitions are subsets of a model pool, and define a model's compositional strength as its Shapley value in that game (§2). Computing compositional strength exactly is, however, infeasible at scale: we estimate it by the leave-one-out (LOO) marginal, the drop in system performance when a model is removed, costing only $n + 1$ runs, and tracking the exact value closely (Appendix A.8). If compositional

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

strength is largely independent of individual capability, then existing evaluation and training efforts are inadequate for system-level intelligence.

Individual strength does not predict compositional strength. Our extensive experiments across five model pools, twelve collaboration methods, and twelve datasets give a consistent answer. In 12 of the 15 (pool, method-family) correlation settings, individual and compositional strength bear no statistically significant relationship to each other, and in the three cells that do reach significance the correlation is weak ($|\rho| \leq 0.29$) despite large samples. The decoupling is sharpest exactly where it matters most: among pools of capable general-purpose models no setting shows any significant relationship at all. A model's leaderboard standing thus says little about its value in a system it joins.

Training LLMs for compositional strength. We propose to train models with *compositional reinforcement learning* (C-RL), which trains a policy inside a live collaborative system and rewards it with the system's outcome rather than with its solo correctness (§3). The pool of frozen collaborators and the collaboration protocol together become the environment the policy acts in. We show training for compositional strength using LOO marginals actually doesn't require running an extra coalition that excludes the policy and thus costs no more than ordinary RL with system outcome.

C-RL reliably improves compositional strength. We train Qwen 3.6 27B inside a pool of three frozen collaborators that are larger than the policy (gpt-oss-120b, kimi-k3, and nemotron-3.5-lightning), under three collaboration protocols and eight tasks spanning reasoning, scientific, multilingual, and pluralism. Across the 24 settings, C-RL[1] yields the strongest collaborator in 21 and is the only method compared to five baselines that never leaves the system worse off than the untrained policy does. C-RL lifts collaborative performance by $+4.0$ to $+5.4$ points on average under each protocol, close to three times the gain of the strongest baseline, and by as much as $+13.1$ points on a single task, and RL with solo reward can sometimes make the policy individually better but the system measurably worse. The gains persist against a pool of frontier proprietary models (GPT-5.5, Gemini 3.5 Flash, Claude Opus 4.8) and transfer to datasets and collaboration protocols the policy never trained on, indicating C-RL teaches generalizable collaboration skills rather than fitting one setup. Together, these results chart a promising route toward system-level intelligence, a modular and collaborative AI future where what a model is worth is what it adds to the whole.

2 Evaluating Language Models by Compositional Strength

2.1 Method

Language models are typically evaluated and ranked by their *individual strength*: how well a single model performs a task on its own. Though well suited to solitary deployment, it says nothing about a question that matters increasingly often in practice: when a model is deployed as one component of a larger collaborative system (multi-agent debate, model routing, ensembling, mixture-of-agents, or collaborative fine-tuning), how much does that model actually contribute to the system's success? We call this quantity a model's *compositional strength*, and in this section we formalize it as a cooperative game and derive a tractable estimator for it.

Collaboration as a Cooperative Game. Let $\mathcal{N} = \{1, \ldots, n\}$ denote a pool of candidate models. A *coalition* is any subset $S \subseteq \mathcal{N}$ of models that collaborate together under a fixed collaboration method m (e.g. a routing, debate, or ensembling protocol) to solve a fixed task t. We define the *characteristic function*, the standard cooperative-game object assigning a scalar worth to every coalition:

$$v_{m,t}(S) \;=\; P_{m,t}(S) \;\in\; \mathbb{R}, \tag{1}$$

where $P_{m,t}(S)$ is the performance (accuracy, pass rate, or any other task-specific score) obtained by running collaboration method m on task t using exactly the models in S, with $v_{m,t}(\varnothing)$ defined as a task-specific floor (e.g. zero or chance performance). Since m and t are fixed throughout this construction, we drop the subscripts and write $v(S)$ for the rest of the paper. The pair $(\mathcal{N}, v)$ is then a cooperative game: the worth of a group of models is simply how well the system they form performs. This presumes $v(S)$ depends only on *which* models are in S and not on any ordering among them,

[1] We will release data and code for both the evaluation and training studies upon acceptance.

which is the case for almost all model collaboration algorithms, including every method we study; where a protocol needs an internal ordering, it derives one from the set itself.

Given this game, a model's compositional strength is its *share* of the value the grand coalition $\mathcal{N}$ (the coalition of all models) creates. We use the Shapley value to define that share (see Appendix A.7), and estimate it in practice using LOO marginals.

The Shapley Value. The Shapley value $\phi_i(v)$ is model i's marginal contribution, averaged over all $n!$ orderings in which the grand coalition is assembled one model at a time (more in Appendix A.7):

$$\phi_i(v) \;=\; \sum_{S \subseteq \mathcal{N} \setminus \{i\}} \frac{|S|!\,(n-|S|-1)!}{n!} \Big[v(S \cup \{i\}) - v(S) \Big]. \tag{2}$$

Leave-One-Out as a Shapley Estimator. Evaluating equation 2 exactly requires $v(S)$ for all 2^n subsets of $\mathcal{N}$. When Shapley values are classically applied, the characteristic function is usually cheap, so enumeration is feasible; here a single $v(S)$ is an evaluation of a multi-model system, which puts exhaustive enumeration out of reach. We therefore estimate compositional strength with a *leave-one-out* (LOO) marginal computed against the grand coalition:

$$\hat{\phi}_i(v) \;\triangleq\; v(\mathcal{N}) - v(\mathcal{N} \setminus \{i\}) \;=\; P_{m,t}(\mathcal{N}) - P_{m,t}(\mathcal{N} \setminus \{i\}). \tag{3}$$

The hat distinguishes this estimate from the exact Shapley value $\phi_i(v)$ of equation 2; both are functions of the same game v, and we write $\hat{\phi}_i$ wherever it is clear from context. The compositional strength of model i is the drop in the full system's performance when i is removed: a positive value means i was contributing, a value near zero means i was redundant, and a negative value means i was actively harmful to that particular collaboration.

The LOO marginal is not an unrelated heuristic: up to a factor $1/n$, it is exactly the $S = \mathcal{N} \setminus \{i\}$ term of the Shapley sum equation 2,

$$\phi_i(v) \;=\; \underbrace{\tfrac{1}{n}\,\hat{\phi}_i(v)}_{|S|=n-1 \text{ term}} \;+\; \sum_{S \subsetneq \mathcal{N} \setminus \{i\}} \frac{|S|!\,(n-|S|-1)!}{n!} \Big[v(S \cup \{i\}) - v(S) \Big], \tag{4}$$

We confirm empirically (Appendix A.8) that this estimate correlates with two costlier ones, the Banzhaf value (Banzhaf III, 1964) and exact Shapley.

Computational cost. Computing $\hat{\phi}_i$ for $i \in \mathcal{N}$ via equation 3 requires only the $n+1$ coalition values $\{v(\mathcal{N})\} \cup \{v(\mathcal{N} \setminus \{i\}) : i \in \mathcal{N}\}$, linear in n, versus the 2^n coalitions for exact Shapley. This reduction makes a broad sweep across models, tasks, and collaboration methods feasible.

2.2 Experiment Settings

We instantiate the framework of §2.1 across a broad cross-product of model pools $\mathcal{N}$, tasks t, and collaboration methods m. We summarize each dimension below; full details of each model pool, every dataset, and the configuration of every collaboration method are given in Appendix B.1.

Model pools. We evaluate five model pools spanning a wide range of scale and provenance: *Specialized Expert Models* ($n = 4$), domain-specialized ∼7B models fine-tuned respectively for general instruction-following, mathematics, coding, and long context; *∼20-30B Open Models* ($n = 7$), open-weight models in the tens-of-billions parameter range from several model families; *>100B Open Models* ($n = 4$), larger open-weight models; *Proprietary API Models* ($n = 3$), frontier proprietary systems served over a common API; and *Diverse Academic Models* ($n = 8$), a set of academically-trained models of heterogeneous origin.

Datasets. We evaluate on 12 datasets spanning five task categories: QA, Reasoning, Coding, Pluralism & Safety, and Science; full list, citations, and dataset sizes are in Table 10 (Appendix B.1).

Collaboration methods. We evaluate 12 collaboration methods in three families. *Routing* methods dispatch each query or part of the query to one model from the pool: Prompt Routing (Frick et al., 2025), Trained Router (Ong et al., 2025), Switch Generation (Feng et al., 2025b), and Graph

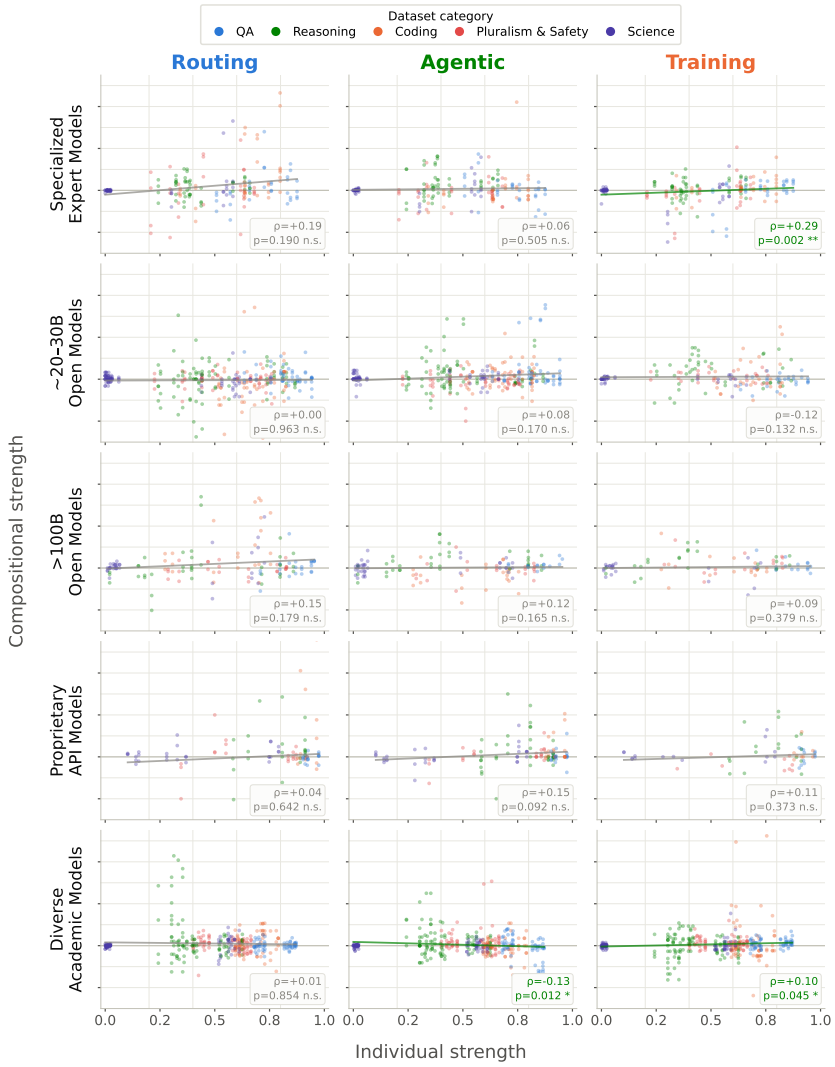

Figure 1: Individual strength (solo score) vs. compositional strength (leave-one-out Shapley, §2.1), by model pool (rows) and collaboration-method family (columns). Each point is one model evaluated on one dataset under one collaboration method; color denotes task category. ρ is Spearman correlation with a permutation-test p-value (10,000 permutations). **Individual and compositional strength are largely uncorrelated**: in 12 of 15 (pool, method) panels the two show no significant relationship, visible from near-flat fitted lines and printed ρ ($|\rho| \leq 0.29$ throughout) and p-value (marked n.s. when $p > 0.05$); the decoupling is sharpest among pools of general-purpose models, namely >100B Open Models and Proprietary Models (rows 3-4), where all six panels are n.s.

Router (Feng et al., 2025c). *Agentic* methods have models interact and exchange text before a final answer is produced: Multiagent Debate (Du et al., 2023), Multiagent Feedback (Feng et al., 2024), Heterogeneous Swarms (Feng et al., 2025a), and SLM-MUX (Wang et al., 2026). *Training* methods train components or the models themselves for collaboration: LLM-Blender (Jiang et al., 2023), AggLM (Zhao et al., 2025a), Multiagent Finetuning (Subramaniam et al., 2025), and Sparta Alignment (Jiang et al., 2025). All methods are implemented within MoCo toolkit (Feng et al., 2026b).

Protocol. For each valid (pool, dataset, method) cell we compute the $n + 1$ coalition values required by equation 3 and derive each model's compositional strength $\hat{\phi}_i$. We additionally measure each model's *individual strength*, its standalone score on the same dataset, by running it alone through the same evaluation pipeline. The result is a large matrix of paired (individual strength, compositional strength) observations, indexed by model, dataset, and collaboration method.

2.3 Results

Collaboration algorithms are effective. We first confirm that collaboration protocols improve performance (mean $+11.5$, 95% CI $[+8.7, +14.7]$; details in Appendix A.4) across diverse (model

pool, task) settings. The question now turns to correlation between individual strength ($v(\{i\})$, model i acting alone) and the level of helpfulness, i.e., compositional strength ($\hat{\phi}_i$, §2.1).

Individual and compositional strength are not correlated. Across the full 5×3 grid of model pools and collaboration-method families (Figure 1), 12 of the 15 cells show no statistically significant relationship between a model's individual strength and its compositional strength. The remaining three cells reach significance only because of the large sample size in each cell; the correlations themselves are weak ($|\rho| \leq 0.29$) and the fitted trend lines are correspondingly flat. A model's standalone benchmark score is therefore mostly uninformative about how much it contributes to the collaborative systems it joins.

The weak signal that does exist favors training-based collaboration. Two of the three significant cells occur in the training-based column (Specialized Expert Models, $\rho = +0.29$; Diverse Academic Models, $\rho = +0.10$), where components or these models themselves are trained to work together. The routing column, in contrast, shows no significant relationship in any of the five pools. The pattern suggests that deeper levels of integration leave slightly more room for individual strength to matter. However, the effect is small even where it narrowly reaches significance, and it is absent from routing and agentic collaboration, which are more popular in day-to-day practices due to their simplicity and lower engineering overhead. For the collaborative systems practitioners are most likely to build, individual strength remains uninformative.

Larger and more capable model pools show the least coupling. The three pools composed of larger or more capable models (∼20-30B Open Models, >100B Open Models, and Proprietary API Models) show *no* significant correlation in any of the nine (pool, method-family) cells they span. All three significant cells instead occur in the two pools of specialization and diversity: Specialized Expert Models (narrowly fine-tuned ∼7B models) and Diverse Academic Models (a heterogeneous, non-frontier set). For capable general-purpose models, the pools practitioners are most likely to deploy, individual benchmarks carry no signal about compositional strength: this suggests that evaluating synergy and complementarity matters most when composing state-of-the-art models (§2.4).

2.4 Analysis

A small number of methods account for most top finishes. Figure 2 ranks each collaboration method by how often it achieves top-3 full-coalition performance across all (pool, dataset) pairs. The distribution is far from uniform: Sparta Alignment (Jiang et al., 2025) and Multiagent Debate (Du et al., 2023) finish #1 most often among the 12 methods, with by far the largest #1 shares of any method, and together with LLM-Blender (Jiang et al., 2023) and AggLM (Zhao et al., 2025a) they account for a disproportionate share of top-3 outcomes.

This complements results in §2.3: while *which model* you add to a coalition is largely uninformative from its solo score, *which collaboration method* you use is highly informative. The agentic methods of collaboration focusing on multiple rounds of inference-time communication therefore strike a nice balance between engineering ease and strong performance, and we recommend them as first choices. Collaboration also helps unevenly across task domains and model pools; see Appendix A.5 (Figure 5) for a full breakdown.

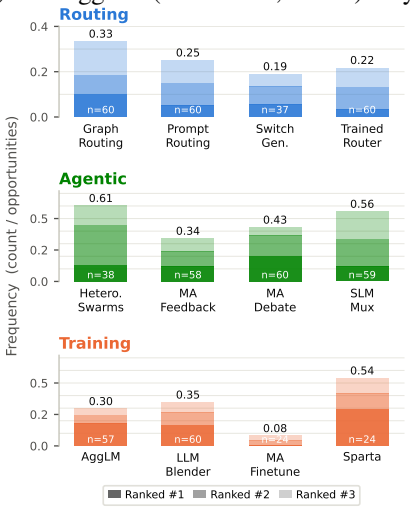

Figure 2: Frequency each method achieves top-3 performance $P_{m,t}(\mathcal{N})$ across (pool, dataset). Ties share a rank.

3 Training for Compositional Strength

3.1 Method

Section 2 establishes that compositional strength is a distinct property mostly uncorrelated with individual strength, the usual focus of existing evaluation and training efforts. We propose to train models

for compositional strength directly rather than hope for it as a byproduct of individual-capability training, through *compositional reinforcement learning* (C-RL).

Objective. We denote a model collaboration method m, a task t, a trainable model as the policy π_θ, and fixed collaborator models $\mathcal{C} = \mathcal{N} \setminus \{\pi_\theta\}$. We use R for the task's reward function and $y_{\text{sys}} \sim m(\cdot \mid q, S)$ as the answer the collaborative system produces on question q from model coalition S, so that system performance $P_{m,t}(S) = \mathbb{E}_{q\sim t,\, y_{\text{sys}}\sim m(\cdot|q,S)}\big[R(q, y_{\text{sys}})\big]$. The pool of collaborators and the model collaboration algorithm m become the *environment* the policy acts in: training a model for compositional strength is thus reinforcement learning grounded in interaction with an environment of other LLMs. Taking equation 3 as the quantity to maximize gives

$$\max_\theta \ \hat{\phi}_{\pi_\theta} \ = \ \max_\theta \Big[P_{m,t}\big(\mathcal{C} \cup \{\pi_\theta\}\big) - P_{m,t}(\mathcal{C})\Big]. \tag{5}$$

To turn equation 5 into a per-sample training signal for policy-gradient methods, the natural approach is to reward each sample by its own LOO marginal,

$$r(q, y_{\text{sys}}) \ = \ R(q, y_{\text{sys}}) - b_q, \qquad b_q \ = \ \mathbb{E}_{y_{\text{sys}}\sim m(\cdot|q,\mathcal{C})}\big[R(q, y_{\text{sys}})\big], \tag{6}$$

subtracting the collaborators-only score b_q on question q from the sample's raw reward. Computed this way, b_q requires an actual extra rollout of $\mathcal{C}$ alone for every question, doubling the number of system rollouts training would need.

We show this extra rollout is unnecessary. Group-relative policy-gradient methods already form the advantage of a sample by subtracting the mean reward over samples drawn for the same question. Letting $y^{(1)}, \ldots, y^{(K)}$ be the system answers from K sampled rollouts on q with the policy plugged in, the advantage of the j-th sample, computed with the LOO-marginal reward $R(q, y^{(j)}) - b_q$, is

$$A^{(j)} \ = \ \Big[R\big(q, y^{(j)}\big) - b_q\Big] \ - \ \frac{1}{K}\sum_{l=1}^{K}\Big[R\big(q, y^{(l)}\big) - b_q\Big] \ = \ R\big(q, y^{(j)}\big) \ - \ \frac{1}{K}\sum_{l=1}^{K} R\big(q, y^{(l)}\big), \tag{7}$$

so b_q cancels out entirely: group-relative advantage recovers exactly the LOO-marginal advantage, without ever running $\mathcal{C}$ alone to compute it. As a result, we can train directly on the system reward

$$\max_\theta \ \mathbb{E}_{q\sim t,\, y_{\text{sys}}\sim m(\cdot|q,\, \mathcal{C}\cup\{\pi_\theta\})}\big[\, R\big(q, y_{\text{sys}}\big)\,\big], \tag{8}$$

the expectation running over both the question draw and the policy and collaborators' stochastic generations, under a collaboration protocol m. Unlike a general Shapley or Banzhaf estimator, which samples many coalitions $S \subseteq \mathcal{N}$, C-RL always evaluates the single grand coalition $\mathcal{C} \cup \{\pi_\theta\}$ — the leave-one-out simplification of §2.1 — keeping every rollout cheap. We can pin both $\mathcal{C}$ and t to train a specialist policy, or unfreeze either/both by sampling collaborator sets/methods from a larger population or drawing tasks from a distribution.

The reward is the system's output, not the policy's own. The C-RL objective indicates that the policy is never rewarded for being individually correct in solving the whole problem, but for how much its participation moves y_{sys} and improves system performance. A rollout in which the policy's own answer is wrong can still carry full reward if the system answered correctly, and a rollout in which the policy is right can carry zero reward if the system was wrong (e.g., not explaining the reasoning clear enough for other models to trust). This is the same distinction Section 2 shows individual benchmarks cannot see, now formalized as a training signal in C-RL.

Credit assignment across roles and turns. A collaboration protocol may query the policy several times per question, and in several distinct roles. Let $\tau_1, \ldots, \tau_L$ denote the policy's generations within one execution of the protocol on question q, where L is itself a random quantity: a protocol may invoke the policy a different number of times from one execution to the next, depending on routing decisions, early termination, or degenerate outputs. Since R scores only the final system output, we assign the same scalar to every generation in that execution,

$$r(\tau_\ell) \ = \ R\big(q, y_{\text{sys}}\big) \quad \text{for all } \ell = 1, \ldots, L, \tag{9}$$

so that each generation is credited with the system-level outcome it helped produce. We leave differentiated credit assignment to future work; uniform assignment already works well empirically.

Group-relative advantage over variable-length rollouts. We optimize equation 8 with a group-relative policy-gradient method (Shao et al., 2024), which estimates the advantage of a generation by comparing its reward against a baseline formed from other samples for the same question, avoiding a learned value model. Sampling K independent rollouts of the protocol for each of N questions per step, and denote $\mathcal{T}_q$ as the set of all policy generations produced across the K rollouts for question q, the advantage of generation $\tau \in \mathcal{T}_q$ is

$$A(\tau) \;=\; \frac{r(\tau) - \mu_q}{\sigma_q}, \qquad \mu_q = \frac{1}{|\mathcal{T}_q|} \sum_{\tau' \in \mathcal{T}_q} r(\tau'), \qquad \sigma_q = \mathrm{std}\big(\{r(\tau')\}_{\tau' \in \mathcal{T}_q}\big), \tag{10}$$

this is the mean-subtraction that produced the cancellation in equation 7. The one difference from standard GRPO is that the group is formed *per question*, over the pooled generations of the policy model from that question's K rollouts, rather than over a fixed number of samples per prompt. Because L, how many times the policy is called in a rollout, varies from one rollout to the next, $|\mathcal{T}_q|$ is not a fixed multiple of K, so the group size won't be fixed. The policy is then updated with the usual clipped policy-gradient objective (Schulman et al., 2017) using $A(\tau)$, with the loss applied over the response tokens of each generation of the policy model; Appendix B.2 details how μ_q, σ_q interact with the baseline cancellation and how variable group sizes affect the loss.

Practical requirements. Two things set this apart from standard RL post-training. First, the environment is a whole multi-model collaborative system containing participants the trainer does not own, including models larger than the policy and closed-weight APIs; none of equation 5 through equation 10 requires differentiating through it, only the scalar $R(q, y_{\mathrm{sys}})$ crosses the policy-environment boundary. Second, a rollout mixes the policy's own generations with collaborators' contributions; training on collaborator text would teacher-force the policy off-policy, so we attribute each generation to the model that produced it and apply the loss only to the policy's own tokens under their own sampling distribution. Appendix B.2 describes how this attribution is implemented.

3.2 Experiment Settings

Policy and collaborators. The trainable policy is Qwen 3.6 27B, full fine-tuning. Its collaborators are three fixed models, all larger than the policy model: gpt-oss-120b, kimi-k3, and nemotron-3.5-lightning, none of which is updated during training. Holding them frozen is what makes the baseline cancellation in equation 7 apply. We vary the model settings in §3.4.

Collaboration environments. We train under three collaboration methods, chosen to span distinct modes of composition. LLM-Blender (Jiang et al., 2023) has every model produce a candidate answer, a ranker score the candidates, and a fuser synthesize the top-ranked ones into the system's answer; the policy is credited under equation 9 for diverse roles. Multiagent Debate (Du et al., 2023) has models answer independently and then revise over successive rounds based on each other's responses. SLM-MUX (Wang et al., 2026) selects and runs several models in parallel and produces a final answer based on their outputs. Each is a different collaborative environment (§3.1); §3.4 explores generalization to held-out algorithms.

Tasks. We evaluate on eight tasks spanning QA, reasoning, and science: GPQA-Diamond (Rein et al., 2024), graduate-level science questions; TheoremQA (Chen et al., 2023), theorem-driven mathematical reasoning; MMLU-ProX (Xuan et al., 2025), multilingual multiple-choice questions across 20 languages; CulturalBench-Hard (Chiu et al., 2024), cultural reasoning and knowledge; ChemBench (Mirza et al., 2025), chemistry questions across nine subfields; KernelBench (Ouyang et al., 2025), rewriting PyTorch operators as faster custom CUDA kernels; SMDD-Bench (Han et al., 2026), small-molecule drug design; and AssayBench (Brouwer et al., 2026), CRISPR genetic-screen gene ranking. §3.4 tests generalization to held-out data.

Baselines. *Base* is the untrained policy. *Individual-reward RL* applies GRPO (Shao et al., 2024) and DAPO (Yu et al., 2026) to the policy's own solo correctness. *Collaboration-aware baselines*: *prompt nudge* leaves the policy untrained and instead informs it in its system prompt how the collaborative system it is in works, and *collaborative RAFT* (Dong et al., 2023; Feng et al., 2026d) runs the real collaboration pipeline with the untrained policy, keeps the transcripts whose collaborative outcome scored well, and fine-tunes on them.

Method	GPQA-dia	Theorem	KernelB	SMDD	AssayB	ChemB	Culture-H	MMLU-PX	Avg.
Collaboration protocol: LLM-Blender									
Base	.818	.613	.144	.635	.035	.720	.806	.804	.572
GRPO-indiv	.849	.647	.157	.635	.033	.732	.825	.822	.587
DAPO-indiv	.859	.657	.129	.635	.041	.750	.819	.812	.588
prompt-collab	.838	.630	.159	.654	.036	.742	.822	.827	.588
RAFT-collab	.808	.645	.147	.652	.031	.708	.827	.796	.577
C-RL (ours)	**.879**	**.708**	**.183**	**.686**	**.046**	**.776**	**.843**	**.840**	**.620**
Collaboration protocol: Multiagent Debate									
Base	.768	.623	.155	.655	.036	.720	.817	.818	.574
GRPO-indiv	.869	**.670**	.177	.686	.038	.732	.833	.822	.603
DAPO-indiv	.879	.640	.172	**.689**	.038	.740	.833	.832	.603
prompt-collab	.879	.645	.139	.670	.037	.744	.810	.838	.595
RAFT-collab	.849	.655	.155	.629	.038	.748	.822	.814	.589
C-RL (ours)	**.899**	.655	**.194**	.664	**.044**	**.764**	**.845**	**.846**	**.614**
Collaboration protocol: SLM-MUX									
Base	.838	.568	.112	.721	.079	.718	.845	.856	.592
GRPO-indiv	.859	.588	.173	.721	.078	.732	.845	.830	.603
DAPO-indiv	.869	.650	.112	.750	.091	.736	.855	.820	.610
prompt-collab	.828	.615	.128	.714	**.109**	.740	.850	.866	.606
RAFT-collab	.838	.542	.148	.661	.095	.720	.848	.800	.582
C-RL (ours)	**.879**	**.672**	**.201**	**.793**	.109	**.766**	**.873**	**.874**	**.646**

Table 1: **Collaborative performance**: each policy is evaluated inside the full collaborative system. Best per column within each collaboration protocol in **bold**, second-best underlined; tied values share the same mark. *Avg.* is the mean over all eight datasets. Confidence intervals are reported in Table 4; individual-mode scores in Appendix A.1.

Evaluation. Every method is scored in two modes on the test set: *individually*, with the trained policy model answering alone, and *collaboratively*, with the trained policy inside the full collaborative environment. Checkpoints are selected on a held-out development split scored during training.

3.3 Results

C-RL produces the strongest collaborator under every protocol. Table 1 reports collaborative performance for six methods across three collaboration protocols. C-RL achieves the best score in 21 of the 24 cells, and it is the only method that never degrades collaborative performance relative to the untrained policy: it improves on *Base* in all 24 cells, whereas each of the five baselines regresses somewhere. Averaged over the eight datasets, C-RL adds $+4.8$ pts over *Base* under LLM-Blender, $+4.0$ pts under Multiagent Debate, and $+5.4$ pts under SLM-MUX, and it exceeds the best baseline in each block by $+3.2$ pts, $+1.1$ pts, and $+3.6$ pts respectively. The fact that margins hold across three protocols that compose models in quite different ways indicates that the policy is not exploiting one particular mechanism, but is picking up generalizable collaboration skills and compositional strength through C-RL training. We study generalization to held-out data and collaborations in §3.4.

Individual-reward RL can improve a model while making it a worse collaborator. Optimizing solo correctness does usually help for collaboration: across the 24 cells, GRPO-indiv improves on *Base* in 19 and DAPO-indiv in 21. However, the exceptions show that a policy could get individually *better* and collaboratively *worse* under solo RL, and the two effects can be badly out of proportion. Together with the individual performance in Appendix A.1, DAPO-indiv gains $+6.2$ pts alone on MMLU-ProX yet costs the system 3.6 pts under SLM-MUX, and GRPO-indiv gains $+2.6$ pts alone while costing the system 2.6 pts. C-RL never exhibits this: it improves on *Base* in all 24 cells. Optimizing the collaborative outcome directly therefore avoids a failure mode that optimizing solo correctness leaves open. Furthermore, C-RL also improves individual performance in most settings and by a wide margin in places, as a by-product rather than as the objective; Appendix A.1 reports individual scores showing that it improves on *Base* in 21 of 24 cells, by $+5.3$ pts on average.

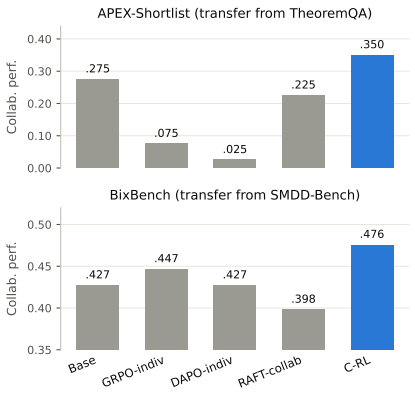

Figure 3: Collaboration performance on two held-out datasets, APEX-Shortlist and BixBench. Prompt-collab is not applicable because it performs no training and produces no checkpoint, so it has nothing to transfer.

The gain concentrates on execution-verified and reasoning tasks. The size of C-RL's improvement varies by dataset. Relative to *Base* and averaged over the protocols where both are available,

the largest relative gain by far is on KernelBench (+5.6 pts absolute, +41% relative), followed by AssayBench (+1.6 pts absolute, +33% relative) and TheoremQA (+7.7 pts absolute, +13% relative). The smallest are MMLU-ProX (+2.7 pts absolute, +3% relative) and CulturalBench-Hard (+3.1 pts absolute, +4% relative). Grouping by how the reward is computed sharpens the pattern: on the three execution-verified tasks the average relative gain is +27.0%, against +7.4% on the five answer-matching tasks. Two properties plausibly account for this. They ask for open-ended artifacts, a CUDA kernel, a molecule, a ranked gene list, which are scored on a graded scale rather than marked right or wrong, so the reward in equation 8 can register partial improvements that a binary match would discard; and they start from low absolute scores, leaving many problems the system still gets wrong, which is where a stronger policy has the most room to help.

3.4 Analysis

Compositional strength transfers to unseen datasets and collaboration protocols, while individual-reward RL does not. Section 3.3 evaluates the policy only on its training dataset and protocol, leaving open whether C-RL teaches something general about collaborating. We evaluate checkpoints, without further training, on two held-out datasets (MathArena APEX-Shortlist (Dekoninck et al., 2026), BixBench (Mitchener et al., 2025); from TheoremQA/SMDD-Bench checkpoints under LLM-Blender) and two held-out protocols (Mixture-of-Agents (Wang et al., 2025b), Heterogeneous Swarms (Feng et al., 2025a); from KernelBench/AssayBench checkpoints under LLM-Blender). Figures 3 and 4 show C-RL is the only method that improves on *Base* in every held-out cell, by up to +9.6 pts, while baselines are erratic: GRPO-indiv and DAPO-indiv fall as much as 25.0 pts below *Base* on APEX-Shortlist and 5.2 pts on KernelBench under Heterogeneous Swarms, and collaborative RAFT collapses to 0.011 on AssayBench under the same protocol. This indicates C-RL teaches generalizable collaboration skills rather than fitting the policy to its training setup.

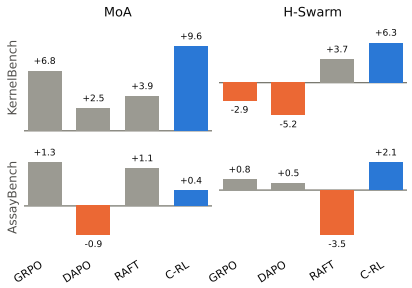

Figure 4: Change in collaborative performance relative to *Base* under two held-out protocols, Mixture-of-Agents (MoA) and Heterogeneous Swarms (H-Swarm). Orange marks a method worse than *Base*; Prompt-collab is omitted (no checkpoint to transfer).

C-RL empowers small policy to collaborate with state-of-the-art proprietary LLMs. We replace the three collaborators with frontier proprietary systems, GPT-5.5, Gemini 3.5 Flash and Claude Opus 4.8, retrain under LLM-Blender, and evaluate on KernelBench and AssayBench. The stronger pool raises the ceiling: *Base* scores 0.174/0.044 here versus 0.144/0.035 with the original collaborators. Table 2 shows C-RL still adds to it more than any baseline, reaching 0.254 on KernelBench (+8.0 pts) and 0.055 on AssayBench (+1.1 pts). AssayBench separates the methods most sharply: every baseline leaves the system 0.2–1.9 pts worse than *Base*, and only C-RL improves on it. A policy small enough to train cheaply can thus help a system built from models far larger than itself, provided it is trained for what it contributes rather than what it can do alone.

Method	KernelBench		AssayBench	
	score	Δ	score	Δ
Base	.174	–	.044	–
GRPO-indiv	.185	+1.1	.033	−1.1
DAPO-indiv	.223	+4.9	.026	−1.9
prompt-collab	.171	−0.3	.043	−0.2
RAFT-collab	.215	+4.1	.032	−1.2
C-RL (ours)	**.254**	+8.0	**.055**	+1.1

Table 2: **Frontier collaborators.** Collaborative performance with GPT-5.5, Gemini 3.5 Flash and Claude Opus 4.8 as the pool, under LLM-Blender. Δ is the change in percentage points relative to the untrained *Base* policy in the same pool. Best per dataset in **bold**.

4 Conclusion

We propose to evaluate and train language models by their *compositional strength*, formalized as a Shapley value and estimated with a leave-one-out marginal. Compositional strength is largely uncorrelated with individual strength, so standard benchmarks say little about a model's value in collaborative systems; we therefore optimize it directly with *compositional reinforcement learning* (C-RL), which produces the strongest collaborator in 21 of 24 settings and generalizes to held-out data, protocols, and stronger collaborators. We see this as a step toward system-level intelligence, where models are trained for what they make possible together, not for what they can do alone.

AI USE STATEMENT

We used generative AI tools to help with language polishing. The AI tools were used solely for language editing and did not contribute to the intellectual content of the work. We have reviewed all AI-assisted edits and take responsibility for the final content of this work.

REPRODUCIBILITY STATEMENT

Model pools, collaboration methods, and datasets for the evaluation study are listed in §2.2, with dataset splits and pool composition in Appendix B.1; the policy, collaborators, protocols, tasks, and baselines for the training study are listed in §3.2, with hyperparameters, implementation, and checkpoint selection in Appendix B.2. Confidence intervals are in Appendix A.2. Two caveats limit exact reproduction: several collaborators are proprietary models served behind APIs whose updates we do not control, so numbers involving them may shift as APIs change. We will release code for both the evaluation and training studies, together with the configurations needed to reproduce every reported run, upon acceptance.

ETHICS STATEMENT

C-RL's objective, rewarding a policy for its contribution to a collaborative system's outcome, has dual-use potential: the same training signal could let a malicious model more effectively inject harmful intent into a multi-model system while remaining individually inconspicuous, compounding the risks studied in adversarial collaboration settings (Yang et al., 2026). Enhancing the safety of collaborative systems against such malicious contributions is an important direction for future work.

REFERENCES

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, et al. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*, 2021.

John F Banzhaf III. Weighted voting doesn't work: A mathematical analysis. *Rutgers L. Rev.*, 19: 317, 1964.

Faeze Brahman, Sachin Kumar, Vidhisha Balachandran, Pradeep Dasigi, Valentina Pyatkin, Abhilasha Ravichander, Sarah Wiegreffe, Nouha Dziri, Khyathi Chandu, Jack Hessel, et al. The art of saying no: Contextual noncompliance in language models. *Advances in Neural Information Processing Systems*, 37:49706–49748, 2024.

Edward De Brouwer, Carl Edwards, Alexander Wu, Jenna Collier, Graham Heimberg, Xiner Li, Meena Subramaniam, Ehsan Hajiramezanali, David Richmond, Jan-Christian Hütter, Sara Mostafavi, and Gabriele Scalia. Assaybench: An assay-level virtual cell benchmark for llms and agents, 2026.

Mert Cemri, Melissa Z Pan, Shuyi Yang, Lakshya A Agrawal, Bhavya Chopra, Rishabh Tiwari, Kurt Keutzer, Aditya Parameswaran, Dan Klein, Kannan Ramchandran, et al. Why do multi-agent llm systems fail? *Advances in Neural Information Processing Systems*, 38, 2026.

Justin Chen, Sukwon Yun, Elias Stengel-Eskin, Tianlong Chen, and Mohit Bansal. Symbolic mixture-of-experts: Adaptive skill-based routing for heterogeneous reasoning. In *International Conference on Machine Learning*, 2026.

Lingjiao Chen, Matei Zaharia, and James Zou. Frugalgpt: How to use large language models while reducing cost and improving performance. *Transactions on Machine Learning Research*, 2024a.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex

Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. Evaluating large language models trained on code. 2021.

Weize Chen, Yusheng Su, Jingwei Zuo, Cheng Yang, Chenfei Yuan, Chi-Min Chan, Heyang Yu, Yaxi Lu, Yi-Hsin Hung, Chen Qian, et al. Agentverse: Facilitating multi-agent collaboration and exploring emergent behaviors. In *International Conference on Learning Representations*, volume 2024, pp. 20094–20136, 2024b.

Wenhu Chen, Ming Yin, Max Ku, Pan Lu, Yixin Wan, Xueguang Ma, Jianyu Xu, Xinyi Wang, and Tony Xia. Theoremqa: A theorem-driven question answering dataset. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 7889–7901, 2023.

Wei-Lin Chiang, Lianmin Zheng, Ying Sheng, Anastasios Nikolas Angelopoulos, Tianle Li, Dacheng Li, Banghua Zhu, Hao Zhang, Michael Jordan, Joseph E. Gonzalez, and Ion Stoica. Chatbot arena: An open platform for evaluating llms by human preference. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 8359–8388. PMLR, 2024.

Yu Ying Chiu, Liwei Jiang, Bill Yuchen Lin, Chan Young Park, Shuyue Stella Li, Sahithya Ravi, Mehar Bhatia, Maria Antoniak, Yulia Tsvetkov, Vered Shwartz, et al. Culturalbench: A robust, diverse, and challenging cultural benchmark by human-ai culturalteaming. *arXiv preprint arXiv:2410.02677*, 2024.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018.

Jasper Dekoninck, Nikola Jovanović, Tim Gehrunger, Kári Rögnvaldsson, Ivo Petrov, Chenhao Sun, and Martin Vechev. Beyond benchmarks: Matharena as an evaluation platform for mathematics with llms. 2026.

Dujian Ding, Ankur Mallick, Shaokun Zhang, Chi Wang, Daniel Madrigal, Mirian Del Carmen Hipolito Garcia, Menglin Xia, Laks VS Lakshmanan, Qingyun Wu, and Victor Rühle. Bestroute: Adaptive llm routing with test-time optimal compute. In *Forty-second International Conference on Machine Learning*, 2025.

Hanze Dong, Wei Xiong, Deepanshu Goyal, Yihan Zhang, Winnie Chow, Rui Pan, Shizhe Diao, Jipeng Zhang, KaShun SHUM, and Tong Zhang. RAFT: Reward ranked finetuning for generative foundation model alignment. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856.

Yilun Du and Leslie Pack Kaelbling. Position: Compositional generative modeling: A single model is not all you need. In *ICML*, pp. 11721–11732, 2024.

Yilun Du, Shuang Li, Antonio Torralba, Joshua B Tenenbaum, and Igor Mordatch. Improving factuality and reasoning in language models through multiagent debate. In *Forty-first International Conference on Machine Learning*, 2023.

Andrew Estornell, Jean-Francois Ton, Muhammad Faaiz Taufiq, and Hang Li. How to train a leader: Hierarchical reasoning in multi-agent llms. *arXiv preprint arXiv:2507.08960*, 2025a.

Andrew Estornell, Jean-Francois Ton, Yuanshun Yao, and Yang Liu. Acc-collab: An actor-critic approach to multi-agent llm collaboration. In *International Conference on Learning Representations*, volume 2025, pp. 90558–90581, 2025b.

Lang Feng, Zhenghai Xue, Tingcong Liu, and Bo An. Group-in-group policy optimization for llm agent training. *Advances in Neural Information Processing Systems*, 38:46375–46408, 2026a.

Shangbin Feng, Weijia Shi, Yike Wang, Wenxuan Ding, Vidhisha Balachandran, and Yulia Tsvetkov. Don't hallucinate, abstain: Identifying llm knowledge gaps via multi-llm collaboration. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 14664–14690, 2024.

Shangbin Feng, Zifeng Wang, Palash Goyal, Yike Wang, Weijia Shi, Huang Xia, Hamid Palangi, Luke Zettlemoyer, Yulia Tsvetkov, Chen-Yu Lee, et al. Heterogeneous swarms: Jointly optimizing model roles and weights for multi-llm systems. *arXiv preprint arXiv:2502.04510*, 2025a.

Shangbin Feng, Wenhao Yu, Yike Wang, Hongming Zhang, Yulia Tsvetkov, and Dong Yu. Don't throw away your pretrained model. *arXiv preprint arXiv:2510.09913*, 2025b.

Shangbin Feng, Yuyang Bai, Ziyuan Yang, Yike Wang, Zhaoxuan Tan, Jiajie Yan, Zhenyu Lei, Wenxuan Ding, Weijia Shi, Haojin Wang, et al. Moco: A one-stop shop for model collaboration research. *arXiv preprint arXiv:2601.21257*, 2026b.

Shangbin Feng, Wenxuan Ding, Alisa Liu, Zifeng Wang, Weijia Shi, Yike Wang, Shannon Zejiang Shen, Xiaochuang Han, Hunter Lang, Chen-Yu Lee, et al. When one llm drools, multi-llm collaboration rules. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 17048–17063, 2026c.

Shangbin Feng, Kishan Panaganti, Yulia Tsvetkov, and Wenhao Yu. The single-multi evolution loop for self-improving model collaboration systems. *arXiv preprint arXiv:2602.05182*, 2026d.

Shangbin Feng, Yike Wang, Weijia Shi, Luke Zettlemoyer, Yejin Choi, and Yulia Tsvetkov. Scaling participation in modular ai systems, 2026e.

Tao Feng, Yanzhen Shen, and Jiaxuan You. Graphrouter: A graph-based router for llm selections. In *The Thirteenth International Conference on Learning Representations*, 2025c.

Evan Frick, Connor Chen, Joseph Tennyson, Tianle Li, Wei-Lin Chiang, Anastasios N Angelopoulos, and Ion Stoica. Prompt-to-leaderboard. *arXiv preprint arXiv:2502.14855*, 2025.

Yuqian Fu, Yuanheng Zhu, Jiajun Chai, Guojun Yin, Wei Lin, Qichao Zhang, and Dongbin Zhao. Rlae: Reinforcement learning-assisted ensemble for llms. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 13463–13477, 2025.

Aryo Pradipta Gema, Joshua Ong Jun Leang, Giwon Hong, Alessio Devoto, Alberto Carlo Maria Mancino, Rohit Saxena, Xuanli He, Yu Zhao, Xiaotang Du, Mohammad Reza Ghasemi Madani, et al. Are we done with mmlu? In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 5069–5096, 2025.

Amirata Ghorbani and James Zou. Data shapley: Equitable valuation of data for machine learning. In *International conference on machine learning*, pp. 2242–2251. PMLR, 2019.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Kevin Han, Renfei Zhang, Kathy Wei, Hamed Mahdavi, Niloofar Mireshghallah, and Amir Farimani. Smdd-bench: Can llms solve real-world small molecule drug design tasks?, 2026.

Miriam Horovicz and Roni Goldshmidt. Tokenshap: Interpreting large language models with monte carlo shapley value estimation. In *Proceedings of the 1st Workshop on NLP for Science (NLP4Science)*, pp. 1–8, 2024.

Yun Hua, Haosheng Chen, Shiqin Wang, Wenhao Li, Xiangfeng Wang, and Jun Luo. Shapley-coop: Credit assignment for emergent cooperation in self-interested llm agents. *Advances in Neural Information Processing Systems*, 38:88675–88702, 2026.

Dongfu Jiang, Xiang Ren, and Bill Yuchen Lin. Llm-blender: Ensembling large language models with pairwise ranking and generative fusion. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 14165–14178, 2023.

Yuru Jiang, Wenxuan Ding, Shangbin Feng, Greg Durrett, and Yulia Tsvetkov. Sparta alignment: Collectively aligning multiple language models through combat. *arXiv preprint arXiv:2506.04721*, 2025.

Kishor Jothimurugan, Suguman Bansal, Osbert Bastani, and Rajeev Alur. Compositional reinforcement learning from logical specifications. *Advances in Neural Information Processing Systems*, 34:10026–10039, 2021.

Sayash Kapoor, Benedikt Stroebl, Zachary S Siegel, Nitya Nadgir, and Arvind Narayanan. Ai agents that matter. *Transactions on Machine Learning Research*, 2024.

Sayash Kapoor, Benedikt Stroebl, Peter Kirgis, Nitya Nadgir, Zachary Siegel, Boyi Wei, Tianci Xue, Ziru Chen, Felix Chen, Saiteja Utpala, et al. Holistic agent leaderboard: The missing infrastructure for ai agent evaluation. In *International Conference on Learning Representations*, volume 2026, pp. 98778–98849, 2026.

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

Yongchan Kwon and James Zou. Beta shapley: a unified and noise-reduced data valuation framework for machine learning. In Gustau Camps-Valls, Francisco J. R. Ruiz, and Isabel Valera (eds.), *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*, volume 151 of *Proceedings of Machine Learning Research*, pp. 8780–8802. PMLR, 28–30 Mar 2022.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, et al. Tulu 3: Pushing frontiers in open language model post-training. *arXiv preprint arXiv:2411.15124*, 2024.

Mina Lee, Megha Srivastava, Amelia Hardy, John Thickstun, Esin Durmus, Ashwin Paranjape, Ines Gerard-Ursin, Xiang Lisa Li, Faisal Ladhak, Frieda Rong, et al. Evaluating human-language model interaction. *Transactions on Machine Learning Research*, 2023.

Weizhen Li, Jianbo Lin, Zhuosong Jiang, Jingyi Cao, Xinpeng Liu, Jiayu Zhang, Zhenqiang Huang, Qianben Chen, Weichen Sun, Qiexiang Wang, et al. Chain-of-agents: End-to-end agent foundation models via multi-agent distillation and agentic rl. *arXiv preprint arXiv:2508.13167*, 2025.

Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, Benjamin Newman, Binhang Yuan, Bobby Yan, Ce Zhang, Christian Alexander Cosgrove, Christopher D Manning, Christopher Re, Diana Acosta-Navas, Drew Arad Hudson, Eric Zelikman, Esin Durmus, Faisal Ladhak, Frieda Rong, Hongyu Ren, Huaxiu Yao, Jue WANG, Keshav Santhanam, Laurel Orr, Lucia Zheng, Mert Yuksekgonul, Mirac Suzgun, Nathan Kim, Neel Guha, Niladri S. Chatterji, Omar Khattab, Peter Henderson, Qian Huang, Ryan Andrew Chi, Sang Michael Xie, Shibani Santurkar, Surya Ganguli, Tatsunori Hashimoto, Thomas Icard, Tianyi Zhang, Vishrav Chaudhary, William Wang, Xuechen Li, Yifan Mai, Yuhui Zhang, and Yuta Koreeda. Holistic evaluation of language models. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856. Featured Certification, Expert Certification.

Bo Liu, Simon Yu, Zichen Liu, Leon Guertler, Penghui Qi, Daniel Balcells, Mickel Liu, Cheston Tan, Weiyan Shi, Min Lin, et al. Spiral: Self-play on zero-sum games incentivizes reasoning via multi-agent multi-turn reinforcement learning. In *International Conference on Learning Representations*, volume 2026, pp. 35407–35434, 2026a.

Shuo Liu, Zeyu Liang, Xueguang Lyu, and Christopher Amato. Llm collaboration with multi-agent reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 32150–32158, 2026b.

Mingyu Lu, Yushan Huang, Chris Lin, and Su-In Lee. Agents that matter: Optimizing multi-agent llms via removal-based attribution. *arXiv preprint arXiv:2605.27621*, 2026.

Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30, 2017.

Hao Ma, Tianyi Hu, Zhiqiang Pu, Boyin Liu, Xiaolin Ai, Yanyan Liang, and Min Chen. Coevolving with the other you: Fine-tuning llm with sequential cooperative multi-agent reinforcement learning. *Advances in Neural Information Processing Systems*, 37:15497–15525, 2024.

Adrian Mirza, Nawaf Alampara, Sreekanth Kunchapu, Martiño Ríos-García, Benedict Emoekabu, Aswanth Krishnan, Tanya Gupta, Mara Schilling-Wilhelmi, Macjonathan Okereke, Anagha Aneesh, Mehrdad Asgari, Juliane Eberhardt, Amir Mohammad Elahi, Hani M. Elbeheiry, María Victoria Gil, Christina Glaubitz, Maximilian Greiner, Caroline T. Holick, Tim Hoffmann, Abdelrahman Ibrahim, Lea C. Klepsch, Yannik Köster, Fabian Alexander Kreth, Jakob Meyer, Santiago Miret, Jan Matthias Peschel, Michael Ringleb, Nicole C. Roesner, Johanna Schreiber, Ulrich S. Schubert, Leanne M. Stafast, A. D. Dinga Wonanke, Michael Pieler, Philippe Schwaller, and Kevin Maik Jablonka. A framework for evaluating the chemical knowledge and reasoning abilities of large language models against the expertise of chemists. *Nature Chemistry*, 17(7): 1027–1034, May 2025. ISSN 1755-4349. doi: 10.1038/s41557-025-01815-x.

Ludovico Mitchener, Jon M Laurent, Alex Andonian, Benjamin Tenmann, Siddharth Narayanan, Geemi P Wellawatte, Andrew White, Lorenzo Sani, and Samuel G Rodriques. Bixbench: a comprehensive benchmark for llm-based agents in computational biology, 2025.

Sumeet Ramesh Motwani, Chandler Smith, Rocktim Jyoti Das, Rafael Rafailov, Ivan Laptev, Philip HS Torr, Fabio Pizzati, Ronald Clark, and Christian Schroeder de Witt. Malt: Improving reasoning with multi-agent llm training. In *Conference on Language Modeling*, 2025.

Isaac Ong, Amjad Almahairi, Vincent Wu, Wei-Lin Chiang, Tianhao Wu, Joseph E Gonzalez, Mohammed Kadous, and Ion Stoica. Routellm: Learning to route llms from preference data. In *International Conference on Learning Representations*, volume 2025, pp. 34433–34448, 2025.

Anne Ouyang, Simon Guo, Simran Arora, Alex L Zhang, William Hu, Christopher Ré, and Azalia Mirhoseini. Kernelbench: Can llms write efficient gpu kernels? In *Forty-second International Conference on Machine Learning*, 2025.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35: 27730–27744, 2022.

Aneesh Pappu, Batu El, Hancheng Cao, Carmelo di Nolfo, Yanchao Sun, Meng Cao, and James Zou. Multi-agent teams hold experts back. In *International Conference on Machine Learning*, volume 2026, 2026.

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

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741, 2023.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R Bowman. Gpqa: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*, 2024.

Benedek Rozemberczki, Lauren Watson, Péter Bayer, Hao-Tsung Yang, Olivér Kiss, Sebastian Nilsson, and Rik Sarkar. The shapley value in machine learning. In Lud De Raedt (ed.), *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, pp. 5572–5579. International Joint Conferences on Artificial Intelligence Organization, 7 2022. doi: 10.24963/ijcai.2022/778. Survey Track.

Huascar Sanchez and Briland Hitaj. Llm chemistry estimation for multi-llm recommendation. *arXiv preprint arXiv:2510.03930*, 2025.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Lloyd S Shapley et al. A value for n-person games. 1953.

Vighnesh Subramaniam, Yilun Du, Joshua B Tenenbaum, Antonio Torralba, Shuang Li, and Igor Mordatch. Multiagent finetuning: Self improvement with diverse reasoning chains. In *The Thirteenth International Conference on Learning Representations*, 2025.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc Le, Ed Chi, Denny Zhou, et al. Challenging big-bench tasks and whether chain-of-thought can solve them. In *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 13003–13051, 2023.

David Wadden, Kejian Shi, Jacob Morrison, Alan Li, Aakanksha Naik, Shruti Singh, Nitzan Barzilay, Kyle Lo, Tom Hope, Luca Soldaini, et al. Sciriff: A resource to enhance language model instruction-following over scientific literature. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 6083–6120, 2025.

Chenyu Wang, Zishen Wan, Hao Kang, Emma Chen, Zhiqiang Xie, Tushar Krishna, Vijay Janapa Reddi, and Yilun Du. Slm-mux: Orchestrating small language models for reasoning. In *International Conference on Learning Representations*, volume 2026, pp. 73697–73723, 2026.

Jiachen T Wang and Ruoxi Jia. Data banzhaf: A robust data valuation framework for machine learning. In *International conference on artificial intelligence and statistics*, pp. 6388–6421. PMLR, 2023.

Jiachen Tianhao Wang, Prateek Mittal, Dawn Song, and Ruoxi Jia. Data shapley in one training run. In *International conference on learning representations*, volume 2025, pp. 12358–12395, 2025a.

Jingtan Wang, Xiaoqiang Lin, Rui Qiao, Chuan-Sheng Foo, and Bryan Kian Hsiang Low. Helpful or harmful data? fine-tuning-free shapley attribution for explaining language model predictions. In *Proc. ICML*, 2024.

Junlin Wang, Jue Wang, Ben Athiwaratkun, Ce Zhang, and James Y Zou. Mixture-of-agents enhances large language model capabilities. In *International Conference on Learning Representations*, volume 2025, pp. 33944–33963, 2025b.

Xin Xu, Clive Bai, Kai Yang, Tianhao Chen, Yangkun Chen, Weijie Liu, Hao Chen, Yang Wang, Saiyong Yang, and Can Yang. Composition-rl: Compose your verifiable prompts for reinforcement learning of large language models. *arXiv preprint arXiv:2602.12036*, 2026.

Weihao Xuan, Rui Yang, Heli Qi, Qingcheng Zeng, Yunze Xiao, Aosong Feng, Dairui Liu, Yun Xing, Junjue Wang, Fan Gao, et al. Mmlu-prox: A multilingual benchmark for advanced large language model evaluation. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 1513–1532, 2025.

Yingxuan Yang, Bo Huang, Siyuan Qi, Chao Feng, Haoyi Hu, Yuxuan Zhu, Jinbo Hu, Haoran Zhao, Ziyi He, Xiao Liu, et al. Understanding and optimizing agentic workflows via shapley value. *arXiv preprint arXiv:2502.00510*, 2025.

Ziyuan Yang, Wenxuan Ding, Shangbin Feng, and Yulia Tsvetkov. Among us: Measuring and mitigating malicious contributions in model collaboration systems. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15969–15988, 2026.

Yilun Yao, Xinyu Tan, Chao-Hsuan Liu, Yaoming Li, Zhengyang Wang, Wenhan Yu, Zhewen Tan, Yuxuan Tian, Guangxiang Zhao, Lin Sun, et al. Harness-bench: Measuring harness effects across models in realistic agent workflows. *arXiv preprint arXiv:2605.27922*, 2026.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. Dapo: An open-source llm reinforcement learning system at scale. *Advances in Neural Information Processing Systems*, 38:113222–113244, 2026.

Matei Zaharia et al. The shift from models to compound AI systems. `https://bair.berkeley.edu/blog/2024/02/18/compound-ai-systems/`, 2024. BAIR Blog.

Haozhen Zhang, Tao Feng, and Jiaxuan You. Router-r1: Teaching llms multi-round routing and aggregation via reinforcement learning. *Advances in Neural Information Processing Systems*, 38: 141233–141265, 2026a.

Shaokun Zhang, Ming Yin, Jieyu Zhang, Jiale Liu, Zhiguang Han, Jingyang Zhang, Beibin Li, Chi Wang, Huazheng Wang, Yiran Chen, et al. Which agent causes task failures and when? on automated failure attribution of llm multi-agent systems. In *International Conference on Machine Learning*, 2026b.

Yichi Zhang, Kevin Lu, Yuang Zhang, Jie Gao, Lirong Xia, and Fang-Yi Yu. Mixture of complementary agents for robust llm ensemble. *arXiv preprint arXiv:2605.24048*, 2026c.

Yusen Zhang, Ruoxi Sun, Yanfei Chen, Tomas Pfister, Rui Zhang, and Sercan Ö Arık. Chain of agents: Large language models collaborating on long-context tasks. *Advances in Neural Information Processing Systems*, 37:132208–132237, 2024.

Andrew Zhao, Yiran Wu, Tong Wu, Quentin Xu, Yang Yue, Matthieu Lin, Shenzhi Wang, Qingyun Wu, Zilong Zheng, and Gao Huang. Absolute zero: Reinforced self-play reasoning with zero data. *Advances in Neural Information Processing Systems*, 38:105816–105879, 2026.

Wenting Zhao, Pranjal Aggarwal, Swarnadeep Saha, Asli Celikyilmaz, Jason Weston, and Ilia Kulikov. The majority is not always right: Rl training for solution aggregation. *arXiv preprint arXiv:2509.06870*, 2025a.

Yujie Zhao, Lanxiang Hu, Yang Wang, Minmin Hou, Hao Zhang, Ke Ding, and Jishen Zhao. Stronger together: On-policy reinforcement learning for collaborative llms. *arXiv e-prints*, pp. arXiv–2510, 2025b.

Yifei Zhou, Andrea Zanette, Jiayi Pan, Sergey Levine, and Aviral Kumar. Archer: Training language model agents via hierarchical multi-turn rl. In *International Conference on Machine Learning*, 2024.

Mingchen Zhuge, Wenyi Wang, Louis Kirsch, Francesco Faccio, Dmitrii Khizbullin, and Jürgen Schmidhuber. Gptswarm: Language agents as optimizable graphs. In *Forty-first International Conference on Machine Learning*, 2024.

A Additional Results and Analysis

A.1 Individual Performance

Table 3 scores every method with the policy answering alone, outside any collaborative system. Because a policy's solo score does not depend on the protocol it was later evaluated in, the five baselines are reported once; only C-RL is trained separately per protocol.

Read against Table 1, these numbers make the decoupling of §2 visible within the training experiment rather than across a pool of frozen models. Ranking the five baselines by individual performance does not reproduce their ranking as collaborators. Collaborative RAFT is among the strongest baselines in individual mode (average 0.524, essentially tied with GRPO-indiv at 0.526), yet it is the weakest collaborator overall, improving on *Base* in only 14 of 24 cells and finishing last or second to last under all three protocols. Prompt nudge is the mirror image: it changes no weights and sits near the bottom individually (0.498, against 0.484 for *Base*), yet it is the best baseline under LLM-Blender and the second best under SLM-MUX. The same gap holds for C-RL itself, which leads in individual mode in 13 of the 24 cells against 21 of 24 in collaborative mode. A method's individual standing is thus a poor guide to its collaborative standing in either direction, which is why §3.3 reads C-RL's individual gains as a by-product rather than as the mechanism behind its collaborative ones.

Method	GPQA-dia	Theorem	KernelB	SMDD	AssayB	ChemB	Culture-H	MMLU-PX	Avg.
Base	.687	.547	.103	.279	.023	.694	.783	.758	.484
GRPO-indiv	.808	.608	.111	.322	.026	.714	.832	.784	.526
DAPO-indiv	.758	.640	.114	.261	.024	.724	.819	.820	.520
prompt-collab	.697	.580	.110	.260	.026	.700	.818	.796	.498
RAFT-collab	.768	.637	.108	**.410**	.017	.684	.806	.764	.524
C-RL, LLM-Blender	**.889**	**.690**	.097	.297	.024	.710	.842	**.826**	**.547**
C-RL, MA Debate	.818	.640	**.115**	.340	**.029**	.750	.834	**.826**	.544
C-RL, SLM-MUX	.748	.552	.094	.358	.023	**.752**	**.843**	.810	.523

Table 3: **Individual performance**: the trained policy answers alone, outside any collaborative system. The five baselines do not depend on the collaboration protocol and are reported once; C-RL is reported per training protocol. Best per column in **bold**. *Avg.* is the mean over all eight datasets.

Method	GPQA-dia	Theorem	KernelB	SMDD	AssayB	ChemB	Culture-H	MMLU-PX	Avg.
Collaboration protocol: LLM-Blender									
Base	.818	.613	.144	.635	.035	.720	.806	.804	.572
C-RL (ours)	**.879** [.808,.939]	**.708*** [.662,.752]	**.183** [.120,.256]	**.686** [.606,.759]	**.046** [.024,.069]	**.776*** [.738,.812]	**.843*** [.814,.873]	**.840*** [.808,.872]	**.620*** [.602,.638]
Collaboration protocol: Multiagent Debate									
Base	.768	.623	.155	.655	.036	.720	.817	.818	.574
C-RL (ours)	**.899*** [.838,.949]	**.655** [.608,.703]	**.194** [.128,.264]	**.664** [.584,.745]	**.044** [.024,.069]	**.764*** [.726,.800]	**.845** [.816,.873]	**.846** [.814,.876]	**.614*** [.596,.632]
Collaboration protocol: SLM-MUX									
Base	.838	.568	.112	.721	.079	.718	.845	.856	.592
C-RL (ours)	**.879** [.808,.939]	**.672*** [.625,.718]	**.201*** [.136,.272]	**.793*** [.723,.861]	**.109** [.078,.144]	**.766*** [.728,.802]	**.873*** [.847,.899]	**.874** [.844,.902]	**.646*** [.628,.663]

Table 4: **Confidence intervals for C-RL.** For each collaboration protocol we report the untrained *Base* policy alongside *C-RL*, with 95% confidence intervals for C-RL in brackets. * marks cells where the lower bound of the C-RL interval still exceeds the Base score.

A.2 Confidence Intervals

Table 4 places the untrained policy beside C-RL under each protocol, with a 95% confidence interval on every C-RL score. The intervals are computed at each task's true test-set size, which ranges from 99 items on GPQA-Diamond to 613 on CulturalBench-Hard, so they capture how much of a gap is attributable to having scored a finite test set. A star marks the cells where the lower end of the interval still sits above the *Base* score, which holds for the aggregate column under all three protocols; the exceptions among individual datasets are concentrated on the two smallest and noisiest tasks. We report these for transparency about precision rather than as hypothesis tests: the claim in §3.3 rests on C-RL winning 21 of 24 cells and never falling below *Base*, a pattern across cells that no single interval speaks to.

A.3 Response-Level Analysis

The results in §3.3 are aggregate scores. To see what changes inside a rollout, we examine the per-item logs of every (protocol, task) pair for the untrained policy and for C-RL.

C-RL makes the policy's correct answers more likely to survive. Pairing each item's individual-mode score with its collaborative-mode score partitions the test set by where the policy and the system disagree. Two cells are informative. *Rescue* is the fraction of items the policy gets wrong alone but the system gets right, the case §3.1 appeals to when it argues that a policy can be rewarded without being individually correct. *Drag* is the opposite, items the policy gets right alone but the system gets wrong, which is information the collaboration destroys. Table 5 reports both. Drag falls under all three protocols, from 4.7 to 4.0 percent of items under LLM-Blender, 5.1 to 3.6 under

	LLM-Blender		Multiagent Debate		SLM-MUX	
Task	rescue	drag	rescue	drag	rescue	drag
GPQA-dia	.162 → .040	.030 → .051	.152 → .111	.071 → .030	.172 → .172	.020 → .040
Theorem	.098 → .048	.033 → .030	.100 → .052	.025 → .037	.085 → .152	.065 → .033
KernelB	.136 → .168	.064 → .032	.144 → .160	.056 → .024	.140 → .220	.120 → .020
SMDD	.438 → .438	.080 → .088	.474 → .423	.095 → .095	.504 → .526	.051 → .073
AssayB	.006 → .009	.000 → .000	.000 → .000	.000 → .003	–	–
ChemB	.098 → .120	.072 → .054	.104 → .064	.078 → .050	.112 → .058	.038 → .044
Culture-H	.077 → .034	.054 → .033	.086 → .034	.052 → .023	.096 → .054	.034 → .010
MMLU-PX	.086 → .048	.040 → .034	.092 → .044	.032 → .024	.112 → .102	.014 → .038
Mean	.137 → .113	.047 → .040	.144 → .111	.051 → .036	.174 → .183	.049 → .037

Table 5: **Where the policy and the system disagree.** Each cell reads *base→C-RL*. *Rescue* is the fraction of test items the policy answers incorrectly on its own but the system gets right; *drag* is the fraction the policy answers correctly on its own but the system gets wrong. Cells marked −− lack a usable log. Drag falls under all three protocols.

	option proposed by each model				system	policy len.
	Policy	gpt-oss	kimi-k3	nemotron		
What happens if aromatic nitro compounds are mixed with sodium hydroxide? **(A)** Nothing happens **(B)** They may explode **(C)** They decompose **(D)** They form a salt Correct: **B**						
Base	A	A	**B**	A	A	34,350
C-RL	**B**	A	**B**	A	**B**	4,059
Which of the following statements is true about halogenated organic compounds? **(A)** They are incompatible with strong oxidizing and reducing agents. **(B)** They react slowly with metals. **(C)** The more halogen atoms, the more reactive they become. **(D)** They are highly reactive. Correct: **A**						
Base	D	**A**	**A**	**A**	D	7,292
C-RL	**A**	**A**	**A**	**A**	**A**	6,811

Table 6: **Two ChemBench items where C-RL changes the system's answer.** Each model proposes a candidate and a fuser combines them; the letter shown is the option each candidate ends on, with the correct option in **bold**. In the first item the untrained policy proposes the wrong option and the system follows it; after C-RL the policy proposes the correct option and the system does too. In the second the untrained policy dissents from three collaborators that are all correct and the system adopts the policy's error. The last column gives the length of the policy's own candidate.

Multiagent Debate, and 4.9 to 3.7 under SLM-MUX. Rescue stays substantial throughout, between 11 and 18 percent of items, so the policy continues to be carried by its collaborators on a large share of questions even after training; what changes is that it is dragged down less often. Rescue itself falls slightly under LLM-Blender and Multiagent Debate, which is expected rather than informative, since a policy that answers more questions correctly on its own leaves fewer left to be rescued.

Two illustrative items. Table 6 shows two ChemBench questions under LLM-Blender where C-RL changes the system's answer, with each model's proposed option and the option the fuser settles on. In the first, the untrained policy proposes the wrong option and only one collaborator proposes the right one, and the system follows the majority into the wrong answer; after C-RL the policy proposes the correct option, and the system follows. In the second, the untrained policy dissents from three collaborators that all propose the correct option, and the system nonetheless adopts the policy's error, an instance of drag in which one participant overturns a system that would otherwise have been right. After C-RL the policy agrees and the answer stands. The two cases are the mechanism behind the aggregate numbers: the policy's contribution is not merely more often correct, it is weighted differently by the protocol that consumes it. The length column also records a side effect visible throughout the logs. The untrained policy's candidate in the first item runs to 34,350 characters, against 4,059 after C-RL, so training against the system's outcome shortens contributions that the fuser has to read.

Q. How many ways are there to arrange the letters in the word *BANANA* up to the symmetries of the word?	
Base	The phrase "up to the symmetries" implies treating the arrangement as a bracelet, where two arrangements are considered identical if one can be rotated or flipped to match the other (the dihedral group D_6). Applying Burnside's Lemma, the identity symmetry fixes all 60 distinct linear permutations, while the five non-trivial rotations fix none because the letter counts (3 A's, 2 N's, 1 B) cannot form the required periodic patterns. Among the six reflections, the three axes passing through opposite letter positions each fix 4 arrangements (by assigning the AA and NN pairs to the two swapped slots and the remaining A and B to the fixed slots), while the three axes passing between slots fix none. Summing the fixed counts gives $60 + 3 \times 4 = 72$, and dividing by the 12 total symmetries yields 6 distinct arrangements. Final Answer: $\boxed{6}$
C-RL	First, calculate the number of distinct linear permutations of the letters in BANANA, which is $6!/(3!\,2!) = 60$. The phrase "up to symmetries" for a linear word means we consider an arrangement equivalent to its reverse. Since the letter counts for 'A' (3) and 'B' (1) are odd, no arrangement can be a palindrome, meaning every permutation is distinct from its reverse and forms a unique pair. Dividing the total linear permutations by 2 to account for this reversal symmetry gives 30 distinct arrangements. Final Answer: $\boxed{30}$
Q. Determine the product obtained when 3,4-dimethylhexanedial is reacted with the following set of reagents: 1. KOH, H2O, THF, Heat 2. CH3CH2MgBr, H3O+ 3. PCC, CH2Cl2 4. O3, H2O **A.** 3,4-dimethyl-5,6-dioxooctanoic acid ... **D.** 3,4-dimethyl-5,6-dioxooctanal	
Base	Base-catalyzed aldol condensation of 3,4-dimethylhexanedial forms a cyclohexene derivative, which subsequently undergoes Grignard addition and PCC oxidation to introduce an ethyl ketone group. Ozonolysis cleaves the ring double bond to yield an acyclic chain containing aldehyde and ketone functionalities. The resulting 8-carbon skeleton and carbonyl positions precisely match 3,4-dimethyl-5,6-dioxooctanal, while ozonolysis of an internal alkene does not produce carboxylic acids, eliminating option A. Answer: D
C-RL	3,4-Dimethylhexanedial undergoes an intramolecular aldol condensation under basic conditions to form a five-membered cyclic α,β-unsaturated aldehyde. Subsequent Grignard addition and PCC oxidation convert the side-chain aldehyde into an ethyl ketone. Finally, oxidative ozonolysis cleaves the ring double bond, opening the ring to yield a linear dicarbonyl carboxylic acid chain. Counting the carbons and functional groups confirms the product is 3,4-dimethyl-5,6-dioxooctanoic acid. Answer: A

Table 7: **Collaborative responses the system produces, before and after C-RL.** Two items the untrained system answers incorrectly and the C-RL system answers correctly, quoted in full from the logs (the fuser's internal deliberation is omitted). Both are under LLM-Blender, on TheoremQA and GPQA-Diamond respectively; in both the disagreement is interpretive rather than arithmetic. The second item's two remaining distractors are elided: they are identical to one another in the source dataset and neither system proposes either of them.

What the system's answers look like. Table 7 quotes two collaborative responses in full, on items the untrained system gets wrong and the C-RL system gets right. Neither difference is a matter of arithmetic. In the TheoremQA item, the two systems disagree about what the question asks: the untrained system reads "up to the symmetries of the word" as the symmetries of a bracelet, invokes the dihedral group and Burnside's lemma, and answers 6, while the C-RL system reads the word as a linear string with only reversal symmetry, notes that no arrangement of BANANA is a palindrome, and divides the 60 linear permutations by two to reach 30. In the GPQA item the two agree on the whole reaction sequence and part company at the last step, where the untrained system asserts that ozonolysis "does not produce carboxylic acids" and answers with the aldehyde, while the C-RL system recognises that the aqueous workup specified in the question makes the ozonolysis oxidative and answers with the acid.

We read this as evidence that what C-RL changes is not a single skill but the way the policy's contribution is framed for the models that consume it. Both systems have the same collaborators and the same protocol; what differs is the candidate the policy supplies, and in both cases the decisive content is an interpretive commitment made early, about what is being asked or which variant of a reaction applies, that the rest of the system then follows. This is also why compositional strength is not a restatement of individual strength: the policy's influence runs through whether its framing

Model pool	QA	Reasoning	Coding	Pluralism & Safety	Science
Specialized Expert	65.8→72.4 (+6.6)	32.3→45.5 (+13.1)	44.1→65.5 (+21.4)	38.7→54.0 (+15.3)	41.2→55.4 (+14.2)
∼20-30B Open	82.9→94.9 (+12.0)	24.2→35.6 (+11.4)	70.7→82.5 (+11.8)	78.2→81.2 (+3.1)	57.7→60.8 (+3.1)
>100B Open	80.0→83.7 (+3.7)	48.1→60.0 (+11.9)	48.8→71.2 (+22.4)	77.0→82.0 (+4.9)	53.7→68.3 (+14.6)
Proprietary API	86.8→90.0 (+3.2)	58.5→69.4 (+10.9)	66.7→94.1 (+27.5)	32.1→37.7 (+5.6)	73.1→80.6 (+7.5)
Diverse Academic	66.1→70.0 (+3.9)	32.4→65.3 (+32.9)	37.4→52.7 (+15.4)	64.6→71.6 (+7.0)	50.4→55.1 (+4.7)

Table 8: The best-performing collaboration method across 25 (model pool, task category) values underlying the collaboration-effectiveness claim in §2.3. The mean of the 25 gains is $+11.5$ with a 95% CI of $[+8.7, +14.7]$.

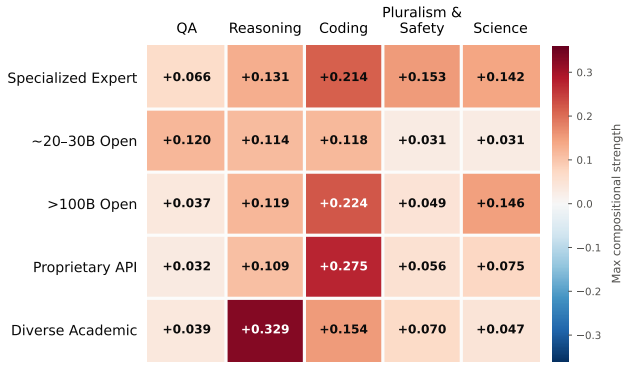

Figure 5: Best compositional strength achieved by collaboration methods, for each (model pool, dataset category). Each cell reports $\max_{m,t \in \text{category}} \bar{\phi}(m,t)$, where $\bar{\phi}(m,t)$ is the mean compositional strength across models for m, t.

is adopted by the mechanism that reads it, and Table 5 shows that an untrained policy which is individually correct can still push a system to the wrong answer.

A.4 Collaboration Effectiveness by Pool and Domain

Table 8 lists the 25 (model pool, task category) values underlying the collaboration-effectiveness claim in §2.3: for each cell, the leave-one-out and full-coalition scores and the resulting gain, for the (dataset, method) pair that maximizes it.

A.5 Collaboration Helps Unevenly Across Task Domains

Figure 5 aggregates the best achieved compositional strength for every (pool, domain) combination. Coding stands out as the domain where collaboration helps the most consistently: it is the strongest domain for three of the five pools (Specialized Expert, >100B Open, Proprietary API) and is never the weakest domain for any pool. QA, by contrast, is where collaboration helps the least: it is the weakest domain for four of the five pools, with the sole exception of the ∼20-30B Open pool. Reasoning shows a large but concentrated effect: Diverse Academic Models benefits from collaboration on Reasoning tasks more than any other single (pool, domain) cell in the grid, a peak not shared by the other four pools. Taken together with the win-rate result of §2.4, this indicates that the payoff from collaboration is governed far more by *what kind of task* is being solved, *which method* is used, and in some cases the specific composition of the pool, than by any single model's individual competence. This reinforces why compositional strength cannot be predicted from a model's individual

benchmark score alone. A finer-grained, per-dataset (rather than per-domain) version of this analysis is provided in Appendix A.6.

A.6 Per-Dataset Gain Consistency

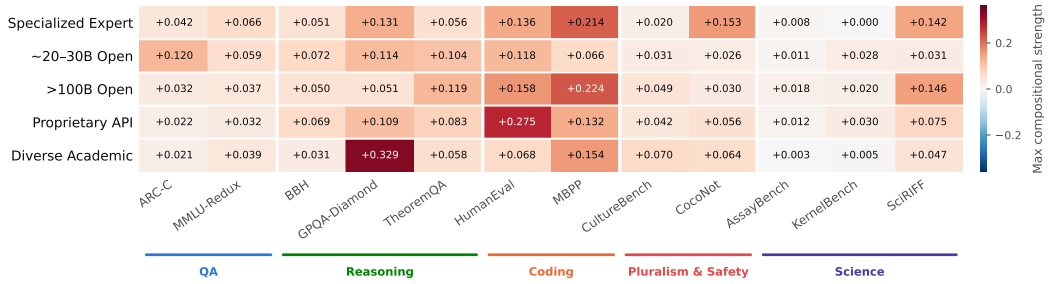

Figure 6: Maximum compositional strength achieved by any collaboration method, for each (model pool, dataset) pair: the per-dataset counterpart to the domain-level aggregation in Figure 5. Datasets are grouped by category (column separators and colored labels); each cell reports $\max_m \bar{\phi}(m, t)$ for dataset t.

Figure 6 refines two of the domain-level patterns from Figure 5. First, the Coding datasets (HumanEval, MBPP) each show consistently positive gains individually, not just as a category average, confirming that Coding's strength is a property of the domain rather than an artifact of aggregating over its two datasets. Second, and in contrast to a naive reading of the domain-level heatmap, Science is *not* a uniformly weak domain: it is bimodal at the dataset level. AssayBench stays near zero across every pool (≤ 0.018), and KernelBench remains small throughout (≤ 0.030), while SciRIFF alone reaches 0.142 and 0.146 for the Specialized Expert and $>$100B Open pools respectively, comparable to those pools' gains on Reasoning or Pluralism & Safety. The domain-level Science column is therefore driven almost entirely by SciRIFF, and would be misleading if read as evidence that collaboration uniformly fails to help on scientific tasks. Finally, within Reasoning, GPQA-Diamond shows a markedly stronger and more consistent gain across pools than BBH or TheoremQA, most strikingly for Diverse Academic Models, whose domain-level Reasoning peak in Figure 5 ($+0.329$) is attributable entirely to GPQA-Diamond.

A.7 The Shapley Value: Derivation and Axioms

We assemble the grand coalition one model at a time and measure how much model i adds at the point it joins. The Shapley value $\phi_i(v)$ is that amount averaged over all $n!$ orders. Since what i adds depends only on which models precede it, the orders can be grouped by that preceding set S, giving the sum in equation 2. The term $v(S \cup \{i\}) - v(S)$ is the *marginal contribution* of model i to coalition S, and the weight $\frac{|S|!\,(n-|S|-1)!}{n!}$ is exactly the fraction of the $n!$ orderings of $\mathcal{N}$ in which all members of S precede i and every other model follows.

The Shapley value is the unique attribution rule satisfying four axioms (Shapley et al., 1953): the shares sum to the value the full coalition creates, interchangeable models receive equal credit, a model that never changes any outcome receives nothing, and attribution is linear when games are combined. We thus take it as the definition of compositional strength rather than one heuristic among many: under it, a model's compositional strength is its expected usefulness, averaged over every coalition it might end up in.

A.8 Comparing Contribution Estimators

Section 2.1 estimates compositional strength with the leave-one-out marginal equation 3 rather than the exact Shapley value equation 2, which would require all 2^n coalition values. On a small enough pool the exact computation is affordable, which lets us check the estimator directly. We do so on the Specialized Expert pool ($n = 4$, so 16 coalitions per cell), computing the exact Shapley value, the Banzhaf value, and the leave-one-out marginal from the same cached coalition values, over 4

Pool	n	Members
Specialized Expert	4	Qwen2.5-7B-Instruct, Qwen2.5-Math-7B, Qwen2.5-Coder-7B-Instruct, Qwen2.5-7B-Instruct-1M
∼20–30B Open	7	Qwen3-14B, Nemotron-3-Nano-30B-A3B, gpt-oss-20b, Llama-3.1-8B-Instruct, gemma-3-12b-it, phi-4, DeepSeek-R1-Distill-Qwen-14B
>100B Open	4	gpt-oss-120b, deepseek-v4-pro, kimi-k2.6, nemotron-3-ultra
Proprietary API	3	gpt-5.5, gemini-3.5-flash, bedrock-claude-opus-4-8
Diverse Academic (Feng et al., 2026e)	8	parti_0_full, parti_1_full, parti_2_full, parti_3_full, parti_4_full, parti_5_full, parti_6_full, parti_7_full

Table 9: **The five model pools.** Each pool defines a separate cooperative game; compositional strengths are computed within a pool and never compared across pools. *Diverse Academic* is a set of independently post-trained research models, each specialised for a different task (narrative claim verification, tool use, planner optimisation, architecture ablation, instruction following, instruction induction, conditional-probability estimation, and multi-document reasoning respectively).

datasets (ARC-Challenge, BBH, HumanEval, SciRIFF) and 4 collaboration methods (Prompt Routing, Multiagent Debate, LLM-Blender, AggLM). This gives 64 (model, dataset, method) observations per estimator.

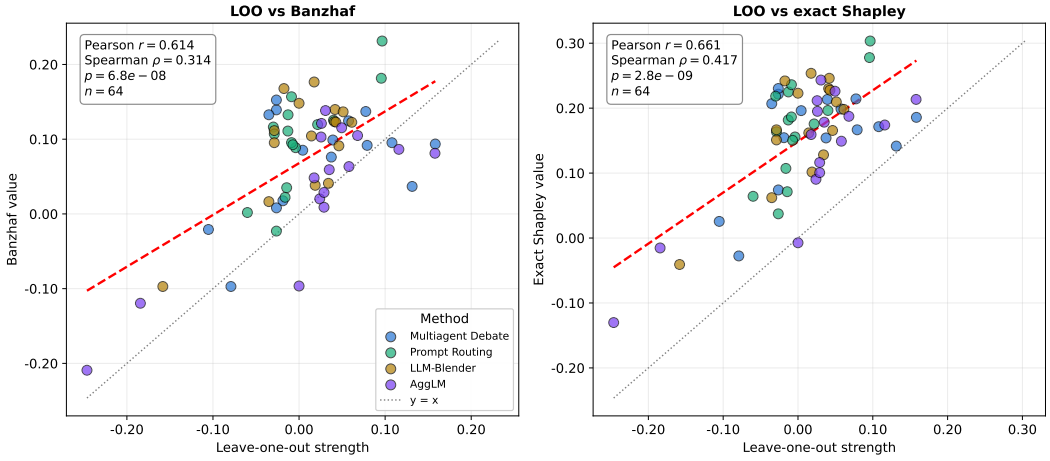

Figure 7: Leave-one-out strength against the Banzhaf value (left) and the exact Shapley value (right) on the Specialized Expert pool. Each point is one (model, dataset, method) cell, colored by collaboration method; the dotted line is $y = x$ and the dashed line is the least-squares fit.

Figure 7 shows that the leave-one-out marginal tracks both of the more expensive estimators. The relationship is positive and highly significant in each case, with Pearson $r = 0.61$ against Banzhaf ($p = 6.8 \times 10^{-8}$) and $r = 0.66$ against exact Shapley ($p = 2.8 \times 10^{-9}$). Two qualifications are visible in the same plots. First, agreement is stronger in magnitude than in rank: the Spearman correlations are 0.31 and 0.42 respectively, so the estimators do not order models identically even when they broadly agree on how much each contributes. Second, nearly every point lies above the $y = x$ line, meaning the leave-one-out marginal reads systematically lower than either alternative. This is the expected direction. Measuring a model's contribution only against the full pool lets the remaining models compensate for its removal, which suppresses the credit assigned to models whose competence is duplicated elsewhere in the pool, exactly the residual term isolated in equation 4.

Dataset	Category	Test items
ARC-Challenge (Clark et al., 2018)	QA	1172
MMLU-Redux (Gema et al., 2025)	QA	1000
GPQA-Diamond (Rein et al., 2024)	Reasoning	99
BBH (Suzgun et al., 2023)	Reasoning	1000
TheoremQA (Chen et al., 2023)	Reasoning	400
HumanEval (Chen et al., 2021)	Coding	114
MBPP (Austin et al., 2021)	Coding	487
CulturalBench-Hard (Chiu et al., 2024)	Pluralism	613
CocoNot (Brahman et al., 2024)	Pluralism	1000
SciRIFF (Wadden et al., 2025)	Science	139
KernelBench (Ouyang et al., 2025)	Science	125
AssayBench (Brouwer et al., 2026)	Science	334

Table 10: **The twelve evaluation datasets.** Coalition values are computed on the full test split. All scoring is rule-based or execution-based; no model acts as a judge.

B Experiment Details

B.1 Evaluation Setup Details

Model pools. Table 9 lists the members of each of the five pools. Pools were chosen to vary along two axes that plausibly matter for collaboration, the scale of the participants and how heterogeneous their training is, ranging from four sibling 7B models differing only in their specialisation to eight independently post-trained research models with nothing in common but a rough size class.

Datasets. Table 10 lists the twelve datasets, their category, and the number of test items. Every coalition value is computed on the full test split, and every scorer is rule-based or execution-based, so no model acts as a judge anywhere in the evaluation. This matters for the leave-one-out differences in equation 3: because the same items are scored the same way for every coalition, a change in $\hat{\phi}_i$ reflects the removal of model i and nothing else.

Collaboration methods. All twelve methods are run through MoCo (Feng et al., 2026b) with the configurations below; unless stated otherwise generation uses temperature 0.7 and top-p 0.9. *Prompt Routing* and *Graph Router* consume one natural-language description per pool member, the latter also training a small GNN router over MiniLM embeddings of the query (8 hidden features, 3 in-edges, learning rate 10^{-4}, 500 epochs). *Trained Router* LoRA-trains a Qwen2.5-7B router (r=64, α=16, five epochs, learning rate 10^{-5}) against a Skywork-Reward-Llama-3.1-8B reward model. *Switch Generation* uses a fixed Qwen2.5-7B selector with a patch size of 50 tokens. *Multiagent Debate* runs three rounds; *Multiagent Feedback* runs one feedback-and-refine cycle with one critic per answer; *Heterogeneous Swarms* searches with a population of five for five iterations. *SLM-MUX* draws three samples per model per question and breaks ties on the development split. *LLM-Blender* ranks candidates listwise and fuses the top three, with both ranker and fuser fixed to an off-the-shelf Qwen2.5-7B; fixing the fuser matters for validity, since letting it default to the first pool member would silently change the fuser whenever that member is the one being left out. *AggLM* GRPO-trains a 1.5B aggregator per coalition, *Multiagent Finetuning* LoRA-trains generator and critic copies of every pool member, and *Sparta Alignment* DPO-trains the pool over three iterations.

B.2 C-RL Implementation Details

Datasets. Table 11 lists the ten datasets used in §3. The eight training datasets are sized by their MoCo development split, which we further divide $8{:}2$ into training and validation for C-RL; the two datasets used only for the held-out generalization evaluation of §3.4 are sized by their test split instead. Two of these, KernelBench and SMDD-Bench, score the system's answer by execution rather than reference-matching. For KernelBench, the submitted CUDA kernel is compiled and benchmarked against the reference implementation on one H100 GPU: correctness is checked over randomized-input trials, and the reward is 0.25 for a correct-but-not-faster kernel, jumping to 0.5 for any measured speedup, with the remaining 0.5 scaled linearly by speedup magnitude up to

Dataset	Category	Development	Test
GPQA-Diamond (Rein et al., 2024)	Answer-matching	99	99
TheoremQA (Chen et al., 2023)	Answer-matching	400	400
MMLU-ProX (Xuan et al., 2025)	Answer-matching	500	500
CulturalBench-Hard (Chiu et al., 2024)	Answer-matching	613	613
ChemBench (Mirza et al., 2025)	Answer-matching	498	500
KernelBench (Ouyang et al., 2025)	Execution-verified	125	125
SMDD-Bench (Han et al., 2026)	Execution-verified	135	137
AssayBench (Brouwer et al., 2026)	Reference-ranking	218	334
APEX-Shortlist (Dekoninck et al., 2026)	Answer-matching	–	40
BixBench (Mitchener et al., 2025)	Answer-matching	–	103

Table 11: **The ten datasets used in §3.** The eight training datasets (top) list both their MoCo development split, which we further divide 8:2 into training and validation for C-RL, and their test split, reported for reference. APEX-Shortlist and BixBench (bottom) are used only for the held-out generalization evaluation of §3.4, so only their test-set size is listed. *Answer-matching*, *execution-verified*, and *reference-ranking* follow the scoring split of §3.2.

a difficulty-dependent cap ($10\times/5\times/2\times$ for Level 1/2/3 problems in KernelBench). For SMDD-Bench, the submitted molecule is checked with RDKit against a set of hard constraints (molecular weight, LogP, TPSA, hydrogen-bond counts, rotatable bonds, PAINS/Brenk filter matches, and Tanimoto similarity to a reference) for lead-optimization items, or scored by the recall and specificity of a submitted pharmacophore rule against held-out active and decoy molecules for pharmacophore items, depending on problem type. AssayBench, by contrast, requires no execution: the model submits a ranked list of genes, which is scored by Adjusted nDCG@100 against a precomputed relevance table from a real CRISPR genetic screen, with unmatched or hallucinated genes scored as zero relevance rather than dropped from the ranking. Whether the collaboration protocol is single-step (routing dispatches each query to one model for one generation) or multi-step (agentic and training-based methods exchange several rounds before a final answer), the execution tool is invoked as many times as the protocol's own structure requires; equation 9 broadcasts the final system reward score to every policy generation the rollout produced.

Infrastructure. The policy is fully fine-tuned, not adapter-tuned. Each run uses three nodes of eight H100 GPUs, with tensor parallelism 4, pipeline parallelism 2, and sequence parallelism enabled; rollout generation is served separately at tensor parallelism 4. The sequence length is $16{,}384$ tokens with a response budget of $15{,}000$, and batches are formed dynamically up to $6{,}144$ tokens per GPU. A complete C-RL run takes roughly 30 hours on this allocation.

Rollouts and optimization. Each optimizer step samples $N = 8$ questions and runs $K = 8$ independent executions of the collaboration protocol per question, so a step involves 64 full protocol executions and a variable number of policy generations depending on how often the protocol calls the policy. Advantages are computed per question over the pooled generations as in equation 10, and the policy is updated with a clipped policy-gradient loss at a learning rate of 10^{-6}. We use no KL anchor to the initial policy: the coefficient is set to zero, so the only pressure on the policy is the system's reward. Training runs for up to 100 rollout steps.

Both μ_q and σ_q in equation 10 are treated as fixed statistics of the sampled group, with no gradient flowing through them: dividing by σ_q rescales every term in the group by the same positive constant, so it does not reintroduce b_q or disturb the cancellation of equation 7, only the scale of the learning signal for that question. Because the loss averages uniformly over every policy-generation token in the batch rather than first normalizing by question, a question whose rollouts call the policy more often (larger $|\mathcal{T}_q|$) contributes proportionally more gradient signal in that step.

Attributing generations to the policy. §3.1 requires that the loss touch only tokens the policy actually sampled, which is non-trivial because a rollout is a transcript produced by four models. We serve the policy to the collaboration protocol behind a shim that exposes an OpenAI-compatible endpoint, so the protocol calls the policy exactly as it calls any other participant and requires no modification. Every call that reaches the shim is recorded as one entry, tagged with the question and

rollout it belongs to and ordered by turn, storing the prompt tokens, the sampled response tokens, and the log-probabilities under which they were sampled. Collaborator generations are issued to other endpoints and never enter this record. The training step then consumes the recorded entries directly, so the loss is applied to the policy's own tokens under their own sampling distribution, and the collaborators' text enters only as context. This is also what makes equation 9 straightforward to implement: the scalar reward for a rollout is broadcast to every recorded entry carrying that rollout's tag.

Protocol configuration during training. Collaboration runs for two rounds. LLM-Blender fuses the top 3 candidates and ranks them listwise with an off-the-shelf gpt-oss-120b ranker, rather than pairwise, which costs one ranking call per question instead of $\binom{n}{2}$. SLM-MUX draws 3 samples per model per question. Generation uses temperature 0.7 and top-p 0.9 throughout, for the policy and collaborators alike.

Checkpoint selection. A held-out development split, the last 20% of the training pool, is reserved for model selection and never used for gradient updates. Every 3 steps we checkpoint and score the current policy on 16 development questions with a single rollout each, and training stops early if the development score has not improved for 25 consecutive evaluations. We retain the three best checkpoints by development score and report the single best. All reported numbers come from that checkpoint evaluated on the full test split with one rollout per question at a fixed seed, in both individual and collaborative mode.

Baselines. GRPO and DAPO use the same infrastructure, sequence budget, and learning rate, differing only in that the reward is the policy's own correctness and the rollout is a single-model generation rather than a protocol execution. Collaborative RAFT runs the full protocol with the untrained policy, keeps the transcripts whose collaborative outcome scored well, and supervised-fine-tunes the policy on its own generations from those transcripts. Prompt nudge changes no weights and prepends a description of the collaborative system to the policy's system prompt.

C Related Work

Multi-LLM collaboration. A growing body of work builds systems in which several language models solve a task together rather than one model solving it alone (Du & Kaelbling, 2024; Feng et al., 2026c), a shift from models to compound systems (Zaharia et al., 2024). Approaches differ in what the models exchange. Routing and cascading methods dispatch each query to a single model, selecting by learned preference (Ong et al., 2025), graph structure (Feng et al., 2025c), generation-time switching (Feng et al., 2025b), prompted choice (Frick et al., 2025), cost-aware cascades (Chen et al., 2024a; Ding et al., 2025), or inferred skill requirements (Chen et al., 2026). Agentic methods let models exchange text over several rounds, through debate (Du et al., 2023), feedback (Feng et al., 2024), optimizable communication graphs (Zhuge et al., 2024), recruited expert teams (Chen et al., 2024b), sequential context passing (Zhang et al., 2024), or swarm search over roles (Feng et al., 2025a). A third family combines outputs or parameters directly, by ranking and fusing candidates (Jiang et al., 2023), layering mixtures of agents (Wang et al., 2025b), training aggregators (Zhao et al., 2025a), fine-tuning within a society of models (Subramaniam et al., 2025), or aligning a pool collectively (Jiang et al., 2025). Two distinctions separate our setting from much of this work. First, a large part of the multi-agent literature instantiates a *single* model repeatedly under different prompts, personas, or manually crafted roles, so the participants differ in their instructions rather than in their parameters. We study pools of genuinely distinct models, which is what makes contribution a property of a model rather than of a prompt. Second, alongside methods that compose frozen models at inference time, we also include collaboration methods that update model weights, i.e., training-time model collaboration, so the pool is not required to stay fixed. Across this entire literature, it is the collaboration strategy that gets optimized: making models better participants and collaborators, independent of which strategy it is placed into, is a separate research question that we address with C-RL.

Evaluating models alone and in systems. Language models are predominantly evaluated on their own, against benchmarks and leaderboards (Liang et al., 2023; Chiang et al., 2024). Another line

evaluates models as parts of systems, finding that measured capability depends on the interactive setting (Lee et al., 2023), the agent scaffold (Kapoor et al., 2024; 2026), and the surrounding harness (Yao et al., 2026). A third asks when collaboration is worthwhile: Pappu et al. (2026) report cases where a team falls short of its strongest member, Cemri et al. (2026) catalogue how multi-agent systems fail, Kim et al. (2026) study how the benefit of coordination varies with the capability of the models involved, and Zhang et al. (2026c) argue that a model's value lies in its complementarity with others. Sanchez & Hitaj (2025) likewise estimate synergistic and antagonistic interaction between models in order to recommend which ensemble to deploy. While previous work focuses on one ensemble or task at a time, we make it general: we measure a model's contribution across many pools, tasks, and collaboration methods, and ask directly whether it can be predicted from the model's own benchmark score.

Attributing contribution. Dividing a collective outcome among participants is the classical problem addressed by the Shapley value (Shapley et al., 1953) and the Banzhaf value (Banzhaf III, 1964), both of which are widely used in machine learning (Lundberg & Lee, 2017; Rozemberczki et al., 2022). Most applications value training data (Ghorbani & Zou, 2019; Kwon & Zou, 2022; Wang & Jia, 2023), where exact computation is intractable and much of the effort goes into cheaper estimators (Wang et al., 2024; 2025a). The relation between leave-one-out marginals and full Shapley values has been studied directly: Kwon & Zou (2022) place both within a single semivalue family, Wang et al. (2024) find leave-one-out less robust than Shapley for attributing individual training instances, and Lu et al. (2026) find it sufficient for identifying weak agents in multi-agent LLM systems. Within language-model systems, attribution has been applied to prompt tokens (Horovicz & Goldshmidt, 2024), workflow modules (Yang et al., 2025), and agents (Lu et al., 2026; Hua et al., 2026; Zhang et al., 2026b), in each case to diagnose or improve one particular system. We instead treat a model's marginal contribution as a property of the model, measure it across many pools, tasks, and collaboration methods, and then optimize it directly.

Reinforcement learning for language models. Post-training with RL is standard practice, from preference feedback (Ouyang et al., 2022; Rafailov et al., 2023) to verifiable rewards (Guo et al., 2025; Lambert et al., 2024), commonly optimized with group-relative estimators (Shao et al., 2024; Yu et al., 2026) built on clipped policy gradients (Schulman et al., 2017). Extending these methods to multi-turn and agentic settings raises questions of credit assignment (Zhou et al., 2024; Feng et al., 2026a), including whether group-relative baselines remain appropriate when prompts differ across roles and turns (Zhao et al., 2025b). Several lines of work apply reinforcement learning to multi-model systems. Some co-train every participant against the collaborative outcome (Park et al., 2025; Liu et al., 2026b; Estornell et al., 2025b; Motwani et al., 2025); some train a model against copies of itself in self-play (Liu et al., 2026a; Ma et al., 2024; Zhao et al., 2026); some distill a multi-agent system into a single model (Li et al., 2025); and some train only the component that routes or aggregates while the rest of the pool stays frozen (Estornell et al., 2025a; Zhang et al., 2026a; Fu et al., 2025). Unlike any of these, our collaborators are neither trained nor copies of the policy, and the reward is not the system's raw outcome but the policy's own marginal contribution to it. As a result, C-RL can be flexibly applied to diverse model/task/protocol combinations, optimize compositional strength directly, and enhance the policy as better collaborators in general. The name *compositional reinforcement learning* has separately been used for composing training prompts into harder ones (Xu et al., 2026) and for decomposing a single-agent control task via logical specifications (Jothimurugan et al., 2021); both target a single policy acting alone, whereas our C-RL trains a policy against the outcome of a live multi-model collaboration.