SEPAL: SEPARATED EXPERT PAIRS WITH ANSWER-LEVEL FUSION FOR RELIABLE LLM COLLABORATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Multi-agent collaboration lets large language models (LLMs) improve question answering through deliberation and feedback. Yet shared discussion couples correction with exposure to the same mistakes, which can erode the diversity needed for voting. Self-consistency offers sampling diversity without feedback, while single-pair Actor–Critic collaboration refines only one candidate. We introduce SEPAL, which assigns three private Actor–Critic teams to direct reasoning, evidence grounding, and verification. Role-specific training gives the teams different reasoning objectives beyond sampling variation. Each Critic guides revisions within its own team, preventing feedback from carrying errors across candidates. Once revision ends, majority voting combines only the final answers, keeping the reasoning histories separate until the decision. Across five open-weight backbones and five question-answering benchmarks, SEPAL improves mean accuracy by 1.81 percentage points over a matched single Actor–Critic pair, with improvements across all five backbones.

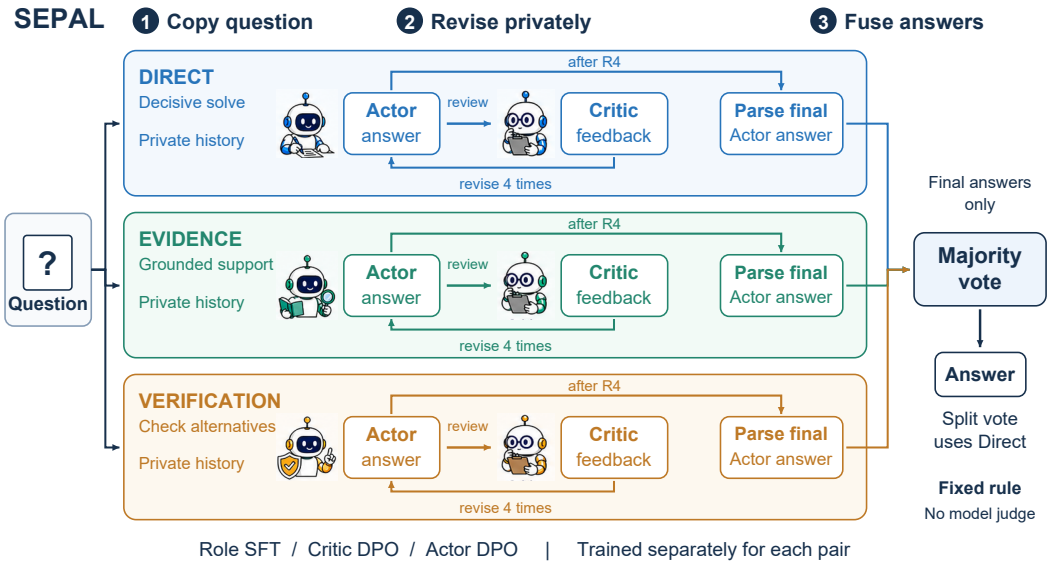

Figure 1: **SEPAL in one view.** A question enters three isolated role pairs, each refined through four private Actor–Critic rounds. Only final parsed answers cross team boundaries. A fixed majority vote returns the prediction and uses Direct for split decisions.

1 INTRODUCTION

Reasoning with large language models (LLMs) increasingly relies on generating, inspecting, and comparing intermediate solutions. Chain-of-thought prompting makes a reasoning path explicit

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.

(Wei et al., 2022), and self-consistency improves reliability by sampling several paths (Wang et al., 2023). Multi-agent discussion adds opportunities to question an argument and revise an answer (Du et al., 2024; Liang et al., 2024). For question answering, these approaches offer both feedback that can repair mistakes and alternative answers for the final decision.

However, existing approaches leave a gap between these two benefits. In shared-history debate, each agent sees peer arguments, so an early error can enter several later candidates. Self-consistency keeps sampled paths separate but provides no corrective feedback. ACC-Collab learns an Actor and a Critic using the correctness of subsequent Actor responses (Estornell et al., 2025), but its interaction refines a single answer. It supplies learned feedback without a set of separately revised candidates for voting.

The key issue is that a vote depends on how candidates fail together, as well as how accurate they are individually. A critique can repair an answer, yet a shared critique can also steer several agents toward the same faulty premise. When revision increases agreement without adding independent evidence, better individual answers can coexist with less benefit from voting. Useful collaboration therefore depends on both individual accuracy and the errors candidates share after feedback.

We hypothesize that restricting feedback to each team and combining only final answers can retain revision gains while preserving useful differences between candidates. Private feedback removes the path by which one team's faulty argument enters another team's reasoning before the vote. The final decision can then draw on solutions developed in separate conversation histories.

We instantiate this hypothesis in SEPAL, **S**eparated **E**xpert **P**airs with **A**nswer-**L**evel fusion (Figure 1). Each of three teams contains an Actor that proposes an answer and a Critic that guides revision. To introduce structured diversity beyond repeated sampling of one pair, we assign different reasoning objectives. Direct derives an answer, Evidence grounds it in relevant facts or passages, and Verification re-solves the question and checks alternatives. These objectives guide both answer generation and critique. Teams revise privately before a fixed majority vote combines their final answers. Direct supplies the fallback for split decisions, with no additional judge.

The teams share a backbone and training questions but learn separate adapters. Role-specific supervised training initializes the Actors, followed by Critic and Actor preference learning. A Critic's feedback is preferred during training when it leads to more correct Actor revisions. We train each role on its own trajectories using ACC-Collab's preference rule.

We evaluate five open-weight backbones ranging from 2B to 8B parameters on five question-answering benchmarks. MMLU supplies all training data; the other four benchmarks test transfer. Relative to a matched single Actor–Critic pair, SEPAL improves macro accuracy for every backbone by 1.06–2.22 percentage points, averaging 1.81 points.

Our contributions are threefold.

- A collaboration method that separates local Actor–Critic revision from final aggregation and trains three teams with distinct reasoning objectives.
- Evidence that revision gives the largest component gain, while final voting exceeds the strongest individual role in 21 of 25 evaluated settings.
- An analysis of diminishing returns from further revision, showing how later repairs are increasingly offset by regressions in correct answers.

2 Related Work

Independent sampling and late aggregation. Self-consistency improves chain-of-thought reasoning by sampling independent paths and aggregating their answers (Wang et al., 2023). Increasing the number of agents extends the same intuition to replicated LLM calls (Li et al., 2024), while Multiagent Finetuning trains separate agents to preserve distinct reasoning behaviors (Subramaniam et al., 2025). Tree of Thoughts preserves several partial solutions and alternates expansion with model-based evaluation, making diversity an explicit search resource (Yao et al., 2023). These methods differ in where alternatives are reduced: search may prune partial states, whereas late-voting systems keep complete candidates until the decision interface. Independent sampling keeps

candidate histories separate, while tree search uses intermediate evaluations to guide expansion. SEPAL gives each complete candidate its own learned reviewer and aggregates the revised answers.

Inference-time communication. Multi-agent debate exposes agents to peer arguments and can improve factuality or reasoning (Du et al., 2024; Liang et al., 2024); ChatEval applies debate to LLM evaluation (Chan et al., 2024). ReConcile aggregates diverse LLMs through a round-table protocol (Chen et al., 2024a), and Mixture-of-Agents passes outputs through layered aggregators (Wang et al., 2025). SEPAL permits revision within each team and applies a fixed answer vote after all teams finish. Its private feedback paths and final decision rule specify the information available to each agent.

Role prompts provide a second source of diversity. CoMM assigns distinct roles and reasoning paths and finds that independently prompted experts are important for science reasoning (Chen et al., 2024b). Unlike its collaborative discussion, however, SEPAL prevents a role from seeing peer content before aggregation. Role labels alone do not establish complementarity when a shared transcript can synchronize the candidates.

Candidate selection and corrective feedback. LLM-Blender learns a pairwise ranker and a generative fuser to select and merge outputs from heterogeneous models (Jiang et al., 2023b). That approach can exploit information beyond exact answers, but introduces a learned selection layer whose errors and training distribution become part of the system. SEPAL uses a fixed answer parser and majority rule to expose the quality of the candidate trajectories at the decision interface. Correction methods expose a related boundary. CRITIC grounds revision in tool-interactive feedback (Gou et al., 2024), while intrinsic self-correction without external feedback can preserve or amplify reasoning errors (Huang et al., 2024). Our Critics receive no tools or gold labels at test time, but are trained using the downstream correctness of Actor continuations. The component and round audits therefore test whether this learned, role-local signal repairs answers rather than assuming that another revision prompt is beneficial.

Learning to revise and collaborate. STaR bootstraps reasoning traces from successful solutions (Zelikman et al., 2022); Self-Refine and Reflexion use generated feedback or verbal memory to improve later behavior (Madaan et al., 2023; Shinn et al., 2023). DPO provides a direct objective for preference learning without an explicit reward model (Rafailov et al., 2023). ACC-Collab uses continuation correctness to construct Actor and Critic preferences (Estornell et al., 2025), while Multiagent Finetuning independently specializes agents to preserve diverse reasoning chains (Subramaniam et al., 2025). We retain ACC-Collab's continuation-valued training rule, replicate it independently for each role, and study the resulting components rather than assuming that the full training stack is uniformly beneficial.

3 METHOD

We study a question x with gold answer y. A collaborative system produces candidate trajectories τ_i ending in responses a_i, and a deterministic task-aware extractor maps each response to $z_i = g(a_i)$. The method is designed for *local repair*, so feedback can change a candidate rather than merely score it; *alternative preservation*, so candidate i never conditions on candidate j before aggregation; and *transparent fusion*, so the final decision does not hide another generative Judge. Shared dialogue can make several votes descendants of one error. Our goal is therefore *useful complementarity*: improve each trajectory while retaining enough residual variation for late fusion to matter.

3.1 ISOLATED ROLE TEAMS

For role $i \in \{D, E, V\}$, Actor A_i and Critic C_i form a separate team. At round 0, A_i produces a_i^0 and C_i returns feedback c_i^0. At rounds $t = 1, \ldots, 4$, the Actor revises from its previous answer and feedback, then the Critic reviews the new answer. The reported team answer is $z_i = g(a_i^4)$, with the following local generation dependencies.

$$a_i^0 \sim A_i(\cdot \mid x), \qquad c_i^t \sim C_i(\cdot \mid x, a_i^t), \qquad a_i^{t+1} \sim A_i(\cdot \mid x, a_i^t, c_i^t). \tag{1}$$

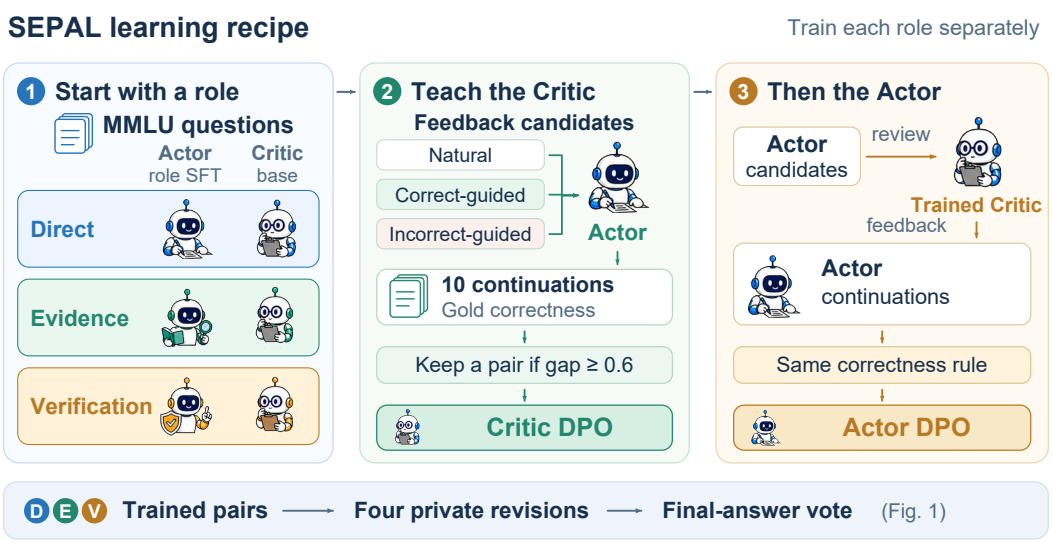

Figure 2: **Training the private teams.** Role SFT initializes Actors; Critics start from the base model. Feedback is valued by Actor continuation correctness. Critic DPO precedes Actor preference construction and Actor DPO. Each role follows this sequence separately before inference in Figure 1.

Thus a team never receives another team's response, rationale, confidence, or adapter. Direct emphasizes a decisive derivation; Evidence grounds its answer in the relevant definition, fact, or passage; Verification independently re-solves the question and checks alternatives. The exact instructions are in Appendix B.

Equation 1 defines the implemented prompt dependencies. Role prefixes are included in A_i and C_i, and $t = 0, \ldots, 3$ for the revision transition. The three pairs share pretrained weights and training questions, so their errors can remain correlated. Separate histories remove cross-pair text from generation; Section 4.3 measures the agreement that remains at the final decision.

3.2 Why Three Encapsulated Pairs?

Three is the smallest odd ensemble that supports a strict majority and a nontrivial diversity analysis. The roles vary the route to an answer while receiving the same full question and base prompt contract. This gives a fixed inference budget of three five-round pair trajectories followed by a vote.

3.3 Balanced Role Initialization

Training begins with role-specific Actor SFT (Figure 2). For each backbone, the base model generates candidates for 10,000 MMLU auxiliary-training questions under every role at temperatures 0.4, 0.7, and 1.0. A target is eligible only when its extracted answer is correct and its generation is not truncated. We keep at most one target per question–role pair and intersect question IDs across all three roles. Consequently, roles within a backbone receive the same questions, target count, and number of SFT updates. The retained count per role is 7,847 for Llama, 8,313 for Qwen2.5, 6,902 for Gemma, 7,755 for Phi, and 7,005 for Mistral. Critics start from the unadapted backbone.

3.4 Continuation-Valued Preference Learning

Every team processes all 1,531 MMLU validation questions, using five sampled trajectories per question for Mistral and one for each other backbone. It follows the ACC-Collab training order: construct Critic preferences, train the Critic, construct Actor preferences with that Critic, then train the Actor (Estornell et al., 2025). For an Actor state (x, a), the generator samples natural feedback c^0, feedback guided toward a correct answer c^+, and feedback guided toward an incorrect answer c^-. Candidate feedback is valued through $K = 10$ Actor continuations:

$$\widehat{R}(c \mid x, a) = \frac{1}{K} \sum_{k=1}^{K} \mathbf{1}[g(a'_k) = y], \qquad a'_k \sim A_i(\cdot \mid x, a, c). \tag{2}$$

The ordered rule retains (c^+, c^0) when its reward gap is at least $\epsilon = 0.6$; otherwise it retains (c^0, c^-) when that gap is at least ϵ. Each state therefore contributes at most one Critic comparison, and weak contrasts are discarded. After Critic DPO, Actor candidates are valued by sampling natural Critic feedback followed by an Actor continuation. The Actor stage uses the same margin rule. The score ties feedback quality to the Actor's next answer. Gold labels supply training supervision; inference follows the private path in Figure 1.

For either stage, a retained preferred/dispreferred pair (u^+, u^-) is trained with DPO (Rafailov et al., 2023):

$$\mathcal{L}_{\mathrm{DPO}}(\theta) = -\mathbb{E} \log \sigma \left(\beta \left[\log \frac{\pi_\theta(u^+ \mid s)}{\pi_{\mathrm{ref}}(u^+ \mid s)} - \log \frac{\pi_\theta(u^- \mid s)}{\pi_{\mathrm{ref}}(u^- \mid s)} \right] \right) + \lambda \mathcal{L}_{\mathrm{NLL}}, \tag{3}$$

with $\beta = 0.1$ and $\lambda = 1$. Here s is the stage-specific prompt, π_{ref} is the fixed reference policy, and $\mathcal{L}_{\mathrm{NLL}}$ is the mean token negative log-likelihood of the preferred completion u^+. Actor DPO starts from the role-SFT adapter. Source questions are not partitioned among roles: every team receives the complete source set but produces independent trajectories and a data-dependent number of retained pairs.

3.5 Judge-Free Late Fusion

After round 4, the system counts valid answers. If an answer receives at least two votes, it is returned; otherwise the Direct answer is the fixed fallback:

$$\widehat{y} = \begin{cases} m, & |\{i : z_i = m\}| \geq 2, \\ z_D, & \text{otherwise.} \end{cases} \tag{4}$$

The fallback is fixed before evaluation. Critic text is never a vote, and no model is invoked to adjudicate disagreements. Compared with one ACC-Collab team, SEPAL uses three pair trajectories; the vote adds negligible cost.

4 Experiments

4.1 Experimental Setup

Models, data, and benchmarks. We evaluate Meta-Llama-3-8B-Instruct (Grattafiori et al., 2024), Qwen2.5-3B-Instruct (Yang et al., 2024), Gemma-2-2B-it (Gemma Team et al., 2024), Phi-4-mini-instruct (Microsoft et al., 2025), and Mistral-7B-Instruct-v0.3 (Jiang et al., 2023a). MMLU contains 57 academic subjects (Hendrycks et al., 2021). BoolQ is yes/no reading comprehension (Clark et al., 2019); BBH collects challenging BIG-Bench tasks (Suzgun et al., 2023); SciQ is multiple-choice science QA (Welbl et al., 2017); and ARC combines the official Easy and Challenge test splits (Clark et al., 2018). We use the available examples for four datasets and a category-stratified 1,260-example BBH subset; only MMLU supplies optimization data.

Baselines and metric. Direct is one response from the unadapted instruction model. Debate uses an untrained Actor and Critic for the same five-round protocol. SoM-2 and SoM-4 use two or four symmetric untrained agents that observe peer responses; the metric is mean final-agent accuracy, matching the locked evaluation protocol. ACC is a single trained Actor–Critic pair using the same MMLU source data, parser, splits, and hyperparameters as SEPAL. The primary metric is exact match after task-aware normalization, and macro accuracy is the unweighted mean over five datasets. All comparisons are descriptive, using one fixed realization per model–method–dataset cell under the recorded protocol.

Dataset	Split	N	Use
MMLU	test	14,042	in-domain
BoolQ	validation	3,270	transfer
BBH	22-category stratified	1,260	transfer
SciQ	test	1,000	transfer
ARC	Easy + Challenge test	3,548	transfer

Table 1: Evaluation datasets. Transfer rows supply no training examples.

Model	Dataset	Direct	Debate	SoM-2	SoM-4	ACC	SEPAL
Llama-3-8B	BoolQ	77.34	76.76	**78.98**	78.74	76.54	76.70
	MMLU	62.41	63.55	63.39	63.33	65.00	**66.93**
	BBH	49.52	50.00	50.52	51.31	53.17	**56.67**
	SciQ	91.90	92.00	92.45	92.03	91.50	**93.40**
	ARC	88.30	88.92	89.04	88.65	89.04	**90.78**
	Macro	73.89	74.25	74.87	74.81	75.05	**76.90**
Qwen2.5-3B	BoolQ	65.17	69.30	67.58	68.21	73.30	**77.71**
	MMLU	65.80	65.67	65.81	66.04	67.45	**68.37**
	BBH	50.87	49.60	53.45	54.86	51.98	**55.63**
	SciQ	**92.70**	92.30	92.05	91.77	91.80	91.90
	ARC	89.49	90.90	89.56	90.17	90.84	**92.42**
	Macro	72.81	73.55	73.69	74.21	75.07	**77.21**
Gemma-2-2B	BoolQ	71.47	79.85	**80.64**	79.61	80.40	80.24
	MMLU	56.48	57.68	58.19	58.12	58.62	**59.50**
	BBH	41.83	40.16	38.21	37.48	42.54	**44.05**
	SciQ	89.20	90.80	90.65	90.38	89.90	**91.80**
	ARC	84.67	86.78	**87.02**	86.94	85.82	87.01
	Macro	68.73	71.05	70.94	70.50	71.46	**72.52**
Phi-4-mini	BoolQ	70.31	80.40	84.16	84.01	82.97	**85.26**
	MMLU	67.29	70.25	69.37	70.17	71.41	**73.04**
	BBH	52.78	56.03	55.95	57.40	57.46	**60.08**
	SciQ	90.70	92.30	90.05	91.27	91.80	**92.90**
	ARC	89.88	92.70	91.05	91.92	91.80	**93.21**
	Macro	74.19	78.34	78.12	78.95	79.09	**80.90**
Mistral-7B	BoolQ	75.78	79.20	79.48	79.12	80.18	**83.88**
	MMLU	56.95	59.17	58.10	57.90	59.69	**62.09**
	BBH	42.14	44.13	44.17	44.03	46.35	**48.10**
	SciQ	85.10	84.90	86.85	86.83	88.60	**89.10**
	ARC	83.77	84.58	84.72	85.03	84.24	**87.01**
	Macro	68.75	70.40	70.66	70.58	71.81	**74.03**

Table 2: Accuracy (%). ACC is the matched single-team implementation and SEPAL uses three teams. Black bold marks each row's best value; pale blue identifies SEPAL throughout. Macro averages the five datasets.

Implementation. We use model-native chat templates, bfloat16 weights, and vLLM (Kwon et al., 2023) on up to four 80 GB NVIDIA A800 GPUs. Exact generation limits, optimizer settings, adapter configuration, seeds, prompts, and answer handling are recorded in Appendix A. The adapters follow the LoRA parameterization (Hu et al., 2022). The matched ACC baseline uses one Actor–Critic trajectory, whereas SEPAL runs three role-local trajectories before voting; other baselines retain their locked call patterns. The main comparison therefore follows each method's recorded protocol rather than imposing an artificial equal-call budget.

These choices make the evaluation answer three connected questions. Does late fusion improve on a matched learned pair? Do private trajectories retain useful alternatives at the decision point? Which

part of the training and revision cycle produces the gain? The main table answers the first question; the decision, ablation, and revision analyses trace each role's trajectory back to the final vote.

4.2 Main Results

The first question has a consistent answer. Table 2 shows that SEPAL exceeds matched ACC in 24 of 25 model–dataset cells. Macro accuracy improves for all five backbones, by 1.06–2.22 points and 1.81 points on average. Phi attains the highest absolute macro accuracy (80.90), while Mistral has the largest gain over ACC (+2.22). Gemma BoolQ is the only negative cell (-0.15 points).

The improvement also transfers beyond the optimization distribution. Training uses MMLU only, while BBH, SciQ, and ARC improve over ACC for all five backbones; MMLU itself improves in every case. BoolQ is mixed. Lower-cost baselines lead on four individual rows: SoM-2 on Llama BoolQ, Gemma BoolQ, and Gemma ARC, and Direct on Qwen2.5 SciQ. These baselines use their own recorded inference protocols, so the comparison reflects both accuracy and the practical cost of the full recorded protocol.

The result is therefore not explained by a uniformly stronger individual role. Some cheaper baselines remain best on individual cells, while the three-role vote raises the macro score for every backbone. We next examine whether that gain comes from retaining different answers until the final decision.

4.3 Decision Analysis

Aggregate accuracy cannot distinguish complementarity from three copies of one policy. We therefore audit majority coverage, unanimity, oracle-any-role accuracy, fallback use, and per-role accuracy for every final decision record.

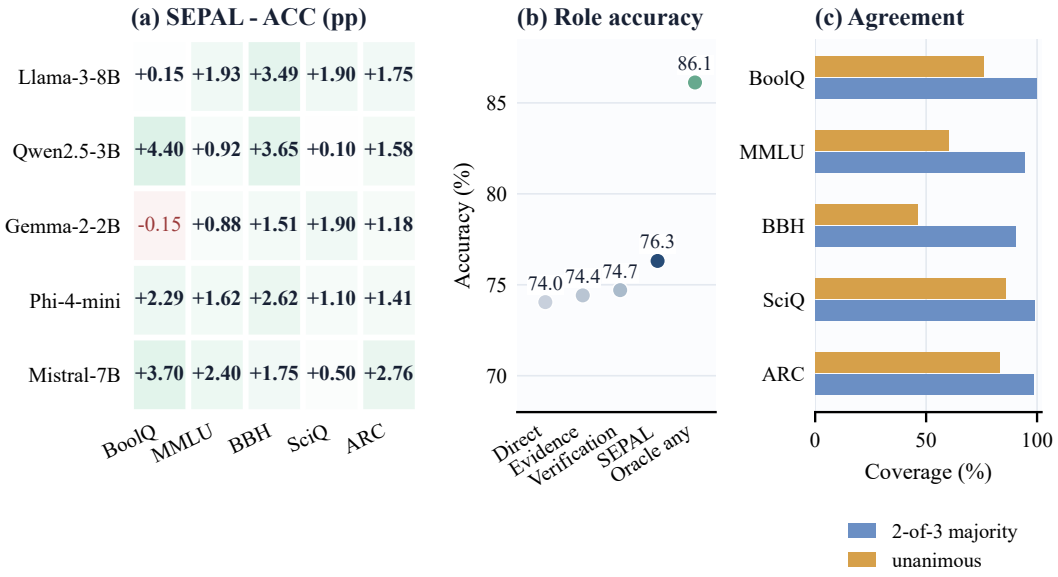

Figure 3: **Decision diagnostics across all 25 cells.** (a) Accuracy change from ACC to SEPAL. (b) Mean role, vote, and oracle-any-role accuracy. (c) Majority coverage and unanimity by dataset, averaged over backbones.

Figure 3 shows how these gains reach the final decision. Direct, Evidence, and Verification average 74.05, 74.42, and 74.71% accuracy; the vote reaches 76.31%, exceeding the strongest role in 21 of 25 cells and by 0.62 points on average.

Consensus and unanimity are different. A two-of-three answer exists for 96.41% of examples on average, so the Direct fallback is used only 3.59% of the time. Yet all three roles agree on 70.34%. BBH is the clearest case: majority coverage is 90.51%, while unanimity is only 46.27%. The system can therefore make a stable decision while retaining substantial role-level variation. Oracle-any-role

accuracy reaches 86.12%, exposing a 9.81-point selection gap. This gap quantifies the opportunity for a future calibrated router or verifier evaluated on a separate validation protocol.

Pairwise agreement covers 78.70–79.22% of examples, with shared-answer accuracy of 81.80–82.09%. Thus pairs retain different answers on roughly one fifth of questions, while agreement predicts greater reliability. Appendix F gives the full role and decision statistics.

The vote improves on the individual roles, while the oracle gap shows that useful answers still go unselected. We now turn to the candidates themselves and examine which stages of training and revision make them more accurate.

4.4 Ablations

We first ask whether the gain comes from stronger one-shot Actors or from interaction with Critics. Table 3 follows the same three-role team through initialization, preference learning, and revision. SFT-only Actors provide the starting point. SFT+Base-C and SFT+Trained-C add private feedback from a base or preference-trained Critic, respectively. Full-R0 evaluates the fully trained Actors before feedback, and Full-R4 includes all four revisions. No-SFT removes role initialization from the full pipeline. The final vote and all 25 evaluation cells are shared across variants.

Model	SFT	Base-C	Trained-C	Full-R0	No-SFT	Full-R4
Llama-3-8B	71.95	76.35	76.72	72.27	**77.79**	76.90
Qwen2.5-3B	75.57	76.71	**77.79**	75.98	75.98	77.21
Gemma-2-2B	69.58	71.96	72.31	69.70	71.98	**72.52**
Phi-4-mini	74.37	80.29	**81.03**	75.41	80.41	80.90
Mistral-7B	70.20	73.38	73.66	69.60	73.65	**74.03**
Mean	72.34	75.74	76.30	72.59	75.96	**76.31**

Table 3: Macro accuracy (%) over five datasets. Black bold marks the best variant for each backbone; pale blue identifies Full-R4.

Critic feedback produces the largest improvement (Figure 4). The same trained Actors gain +3.72 macro points from Full-R0 to Full-R4, with improvements in 23 of 25 cells. Feedback is already useful without Critic preference learning. SFT+Base-C improves on SFT-only by +3.40 points, also in 23 cells. By comparison, Full-R0 gains just +0.26 points over SFT-only before any feedback, improving 17 cells. Better initial answers account for only a small part of the full system's gain.

Preference learning has a smaller effect once feedback is available. Replacing the base Critic with the trained Critic adds +0.57 points on average; subsequently training the Actor adds +0.01. These averages hide differences among backbones. SFT+Trained-C leads on Qwen2.5 and Phi, while No-SFT leads on Llama. The complete pipeline improves on SFT-only by +3.97 points and in 24 of 25 cells, but the best configuration depends on the backbone.

These Actor comparisons address different questions. Full-R0 versus SFT-only measures the update before any feedback. Full-R4 versus SFT+Trained-C measures the same update when a trained Critic can respond. Reading the two together separates initial-answer quality from the benefit that remains after feedback.

All variants keep the three private role histories and the same final vote. The ablations therefore compare candidate generation under a fixed fusion rule. The remaining question is how much further improvement comes from repeating the revision step. All contrasts use unrounded aggregates; Appendix E gives the full per-dataset results.

4.5 Revision Dynamics

How long should a team keep revising? The first Critic-conditioned exchange raises mean accuracy from 72.59 to 75.91%, capturing 89.2% of the final 3.72-point gain. The released decision audit separates two processes behind this trajectory. At R1, 6.50% of all examples change from wrong to right, while 3.19% change from right to wrong. By R4, these fractions reach 7.79% and 4.07%. Every transition is measured against the same R0 decision.

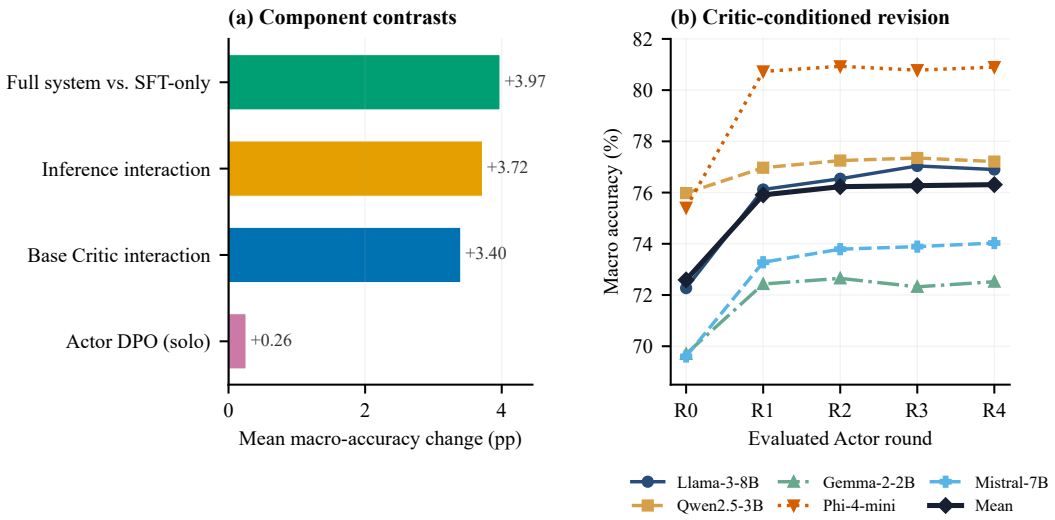

Figure 4: Component and round audit. Left: observed macro-accuracy contrasts averaged over five backbones. Right: majority-vote macro accuracy by Actor round, with the first Critic-conditioned revision at R1.

From R1 to R4, the corrected fraction rises by 1.29 percentage points and the regressed fraction by 0.89 points, for a net gain of 0.40 points. These changes use unrounded cell averages against the same R0 baseline. A useful stopping policy must account for both effects, since additional feedback can rescue an unresolved error or overturn an answer that was already correct.

Later rounds keep correcting errors, but increasingly offset those repairs with regressions. The net benefit also depends on the dataset. From R1 to R4, BoolQ's gain over R0 grows from 5.68 to 6.91 points; BBH's falls from 4.02 to 3.33. The common R4 protocol therefore leaves room for a stopping rule validated on held-out data. The released decisions preserve all these transitions, including the changes that reduce accuracy.

All methods use model-native chat templates and the same answer extractor. Appendix G describes parser corrections and uniform re-scoring of the cached raw generations.

5 Discussion and Conclusion

The decision records separate two ways a team can fail. When all three roles are wrong, stronger candidates or external evidence are needed. When a correct role loses the vote, selection is the bottleneck. The 9.81-point oracle-any-role gap identifies room for a router trained on disjoint validation data. A selector restricted to the three final answers cannot exceed this oracle, giving future routing experiments a measurable ceiling. Correction and regression counts provide a second target: deciding when to stop revising. Private histories make both interventions traceable to the candidates that produced the final answer.

Limitations. SEPAL uses three pair trajectories and reports one fixed run per cell on five 2B–8B backbones and short-answer QA tasks. Repeated training seeds, comparisons at equal compute, and a direct communication-boundary ablation would clarify robustness and the source of the ensemble gain.

SEPAL keeps critique inside three private Actor–Critic teams and fuses only their final answers. Across five backbones, it improves macro accuracy over matched single-team ACC-Collab and wins 24 of 25 model–dataset comparisons. Critic-conditioned revision produces the largest measured component gain, most of it after the first exchange. Late voting then improves on the strongest role in most evaluated cells. The resulting recipe is to revise candidates locally and fuse completed answers. Released records trace repairs and selection errors to individual questions.

ETHICS STATEMENT

The study uses public benchmark questions and open-weight language models; it does not involve human subjects or newly collected personal data. The method can nevertheless inherit factual errors, social biases, and unsafe behavior from its backbones, and majority agreement is not a guarantee of truth. Its additional computation also has environmental cost. The reported system should not be used as an autonomous decision maker in high-stakes settings without domain-specific validation and human oversight.

REPRODUCIBILITY STATEMENT

The supplementary package records exact checkpoint identifiers, data splits, role prompts, hyperparameters, training yields, parser policy, and the complete per-dataset result matrix. It includes the source snapshot used by the pipeline, resolved configuration, plotting scripts, locked CSV files, and a machine-readable file inventory. Appendix A describes the evaluation and optimization protocol, while Appendix E reports every ablation cell. All citation metadata has a machine-readable two-source audit with retrieval dates.

AI USE STATEMENT

During the experiments, we used large language models (LLMs) as an auxiliary tool for experiment monitoring and management. Specifically, the LLMs were used to monitor experiment logs and runtime information, identify potential execution issues or anomalies, and assist in reporting the status of ongoing experiments. The LLMs did not determine the research questions, experimental methodology, hyperparameter settings, or final experimental conclusions. All experimental configurations, result verification, analysis, and scientific conclusions were determined and validated by the authors. We take full responsibility for the final content and results of this work.

REFERENCES

Chi-Min Chan, Weize Chen, Yusheng Su, Jianxuan Yu, Wei Xue, Shanghang Zhang, Jie Fu, and Zhiyuan Liu. ChatEval: Towards better LLM-based evaluators through multi-agent debate. In *International Conference on Learning Representations (ICLR)*, 2024. URL https://openreview.net/forum?id=FQepisCUWu.

Justin Chen, Swarnadeep Saha, and Mohit Bansal. ReConcile: Round-table conference improves reasoning via consensus among diverse LLMs. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 7066–7085, 2024a. doi: 10.18653/v1/2024.acl-long.381. URL https://aclanthology.org/2024.acl-long.381/.

Pei Chen, Shuai Zhang, and Boran Han. CoMM: Collaborative multi-agent, multi-reasoning-path prompting for complex problem solving. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pp. 1720–1738. Association for Computational Linguistics, 2024b. doi: 10.18653/v1/2024.findings-naacl.112. URL https://aclanthology.org/2024.findings-naacl.112/.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. BoolQ: Exploring the surprising difficulty of natural yes/no questions. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 2924–2936. Association for Computational Linguistics, 2019. doi: 10.18653/v1/N19-1300. URL https://aclanthology.org/N19-1300/.

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. *CoRR*, abs/1803.05457, 2018. URL http://arxiv.org/abs/1803.05457.

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

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 (ICLR)*, 2025. URL https://openreview.net/forum?id=nfKfAzkiez.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, et al. Gemma 2: Improving open language models at a practical size. *CoRR*, abs/2408.00118, 2024. doi: 10.48550/ARXIV.2408.00118. URL https://doi.org/10.48550/arXiv.2408.00118.

Zhibin Gou, Zhihong Shao, Yeyun Gong, Yelong Shen, Yujiu Yang, Nan Duan, and Weizhu Chen. CRITIC: Large language models can self-correct with tool-interactive critiquing. In *The Twelfth International Conference on Learning Representations*. OpenReview.net, 2024. URL https://openreview.net/forum?id=Sx038qxjek.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The Llama 3 herd of models. *CoRR*, abs/2407.21783, 2024. doi: 10.48550/ARXIV.2407.21783. URL https://doi.org/10.48550/arXiv.2407.21783.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations (ICLR)*, 2021. URL https://openreview.net/forum?id=d7KBjmI3GmQ.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations (ICLR)*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Wei Yu, Xinying Song, and Denny Zhou. Large language models cannot self-correct reasoning yet. In *The Twelfth International Conference on Learning Representations*. OpenReview.net, 2024. URL https://openreview.net/forum?id=IkmD3fKBPQ.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7B, 2023a. URL https://arxiv.org/abs/2310.06825.

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. Association for Computational Linguistics, 2023b. doi: 10.18653/v1/2023.acl-long.792. URL https://aclanthology.org/2023.acl-long.792/.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with PagedAttention. In *Proceedings of the Symposium on Operating Systems Principles (SOSP)*, pp. 611–626, 2023. doi: 10.1145/3600006.3613165. URL https://doi.org/10.1145/3600006.3613165.

Junyou Li, Qin Zhang, Yangbin Yu, Qiang Fu, and Deheng Ye. More agents is all you need. *Trans. Mach. Learn. Res.*, 2024. URL https://openreview.net/forum?id=bgzUSZ8aeg.

Tian Liang, Zhiwei He, Wenxiang Jiao, Xing Wang, Yan Wang, Rui Wang, Yujiu Yang, Shuming Shi, and Zhaopeng Tu. Encouraging divergent thinking in large language models through multi-agent debate. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 17889–17904, 2024. doi: 10.18653/v1/2024.emnlp-main.992. URL https://aclanthology.org/2024.emnlp-main.992/.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. Self-Refine: Iterative refinement with self-feedback. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. doi: 10.52202/075280-2019. URL https://doi.org/10.52202/075280-2019.

Microsoft, Abdelrahman Abouelenin, Atabak Ashfaq, Adam Atkinson, Hany Awadalla, Nguyen Bach, Jianmin Bao, Alon Benhaim, Martin Cai, Vishrav Chaudhary, Congcong Chen, Dong Chen, Dongdong Chen, Junkun Chen, Weizhu Chen, Yen-Chun Chen, Yi-ling Chen, Qi Dai, Xiyang Dai, Ruchao Fan, Mei Gao, Min Gao, Amit Garg, Abhishek Goswami, Junheng Hao, Amr Hendy, Yuxuan Hu, Xin Jin, Mahmoud Khademi, Dongwoo Kim, Young Jin Kim, Gina Lee, Jinyu Li, Yunsheng Li, Chen Liang, Xihui Lin, Zeqi Lin, Mengchen Liu, Yang Liu, Gilsinia Lopez, Chong Luo, Piyush Madan, Vadim Mazalov, Arindam Mitra, Ali Mousavi, Anh Nguyen, Jing Pan, Daniel Perez-Becker, Jacob Platin, Thomas Portet, Kai Qiu, Bo Ren, Liliang Ren, Sambuddha Roy, Ning Shang, Yelong Shen, Saksham Singhal, Subhojit Som, Xia Song, Tetyana Sych, Praneetha Vaddamanu, Shuohang Wang, Yiming Wang, Zhenghao Wang, Haibin Wu, Haoran Xu, Weijian Xu, Yifan Yang, Ziyi Yang, Donghan Yu, Ishmam Zabir, Jianwen Zhang, Li Lyna Zhang, Yunan Zhang, and Xiren Zhou. Phi-4-Mini technical report: Compact yet powerful multimodal language models via Mixture-of-LoRAs, 2025. URL https://arxiv.org/abs/2503.01743.

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. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. doi: 10.52202/075280-2338. URL https://doi.org/10.52202/075280-2338.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. doi: 10.52202/075280-0377. URL https://doi.org/10.52202/075280-0377.

Vighnesh Subramaniam, Yilun Du, Joshua B. Tenenbaum, Antonio Torralba, Shuang Li, and Igor Mordatch. Multiagent finetuning: Self improvement with diverse reasoning chains. In *International Conference on Learning Representations (ICLR)*, 2025. URL https://openreview.net/forum?id=JtGPIZpOrz.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V. Le, Ed H. Chi, Denny Zhou, and Jason Wei. Challenging BIG-Bench tasks and whether chain-of-thought can solve them. In *Findings of the Association for Computational Linguistics (ACL)*, pp. 13003–13051, 2023. doi: 10.18653/V1/2023.FINDINGS-ACL.824. URL https://doi.org/10.18653/v1/2023.findings-acl.824.

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 (ICLR)*, 2025. URL https://iclr.cc/virtual/2025/poster/28787.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *International Conference on Learning Representations (ICLR)*, 2023. URL https://openreview.net/forum?id=1PL1NIMMrw.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. doi: 10.52202/068431-1800. URL https://doi.org/10.52202/068431-1800.

Johannes Welbl, Nelson F. Liu, and Matt Gardner. Crowdsourcing multiple choice science questions. In *Proceedings of the 3rd Workshop on Noisy User-generated Text*, pp. 94–106. Association for Computational Linguistics, 2017. doi: 10.18653/v1/W17-4413. URL https://aclanthology.org/W17-4413/.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report. *CoRR*, abs/2412.15115, 2024. doi: 10.48550/ARXIV.2412.15115. URL https://doi.org/10.48550/arXiv.2412.15115.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. In *Advances in Neural Information Processing Systems*, volume 36, pp. 11809–11822. Curran Associates, Inc., 2023. doi: 10.52202/075280-0517. URL https://doi.org/10.52202/075280-0517.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah D. Goodman. STaR: Bootstrapping reasoning with reasoning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. doi: 10.52202/068431-1126. URL https://doi.org/10.52202/068431-1126.

A Reproducibility Contract

This appendix records the exact contract behind every result cell. The contract has four identities: (i) a model checkpoint and adapter state, (ii) a dataset split and expected sample count, (iii) a prompt, parser, and decision rule, and (iv) a resolved training or evaluation configuration. A run is considered comparable only when all four identities match the released locked record.

A.1 Checkpoints and Data Lifecycle

Table 4 records the five checkpoint identities.

Backbone	Checkpoint identifier
Llama-3-8B	`meta-llama/Meta-Llama-3-8B-Instruct`
Qwen2.5-3B	`Qwen/Qwen2.5-3B-Instruct`
Gemma-2-2B	`google/gemma-2-2b-it`
Phi-4-mini	`microsoft/Phi-4-mini-instruct`
Mistral-7B	`mistralai/Mistral-7B-Instruct-v0.3`

Table 4: Public checkpoint identities used in every reported run.

Deployments used offline mirrors with the same model configuration and weights. The three phases consume disjoint dataset uses:

1. **Role initialization.** For each backbone, 10,000 questions are sampled from MMLU `auxiliary_train`. Candidate responses are generated separately under the Direct, Evidence, and Verification prefixes.
2. **Preference construction.** All Critic and Actor comparisons use the 1,531-question MMLU validation split. Every role receives the complete source set, including every subject.
3. **Evaluation.** MMLU uses its 14,042-item test split. BoolQ, SciQ, and ARC use the official splits listed in Table 1. BBH uses a 1,260-item subset sampled at approximately 25% from each of 22 evaluable categories.

Only MMLU supplies optimization examples. BoolQ, BBH, SciQ, and ARC are transfer evaluations. Gold evaluation labels are passed to the scorer, never to the generation prompt, role selector, vote, or fallback. Direct is fixed before each full-system evaluation.

A.2 Generation and Optimization Settings

Table 5 expands the compact implementation paragraph in the main text. Generation uses model-native chat templates with thinking mode disabled and bfloat16 weights. Phi uses a 4,096-token model context; the other backbones use 8,192 tokens. The DPO runner enforces a 4,096-token total training limit. The recorded 3,072/1,024 prompt/completion budgets are configuration metadata; the released TRL runner applies the total limit. Both role SFT and DPO use LoRA rank 256 and scaling factor 512.

Setting	Role SFT	Preference / DPO	Evaluation
Source split	MMLU auxiliary train	MMLU validation	benchmark-specific
Source questions	10,000	1,531	all configured
Generation temperature	$\{0.4, 0.7, 1.0\}$	0.7	0.7
Top-p	0.9	0.9	0.9
Maximum new tokens	1,024	1,024	1,024
Epochs	1	3	Not used
Learning rate	5.0×10^{-5}	1.41×10^{-5}	Not used
Effective batch size	16	4	Not used
Warmup ratio	0.10	0.03	Not used
Weight decay	0.01	0.10	Not used
Maximum gradient norm	1.0	0.3	Not used
LoRA rank / scaling	256 / 512	256 / 512	loaded adapters
DPO β / NLL weight	Not used	0.1 / 1.0	Not used
Training token limit	4,096	4,096	Not used

Table 5: Resolved settings for role initialization, preference optimization, and evaluation.

Preference construction uses $K = 10$ Actor continuations per feedback candidate and margin $\epsilon = 0.6$. Mistral uses five independently sampled preference trajectories per source question; the other backbones use one. These are training trajectories, and every reported evaluation cell uses one trial. DPO uses sigmoid loss, AdamW, gradient checkpointing, and optimizer-state-aware resumption. SFT and DPO checkpoints retain only the latest scheduled state. No evaluation score is used to select an optimizer checkpoint.

Model	SFT / role	Critic pairs	Actor pairs
Llama-3-8B	7,847	827–2,041	437–828
Qwen2.5-3B	8,313	429–771	960–1,032
Gemma-2-2B	6,902	494–593	704–927
Phi-4-mini	7,755	276–533	880–975
Mistral-7B	7,005	1,077–2,223	2,811–4,813

Table 6: Observed optimization-data yields. Preference entries give the role-wise minimum–maximum under the fixed margin rule.

Balanced SFT retains only question IDs with an eligible target for every role within a backbone. Consequently, the three roles receive identical SFT counts and question identities even though their generated targets differ. Preference yields are allowed to differ because the continuation-valued margin is applied independently to each role state.

A.3 Runtime, Seeds, and Execution Order

The original environment used Python 3.10, vLLM inference, and up to four 80 GB NVIDIA A800 GPUs. Prefix caching is enabled; eager execution is disabled; Llama and Gemma allow 8,192 batched tokens and 128 sequences, while Qwen2.5, Phi, and Mistral allow 32,768 batched tokens and 256 sequences. The base seed is 42. Direct, Evidence, and Verification use role offsets 0, 10,000, and 20,000 so that they cannot accidentally share a random stream.

For every role, the order is fixed: build Critic preferences, train the Critic, build Actor preferences using that trained Critic, then train the Actor. At evaluation, all three role records are aligned by trial, sample index, sample identifier, task type, and gold-label signature before voting. Any length, identity, parser-fingerprint, or role-manifest mismatch causes aggregation to abort.

B Prompt Templates and Role Contracts

Each prompt consists of an ACC-Collab base template plus exactly one role prefix. The released prompt identity concatenates the base version, the role prefix version, and the role name. This prevents a prompt edit from being silently treated as the same experiment.

B.1 Base Actor and Critic Templates

For a multiple-choice item, the initial Actor receives this semantic template after its role prefix:

> "Please answer the following multiple choice question as accurately as possible. You must provide an extremely brief justification for your answer, and you must give your final answer as a letter by saying 'Final Answer:'. Question: *[question]*. Options: *[labeled choices]*."

Each revision prompt places the previous Actor answer before its paired Critic feedback:

> "Several people have provided answers to a multiple choice question. Person 0 said: *[previous Actor response]*. Person 1 said: *[paired Critic feedback]*. Take these answers into consideration, give an extremely brief justification, and state the final answer as a letter."

The natural Critic template asks for brief additional details that improve the correctness of the supplied Actor response. Guided preference candidates use the same question and response but explicitly request details supporting a target answer. Yes/no tasks substitute a constrained final answer of Yes or No and include the BoolQ passage when present. These templates are identical across roles; only the prefixes below change.

B.2 Exact Role Prefixes

Each following string follows the literal prefix `Role specialization` and the role name in parentheses, then a colon. Two newline characters separate this role instruction from the corresponding base prompt.

Direct Actor. *Solve directly and concisely. Identify the decisive fact or calculation and avoid adding speculative alternatives once the answer is supported.*

Direct Critic. *Check the decisive fact or calculation in the Actor response. Supply only brief missing details that improve correctness, following the original ACC-Collab critic style.*

Evidence Actor. *Ground the answer in the most relevant definition, fact, passage evidence, or domain principle before selecting the final option.*

Evidence Critic. *Check whether the Actor used the relevant evidence or principle correctly and add only brief corrective evidence in the original ACC-Collab critic style.*

Verification Actor. *Independently solve the problem, then verify the selected option against the main alternatives or likely failure mode before stating the final answer.*

Verification Critic. *Independently verify the Actor answer and its strongest alternative, then provide only brief details that improve correctness in the original ACC-Collab critic style.*

The instructions specialize how a pair approaches a question; they do not grant external tools, retrieval, private reference material, different answer choices, or different token budgets. Critic prefixes also preserve the base contract that feedback should be terse and corrective rather than a second full answer presented to another role.

C Algorithms and Computational Structure

C.1 Continuation-Valued Preference Construction

Algorithm 1: one role-local preference stage.
Input: state $s = (x, a)$, gold answer y, Actor A_i, Critic C_i, extractor g, rollout count $K = 10$, margin $\epsilon = 0.6$.
1. Generate natural feedback c^0, correct-guided feedback c^+, and incorrect-guided feedback c^-.
2. For each c, sample K one-step Actor continuations and compute $\widehat{R}(c \mid s) = K^{-1} \sum_k \mathbf{1}[g(a'_k) = y]$.
3. If $\widehat{R}(c^+) - \widehat{R}(c^0) \geq \epsilon$, retain (c^+, c^0); else if $\widehat{R}(c^0) - \widehat{R}(c^-) \geq \epsilon$, retain (c^0, c^-); otherwise discard the state.
4. Train C_i on retained Critic pairs. Rebuild comparisons for Actor candidates using feedback from the trained C_i, then train A_i with the same ordered margin rule.
Output: one independently adapted Actor–Critic pair.

Figure 5: Role-local preference construction. The ordered if/else rule permits at most one retained comparison per state and stage.

The guided candidates are used only to construct preference comparisons; they are not evaluation-time hints. Continuation correctness evaluates whether a piece of feedback helps the Actor reach the gold answer, rather than whether the feedback text resembles a written reference critique.

C.2 Private Revision and Late Fusion

Algorithm 2: SEPAL inference.
Input: question x; pairs (A_i, C_i) for $i \in \{D, E, V\}$; extractor g.
1. In parallel, each Actor produces a_i^0 from x and its role prefix; its paired Critic returns c_i^0.
2. For $t = 1, \ldots, 4$, each Actor revises using only $(x, a_i^{t-1}, c_i^{t-1})$; its paired Critic reviews the new answer.
3. Parse $z_i = g(a_i^4)$. Do not expose a_i^t, c_i^t, or z_i to another pair during Steps 1–3.
4. If a valid answer occurs at least twice among $\{z_D, z_E, z_V\}$, return it. Otherwise return the parsed Direct answer.
Output: final answer $\widehat{y}$ and an auditable decision source (unanimous, majority, or fixed Direct fallback).

Figure 6: Inference contains no learned Judge and no cross-pair message.

With five Actor rounds and one Critic response after each Actor response, one ACC pair uses five Actor and five Critic generations. SEPAL uses three such trajectories: 15 Actor and 15 Critic generations, parallelizable across roles. Answer extraction and voting are deterministic string operations. Direct uses one Actor generation; SoM and Debate follow their locked protocols and are not normalized to the same number of generated tokens.

D Complete Main Comparisons

Table 7 expands the headline comparison into the exact sample count, ACC accuracy, SEPAL accuracy, and difference for every cell. The better accuracy in each row is black bold; positive differences appear in mint and the single negative cell in pale red.

Model	Dataset	N	ACC	SEPAL	Δ
Llama-3-8B	BoolQ	3,270	76.54	**76.70**	**+0.15**
	MMLU	14,042	65.00	**66.93**	**+1.93**
	BBH	1,260	53.17	**56.67**	**+3.49**
	SciQ	1,000	91.50	**93.40**	**+1.90**
	ARC	3,548	89.04	**90.78**	**+1.75**
Qwen2.5-3B	BoolQ	3,270	73.30	**77.71**	**+4.40**
	MMLU	14,042	67.45	**68.37**	**+0.92**
	BBH	1,260	51.98	**55.63**	**+3.65**
	SciQ	1,000	91.80	**91.90**	**+0.10**
	ARC	3,548	90.84	**92.42**	**+1.58**
Gemma-2-2B	BoolQ	3,270	**80.40**	80.24	-0.15
	MMLU	14,042	58.62	**59.50**	**+0.88**
	BBH	1,260	42.54	**44.05**	**+1.51**
	SciQ	1,000	89.90	**91.80**	**+1.90**
	ARC	3,548	85.82	**87.01**	**+1.18**
Phi-4-mini	BoolQ	3,270	82.97	**85.26**	**+2.29**
	MMLU	14,042	71.41	**73.04**	**+1.62**
	BBH	1,260	57.46	**60.08**	**+2.62**
	SciQ	1,000	91.80	**92.90**	**+1.10**
	ARC	3,548	91.80	**93.21**	**+1.41**
Mistral-7B	BoolQ	3,270	80.18	**83.88**	**+3.70**
	MMLU	14,042	59.69	**62.09**	**+2.40**
	BBH	1,260	46.35	**48.10**	**+1.75**
	SciQ	1,000	88.60	**89.10**	**+0.50**
	ARC	3,548	84.24	**87.01**	**+2.76**

Table 7: Complete matched comparison (accuracy, %). The five rows within a model use different evaluation sets but the same trained policy family and decision rule.

All 25 cells contain measured values; no missing cell is copied, interpolated, or replaced by a macro average. The four transfer datasets contribute 20 of the 25 comparisons, of which 19 improve over ACC. MMLU improves for every backbone, while training uses only MMLU questions.

E Complete Component and Round Results

The component comparisons follow recorded paths through the training pipeline. Pale blue identifies the reported Full-R4 row, while black bold identifies the strongest value in each comparable dataset column within a backbone. All six completed configurations are reported. SFT+Trained-C combines the role-SFT Actor with the trained Critic before Actor DPO. No-SFT starts Actors and Critics from the base model and retains the three roles, preference learning, private revision, and fixed vote. Full-R0 and Full-R4 use identical trained policies and differ only in the evaluated Actor round.

The additional controls show why the training stages should be distinguished. SFT+Trained-C reaches 77.79 on Qwen2.5 and 81.03 on Phi, exceeding Full-R4 by 0.58 and 0.14 macro points. No-SFT reaches 77.79 on Llama, exceeding Full-R4 by 0.89 points. Full-R4 is best on Gemma and Mistral and has the highest mean, 76.31, closely followed by SFT+Trained-C at 76.30. These descriptive results support private revision while showing that the preferred initialization and Actor update depend on the backbone.

The cell-level exceptions clarify the aggregate contrasts. Full-R4 improves on Full-R0 in 23 cells; the only decreases are Qwen2.5 BBH (-0.32 points) and Qwen2.5 SciQ (-0.30). Its largest revision gains occur on Phi BoolQ (+14.46), Mistral BBH (+6.12), and Qwen2.5 BoolQ (+5.60). Adding an untrained base Critic is similarly broad but not automatic: the SFT+Base-C contrast is negative on Gemma BBH (-1.03) and Qwen2.5 SciQ (-0.30). Full-R4 improves on SFT-only in 24 cells, with Qwen2.5 SciQ (-0.60) as the sole exception. Table 8 includes both exceptions and all 24 positive cells, and every reported mean uses the complete 25-cell matrix.

Model	Variant	BoolQ	MMLU	BBH	SciQ	ARC	Macro
Llama-3-8B	SFT-only	73.82	60.58	49.84	88.20	87.29	71.95
	SFT+Base-C	78.47	64.48	56.03	92.90	89.85	76.35
	SFT+Trained-C	78.99	65.57	55.71	92.80	90.53	76.72
	Full-R0	71.47	62.15	51.35	88.80	87.57	72.27
	No-SFT	**79.94**	**67.05**	**57.62**	**93.80**	90.53	**77.79**
	Full-R4	76.70	66.93	56.67	93.40	**90.78**	76.90
Qwen2.5-3B	SFT-only	73.70	66.76	53.73	92.50	91.18	75.57
	SFT+Base-C	76.39	66.93	56.43	92.20	91.57	76.71
	SFT+Trained-C	77.28	68.20	**57.46**	**94.00**	92.00	**77.79**
	Full-R0	72.11	68.36	55.95	92.20	91.29	75.98
	No-SFT	75.66	68.31	50.95	93.20	91.77	75.98
	Full-R4	**77.71**	**68.37**	55.63	91.90	**92.42**	77.21
Gemma-2-2B	SFT-only	74.77	56.00	42.70	90.10	84.33	69.58
	SFT+Base-C	**80.95**	58.55	41.67	91.60	87.03	71.96
	SFT+Trained-C	80.40	58.82	43.02	**92.10**	**87.20**	72.31
	Full-R0	75.17	57.37	41.98	89.50	84.47	69.70
	No-SFT	80.49	**59.85**	43.17	90.50	85.88	71.98
	Full-R4	80.24	59.50	**44.05**	91.80	87.01	**72.52**
Phi-4-mini	SFT-only	69.79	68.40	53.25	90.10	90.33	74.37
	SFT+Base-C	84.07	71.82	60.56	92.20	92.81	80.29
	SFT+Trained-C	85.57	72.23	**61.90**	92.70	92.76	**81.03**
	Full-R0	70.80	68.73	56.59	90.40	90.56	75.41
	No-SFT	**85.75**	72.26	58.89	92.50	92.64	80.41
	Full-R4	85.26	**73.04**	60.08	**92.90**	**93.21**	80.90
Mistral-7B	SFT-only	78.75	57.01	44.76	86.70	83.79	70.20
	SFT+Base-C	83.33	60.05	48.10	**89.30**	86.10	73.38
	SFT+Trained-C	83.36	60.72	**48.73**	88.60	86.89	73.66
	Full-R0	79.69	57.11	41.98	86.30	82.89	69.60
	No-SFT	**84.62**	61.19	47.94	88.20	86.30	73.65
	Full-R4	83.88	**62.09**	48.10	89.10	**87.01**	**74.03**

Table 8: Complete component matrix (accuracy, %). Black bold marks the best value per dataset and backbone; Macro weights all five datasets equally.

Model	R0	R1	R2	R3	R4
Llama-3-8B	72.27	76.12	76.54	**77.04**	76.90
Qwen2.5-3B	75.98	76.97	77.25	**77.35**	77.21
Gemma-2-2B	69.70	72.43	**72.65**	72.32	72.52
Phi-4-mini	75.41	80.73	**80.93**	80.78	80.90
Mistral-7B	69.60	73.28	73.79	73.89	**74.03**
Mean	72.59	75.91	76.23	76.27	**76.31**

Table 9: Round-wise macro accuracy (%). R1 is the first Critic-conditioned revision. Pale blue marks the fixed endpoint; black bold marks the best round.

Most of the five-round improvement arrives immediately. The mean gain is +3.32 points at R1, followed by +0.32 from R1 to R2, +0.04 from R2 to R3, and +0.04 from R3 to R4. The fixed R4 endpoint keeps one protocol across benchmarks, and the recorded curve motivates future validation-set stopping rules.

F Role and Decision Diagnostics

Table 10 reports the final accuracy of each role before fusion. The best role varies across cells. Verification has the highest mean, but Direct is strongest on several ARC and MMLU configurations,

and Evidence is strongest for Qwen2.5 on BoolQ, BBH, SciQ, and ARC. The fixed vote therefore does not reduce to selecting one globally dominant role.

Agreement varies much more than parse reliability. Unanimity ranges from 40.16% on Mistral BBH to 89.10% on Llama SciQ, while two-of-three coverage remains 87.46% even in the lowest-coverage cell. Parse rate never falls below 99.76%. The oracle-any-role gap ranges from 7.46 points for Phi to 10.83 for Mistral, so headroom is not an artifact of one weak backbone or one parser format.

Majority coverage is lowest on Qwen2.5 BBH (87.46%) and highest on Phi BoolQ (100.00%). Fallback behavior is concentrated on the more heterogeneous reasoning benchmark rather than uniformly distributed. The Evidence–Verification pair has the highest conditional accuracy (82.09%) and agreement coverage (79.22%), followed closely by Direct–Evidence at 79.17% coverage. Evidence–Verification leads both pairwise summaries.

Model	Dataset	Direct	Evidence	Verification	SEPAL	Oracle-any
Llama-3-8B	BoolQ	74.19	71.56	80.52	76.70	**88.26**
	MMLU	64.86	64.56	65.23	66.93	**80.98**
	BBH	53.81	52.06	56.19	56.67	**75.63**
	SciQ	92.80	91.80	91.20	93.40	**96.60**
	ARC	90.02	89.04	89.29	90.78	**95.29**
Qwen2.5-3B	BoolQ	68.96	79.51	77.49	77.71	**88.13**
	MMLU	65.05	67.28	66.14	68.37	**81.38**
	BBH	49.92	57.62	52.94	55.63	**75.16**
	SciQ	89.90	92.30	90.50	91.90	**96.50**
	ARC	90.28	90.95	89.04	92.42	**96.62**
Gemma-2-2B	BoolQ	76.45	78.69	79.39	80.24	**89.97**
	MMLU	57.97	57.81	58.57	59.50	**72.82**
	BBH	43.41	42.30	42.38	44.05	**60.32**
	SciQ	89.40	89.00	90.60	91.80	**96.30**
	ARC	84.78	84.61	85.79	87.01	**92.98**
Phi-4-mini	BoolQ	82.57	83.06	84.59	85.26	**91.56**
	MMLU	70.91	71.05	70.63	73.04	**83.24**
	BBH	59.21	56.67	57.94	60.08	**74.21**
	SciQ	92.20	92.20	90.50	92.90	**96.50**
	ARC	92.62	91.66	91.63	93.21	**96.28**
Mistral-7B	BoolQ	82.08	82.32	81.68	83.88	**91.41**
	MMLU	60.03	59.21	59.30	62.09	**76.78**
	BBH	47.38	44.76	44.68	48.10	**67.38**
	SciQ	88.00	85.80	86.90	89.10	**94.70**
	ARC	84.36	84.58	84.50	87.01	**94.05**

Table 10: Per-role, voted, and oracle-any-role accuracy (%). Pale blue marks the actual SEPAL decision; mint marks diagnostic oracle headroom.

Role pair	Agreement coverage	Accuracy when agreeing
Direct + Evidence	79.17	81.80
Direct + Verification	78.70	82.04
Evidence + Verification	**79.22**	**82.09**

Table 11: Pairwise final-answer agreement averaged over all 25 cells (%). Conditional accuracy evaluates the shared answer only on agreeing examples.

Across all cells, parse rates average 99.93%, so the agreement pattern is driven by answer differences rather than systematic extraction failure.

Model	Dataset	Majority	Unanimous	Oracle-any	Fallback	Parsed
Llama-3-8B	BoolQ	99.42	73.06	**88.26**	0.58	99.79
	MMLU	93.51	58.48	**80.98**	6.49	99.94
	BBH	89.52	43.97	**75.63**	10.48	99.76
	SciQ	99.20	89.10	**96.60**	0.80	100.00
	ARC	98.70	85.96	**95.29**	1.30	100.00
Qwen2.5-3B	BoolQ	99.88	71.90	**88.13**	0.12	100.00
	MMLU	94.52	60.52	**81.38**	5.48	99.99
	BBH	87.46	43.10	**75.16**	12.54	100.00
	SciQ	99.40	86.60	**96.50**	0.60	100.00
	ARC	98.62	83.74	**96.62**	1.38	100.00
Gemma-2-2B	BoolQ	99.91	74.31	**89.97**	0.09	99.94
	MMLU	95.20	59.98	**72.82**	4.80	99.81
	BBH	94.68	47.62	**60.32**	5.32	99.84
	SciQ	98.40	84.60	**96.30**	1.60	99.80
	ARC	98.22	80.61	**92.98**	1.78	99.92
Phi-4-mini	BoolQ	100.00	81.83	**91.56**	0.00	100.00
	MMLU	96.59	67.93	**83.24**	3.41	99.92
	BBH	92.54	56.51	**74.21**	7.46	100.00
	SciQ	99.50	88.50	**96.50**	0.50	100.00
	ARC	99.35	89.32	**96.28**	0.65	99.97
Mistral-7B	BoolQ	99.91	79.24	**91.41**	0.09	99.94
	MMLU	92.29	53.26	**76.78**	7.71	99.96
	BBH	88.33	40.16	**67.38**	11.67	99.76
	SciQ	97.50	81.30	**94.70**	2.50	99.90
	ARC	97.63	76.97	**94.05**	2.37	99.94

Table 12: Complete final-round decision diagnostics (%). Majority is the fraction with a valid two-of-three answer; Unanimous requires all three normalized answers to agree; Fallback is the complement of Majority.

G Answer Extraction and Evaluation Integrity

All methods share a deterministic, dataset-aware answer extractor. It applies the following normalization rules in order.

1. remove model-specific thinking blocks without editing visible answer text;
2. prioritize explicit final-answer markers near the end of the response;
3. for multiple-choice tasks, extract an option letter by checking final-marker, tail-claim, then weak-tail patterns;
4. for yes/no tasks, accept constrained final labels and label-plus-text forms such as "A: Yes" and "B: No"; and
5. select the last recognized match within the highest-priority pattern group, or return an invalid parse when no pattern matches.

The test suite covers answer-marker precedence, case normalization, punctuation, option-letter extraction, label-plus-text BoolQ forms, thinking-block removal, truncation metadata, and anti-overmatching. Anti-overmatching examples include ordinary mentions of "yes" or "no" inside a rationale that do not declare a final answer. Parser identity is stored as a versioned fingerprint in every finalized metric artifact.

During development, the label-plus-text forms exposed an overly narrow BoolQ rule. The correction was applied to the shared extractor, after which every affected cached raw generation was re-scored. Training and generation outputs were not selectively repeated. Aggregate collectors reject missing-value sentinels, inconsistent sample counts, incomplete trial grids, and any table value that differs from the locked CSV at the reported precision.

H Artifact Provenance and Reproduction

The supplementary archive contains two complementary records. The compact paper source contains the LaTeX, vector figures, locked CSVs, plotting code, table-generation code, citation audit, and numerical validation report. The reproducibility supplement contains the experiment source snapshot, sanitized high-level configurations, resolved per-role configurations, aggregate server metrics, component manifests, tests, and a provenance manifest.

Item	Released record
Model identity	Exact public checkpoint identifiers and model families
Data identity	Dataset, split, expected count, and sampling rule
Training identity	Per-role YAML, optimizer, LoRA and token limits
Prompt identity	Base prompt version and exact role prefixes
Evaluation identity	Vote, parser hash, round and trial counts
Result identity	Locked CSVs and server metrics per cell
Code identity	Code snapshot, tests, dependencies and entry points
Artifact inventory	File list, source locations, sizes, and redaction status

Table 13: Released reproducibility identities. The resolved configurations take precedence over high-level configuration templates.

H.1 Numerical Audit

The one-command verifier reads aggregate experiment artifacts rather than LaTeX. It checks 25 main-comparison rows, 150 component cells, four full-precision contrasts, 30 round-wise macro entries, 25 decision-diagnostic rows, the absence of missing-value sentinels, and the expected file inventory. Separately, the paper validator reconstructs table entries from the locked CSVs, checks macro averages and positive-cell counts, verifies that every citation key is present in the manuscript, bibliography, source registry, and online audit, and confirms that the official style files are present.

The revision audit is rebuilt from the 25 full-system decision files. For each cell and round, a correction changes an R0 error to a correct final vote; a regression changes an R0 success to an error. Both counts use all examples as the denominator. Their difference equals the accuracy change from R0. The plotting script releases 100 cell–round records and checks this identity before taking equal-weight averages across cells or backbones.

H.2 Citation Audit

Every bibliography entry is represented in a machine-readable registry with a primary paper page, a second independent metadata source, and a BibTeX source. The final audit re-fetches all three, records HTTP status and title matching, and requires the cited-key, BibTeX-key, registry-key, and audit-key sets to be identical across all four records. Each citation has a verified registry entry.

H.3 Full Rerun Path

A full rerun begins by installing the declared project dependencies, placing the official benchmark data under the configured root, and obtaining the five public checkpoints. The multi-pair pipeline then runs role SFT, materializes resolved role configurations, executes Critic and Actor preference stages, and evaluates all configured datasets. Baselines use their separate locked entry point; component variants use the policy-lattice runner. The final collector aligns artifacts by model, method, dataset, trial, and parser fingerprint before emitting the CSVs consumed by this paper.

The package includes 125 compressed per-example decision files covering all completed configurations for the 25 model–dataset cells. They retain sample identities, gold labels, role answers, rounds, and final votes. The standalone decision verifier recomputes 2,427,600 round decisions from 578,000 example records and checks 4,950 corresponding aggregate quantities. Model weights, adapters, benchmark question text, and full generated rationales remain outside the compact archive; their source configurations and artifact identities are retained for a full rerun.

H.4 Reproduction Acceptance Criteria

We distinguish a *paper audit* from a *full rerun*. A paper audit requires no accelerator and succeeds only if the expected source and template files are present, every expected CSV row is present, all reported aggregates recompute at full precision, and the rounded TeX cells match the regenerated tables. A full rerun additionally requires the checkpoint revisions, benchmark identities, resolved configurations, role seeds, and parser fingerprint recorded in the provenance manifest. A different model revision or evaluation split is a new experiment, not a reproduction of a locked cell.

The collector accepts a rerun only after five checks. First, every model–method pair must contain the declared dataset grid and expected sample count. Second, trial identifiers must align across roles before voting. Third, every raw answer must carry a valid parser status, including explicit invalid parses. Fourth, macro values must be reconstructed from dataset-level numerators and denominators rather than copied from logs. Fifth, each result must link to its resolved configuration and source manifest. Any failed check aborts collection instead of emitting a partial table.

Stochastic generation need not reproduce individual strings bit-for-bit across hardware stacks. The acceptance target is instead protocol identity plus the declared aggregate output of the released run. New reruns should report their own seeds, library versions, artifact identifiers, and uncertainty alongside the locked reference values. These records distinguish a source audit from an independent empirical replication and make every manuscript number traceable to an immutable artifact path in the released package.