GLOBAL-TO-LOCAL DECISION PROJECTION FOR COLLABORATIVE REASONING AMONG HETEROGENEOUS LARGE LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Complementarity among heterogeneous large language models (LLMs) underpins collaborative reasoning, yet reaching a reliable collective decision remains challenging when initial answers disagree. Majority support may favor an incorrect answer, while dispersed support may leave the choice unclear. We observe that models with incorrect initial answers may shift their support to the correct answer when directly comparing disputed candidates, but these judgments may conflict or provide clear support for only some candidate pairs. To this end, we introduce *PairRecon*, a *global-to-local decision projection* framework for collaborative reasoning among heterogeneous LLMs. PairRecon organizes collaboration using the candidates and the distribution of model support revealed by initial disagreement. Guided by this structure, it projects global candidate selection into local pairwise comparisons, enabling models to provide additional judgments on specific disputes. For each candidate pair, judgments that agree across both presentation orders are retained as directed relations supporting one candidate and opposing the other. These retained relations, together with initial support, enable candidate reconstruction despite unresolved local disagreements. Depending on the disagreement pattern, candidate admission or staged arbitration further constrains answer updates to reduce harmful revisions. PairRecon requires no training or parameter updates and allocates additional inference only to initial disagreement cases. Across nine benchmarks, PairRecon improves collective decision accuracy. In the primary evaluation, it improves overall and disagreement-case accuracy over direct three-model aggregation by 1.50% and 6.89%, respectively. It also achieves the greatest net gain and the lowest inference cost among evaluated methods requiring additional inference. Accuracy gains extend to smaller models and open-ended tasks.

1 INTRODUCTION

Heterogeneous large language models (LLMs) differ in knowledge coverage, reasoning capabilities, and error patterns, providing a basis for complementarity in collaborative reasoning (Jiang et al., 2023; Huang et al., 2024). Combining independent judgments from different models can yield more accurate collective decisions than individual models can achieve alone (Chen et al., 2024b; Du et al., 2023; Wang et al., 2025; Chen et al., 2024a; Feng et al., 2026). However, model complementarity does not automatically translate into collaborative gains. Majority support may concentrate on an incorrect answer, while dispersed support may provide no clear basis for choosing among candidates. Obtaining useful evidence from model disagreement and using it to support more accurate collective choices is therefore a central challenge in heterogeneous model collaboration.

Existing approaches aggregate model outputs through repeated sampling, voting, or confidence aggregation (Wang et al., 2022; Li et al., 2024; Choi et al., 2026; Chen et al., 2024a). Multi-round interaction and debate allow models to exchange answers and reasoning, evaluate one another, and revise their judgments (Du et al., 2023; Liang et al., 2024; Chen et al., 2024a; Tian et al., 2026). Other approaches use coordinators, graph structures, or learned strategies to organize agent roles and information flow (Chen et al., 2024b; Zhuge et al., 2024; Wang et al., 2025). These studies suggest that collective decision-making can draw on both existing judgments and additional information elicited through interaction or comparison when initial model judgments diverge.

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.

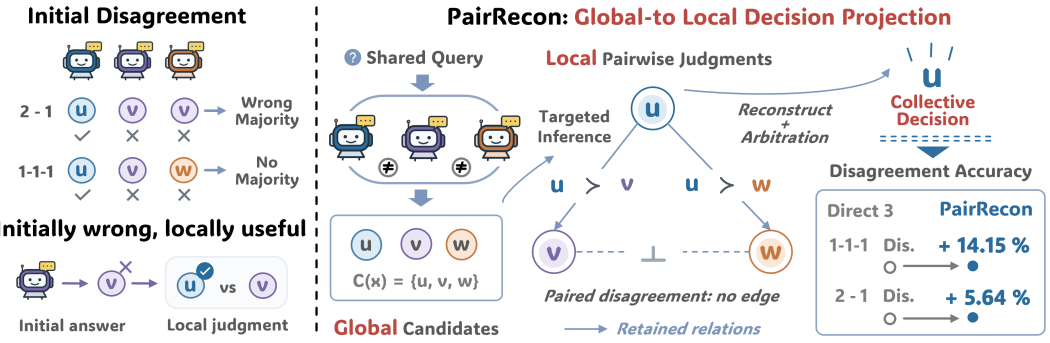


Figure 1: **Global-to-local decision projection in PairRecon.** Initial support can be misleading or inconclusive, yet initially incorrect models can provide useful local judgments about the competing candidates. PairRecon uses initial candidates and support to organize counterbalanced comparisons, retaining agreed relations and leaving conflicts unresolved. In the illustrated 1-1-1 case, these partial relations and initial support guide reconstruction followed by staged arbitration.

An initial answer reflects a model's decision on the task as a whole, but may not fully capture the value of its judgments in subsequent collaboration. We observe that a model that initially answers incorrectly may still favor the correct candidate in local comparisons focused on specific pairs of candidates. This observation suggests that a model can contribute to collective error correction through useful judgments about competing candidates, even when its independent answer is incorrect. However, these local judgments are distributed across different models and candidate pairs, and may conflict with one another or yield clear conclusions for only some comparisons.

Building on these observations, we introduce *PairRecon*, a *global-to-local decision projection* framework for collaborative reasoning among heterogeneous LLMs. Guided by the initial distribution of model support, PairRecon projects the global choice among candidates into local pairwise comparisons and extracts the direction of support between candidates from paired judgments, as illustrated in Figure 1. For each candidate pair, a directed relation is retained only when the judgments under both presentation orders select the same candidate, recording support for that candidate and opposition to the other; inconsistent observations remain unresolved and contribute neither support nor opposition. PairRecon combines the resulting candidate graph with initial support to reconstruct a candidate decision, while candidate admission and, when required, staged arbitration determine whether to retain or revise the current answer. This allows retained judgments to support a global choice even when some comparisons remain unresolved. PairRecon operates at inference time without training or parameter updates and allocates additional inference only to instances with initial disagreement.

We evaluate PairRecon across tasks, model scales, and answer formats. PairRecon improves decision accuracy and, in our primary evaluation, achieves a greater net gain in correct answers at lower inference cost than other methods requiring additional inference. Aggregate gains with smaller model ensembles and on open-ended tasks support its applicability. Mechanism analysis shows that, among analyzed cases containing the correct candidate, initially incorrect models can still produce retained relations supporting it, providing empirical evidence that local judgments can elicit useful model contributions not reflected in initial answers. Our main contributions are summarized as follows:

- We introduce PairRecon, a global-to-local decision projection framework for collective decision-making among heterogeneous LLMs. PairRecon projects global candidate selection into targeted pairwise judgments using the disputed candidates and their initial support. These judgments elicit useful model contributions not expressed in the initial answers.
- We develop a decision reconstruction mechanism that distinguishes unresolved comparisons from retained support and opposition relations. It combines partial judgments from different models with initial support to reconstruct a collective choice and determine whether to revise the current answer, without requiring every candidate comparison to be resolved.
- We evaluate PairRecon across reasoning tasks, model scales, and answer formats, showing improved collective decision accuracy and, in the primary evaluation, greater net gains at lower inference cost than other methods requiring additional inference. Mechanism analysis and component ablations examine how local judgments contribute to collective revision.

2 Related Work

2.1 Collaborative Reasoning with Language Models

Collaborative reasoning combines the perspectives of language-model agents through role specialization, communication, and mutual revision. CAMEL and AgentVerse explore cooperation through role-based interaction and team coordination (Li et al., 2023; Chen et al., 2024b). For reasoning tasks, multi-agent debate enables agents to exchange proposed solutions and arguments, supporting iterative refinement through different perspectives (Du et al., 2023; Liang et al., 2024).

Further research investigates how to structure interactions and determine which information agents should use during the collaborative reasoning process. GPTSwarm optimizes agent workflows as computational graphs (Zhuge et al., 2024), while Mixture-of-Agents organizes generation through successive layers of model outputs (Wang et al., 2025). To improve interaction reliability and efficiency, MAD-M^2 masks erroneous memories before subsequent debate rounds (Tian et al., 2026), and DOWN uses initial response confidence to determine whether debate is needed (Eo et al., 2025).

2.2 Collective Decision Making at Inference Time

Ensembling methods combine model predictions at different levels to integrate complementary information from multiple models. Self-Consistency selects an answer by aggregating sampled reasoning paths (Wang et al., 2022), and Agent Forest scales sampling and voting across agents (Li et al., 2024). For heterogeneous ensembles, LLM-Blender learns to rank pairs of responses and fuse the highest-ranked candidates (Jiang et al., 2023), whereas DeePEn aligns and combines token distributions during decoding (Huang et al., 2024). More adaptive approaches include DMoA, which uses output consistency to guide ensemble refinement (Abdulaal et al., 2025), and SpecEM, which combines drafting and verification with online updates to model voting weights (Lv et al., 2026).

Decision protocols also shape collective reasoning by determining how individual judgments are consolidated. ReConcile combines discussion among heterogeneous models with confidence-weighted voting (Chen et al., 2024a), while GEDI incorporates electoral rules based on ordinal preferences (Zhao et al., 2024). Recent analyses compare voting and consensus protocols and examine the respective contributions of voting and debate (Kaesberg et al., 2025; Choi et al., 2026). PairRecon instead frames collective decision making as global-to-local decision projection, using initial disagreement to organize targeted local pairwise judgments and reconstruct a collective choice from retained relations and initial support even when some comparisons remain unresolved.

3 PairRecon: Global-to-Local Decision Projection

In this section, we first formulate the collective decision problem for heterogeneous language models at inference time. We then introduce the local decision projection and global reconstruction mechanisms of PairRecon, whose overall framework is illustrated in Figure 2.

3.1 Problem Formulation

Given a query $x \in \mathcal{X}$ with answer space $\mathcal{Y}(x)$, consider M heterogeneous language model agents $\mathcal{A} = \{A_i\}_{i=1}^{M}$. Each agent independently produces a canonicalized answer $a_i \in \mathcal{Y}(x)$. Together, these answers represent the agents' initial judgments and induce the candidate set:

$$\mathcal{C}(x) = \{a_i \mid i = 1, \ldots, M\}, \qquad K = |\mathcal{C}(x)|.$$

Each candidate $c \in \mathcal{C}(x)$ receives an initial support count $n(c) = \sum_{i=1}^{M} \mathbf{1}[a_i = c]$, where $\mathbf{1}[\cdot]$ is the indicator function. These counts characterize the initial agreement pattern among the agents. When $K = 1$, the consensus answer is returned directly; when $K > 1$, PairRecon coordinates the remaining disagreement. For disagreement cases, let $S(a \mid x)$ denote the initial aggregate support for an answer a, and let y_0 denote the initial aggregation decision. For multiple-choice inputs, each agent provides normalized support scores $p_i(a \mid x)$ over the full option set $\mathcal{Y}(x)$, yielding:

$$S(a \mid x) = \sum_{i=1}^{M} \log p_i(a \mid x), \qquad y_0 = \arg\max_{a \in \mathcal{Y}(x)} S(a \mid x).$$

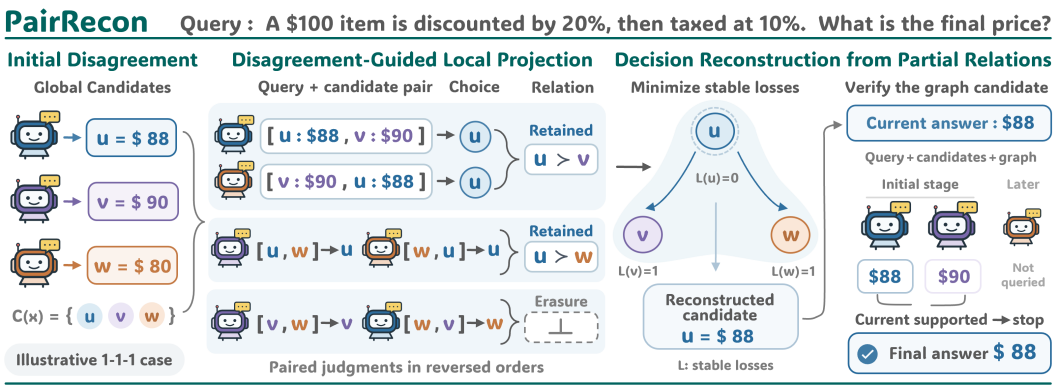


Figure 2: **An illustrative 1-1-1 example of PairRecon.** Paired judgments under reversed candidate orders yield a directed relation when they agree; otherwise, no relation is retained. PairRecon reconstructs a candidate from the resulting partial relation graph and applies staged arbitration, which in this example retains the current decision upon initial-stage support and stops further evaluation.

For open-ended inputs, aggregation operates on the canonicalized candidate set, with $S(c \mid x) = n(c)$ and y_0 selected by maximum support. Any remaining ties are resolved by a fixed deterministic rule.

Starting from y_0, our goal is to obtain a final collective decision $\hat{y}$ through further coordination among the agents, with model parameters held fixed. For multiple-choice tasks, $\hat{y} \in \mathcal{Y}(x)$; for open-ended tasks, $\hat{y} \in \mathcal{C}(x)$. Local projection and candidate reconstruction operate on $\mathcal{C}(x)$, while subsequent candidate admission or staged arbitration determines the final decision.

3.2 Disagreement-Guided Local Projection

When the agents' answers differ, they induce $K > 1$ competing candidates and a pattern of model support. PairRecon uses this disagreement structure to project the global choice over $\mathcal{C}(x)$ into pairwise judgments, allowing models to express relative support within specific candidate pairs.

Local pairwise judgments. The candidate set induces the space of unordered candidate pairs:

$$\mathcal{P}(x) = \big\{\{u, v\} \mid u, v \in \mathcal{C}(x),\ u \neq v\big\}, \qquad |\mathcal{P}(x)| = \binom{K}{2}.$$

Although the candidate set may contain more than two answers, each local query considers only one pair. For a queried pair $\{u, v\} \in \mathcal{P}(x)$, an assigned agent receives the query and two candidates, together with candidate-specific supporting work when available, and selects the more likely candidate. Candidate provenance and initial support counts are not disclosed. We denote agent A_i's judgment by $\Pi_i(x; [u, v]) \in \{u, v\}$, where $[u, v]$ specifies the presentation order. Each query therefore elicits a binary local judgment while preserving the full candidate set for subsequent reconstruction.

Paired relational observations. A single local judgment does not yet define a retained relation. PairRecon pairs two judgments of the same candidate pair under reversed presentation orders and maps both outputs back to their canonical candidate identities. Evaluator assignment is guided by the initial model–candidate support structure and may draw on agents that supported either candidate as well as, when available, agents that supported another candidate. The two judgments in a paired observation may therefore come from the same or different agents; detailed assignments are provided in Appendix A.4. For the q-th paired observation, let $z_{uv}^{(q,+)}$ and $z_{uv}^{(q,-)}$ denote the canonicalized outputs under $[u, v]$ and $[v, u]$, respectively. The resulting relational observation is:

$$r_{uv}^{(q)} = \begin{cases} u, & z_{uv}^{(q,+)} = z_{uv}^{(q,-)} = u, \\ v, & z_{uv}^{(q,+)} = z_{uv}^{(q,-)} = v, \\ \perp, & z_{uv}^{(q,+)} \neq z_{uv}^{(q,-)}. \end{cases}$$

Here, $r_{uv}^{(q)} = u$ represents the directed relation $u \to v$, recording support for u and opposition to v; $r_{uv}^{(q)} = v$ represents the reverse direction. When the two judgments disagree, $\perp$ leaves the relation

unresolved and contributes neither support nor opposition. The retained observations from evaluated candidate pairs form the partial relation graph used for candidate reconstruction.

3.3 Decision Reconstruction from Partial Relations

PairRecon reconstructs a candidate decision from the retained local relations and determines the final answer through candidate admission or staged arbitration. This allows partial relations to inform collective decisions without requiring every candidate comparison to be resolved.

Candidate reconstruction. Let $\mathcal{C} = \mathcal{C}(x)$. PairRecon represents each retained relation as an edge from the supported candidate to the opposed candidate, while an unresolved observation contributes no edge. Multiple retained relations for the same pair are preserved in a directed multigraph $G(x) = (\mathcal{C}, E(x))$. For each candidate $c \in \mathcal{C}$, its support and opposition counts are:

$$W(c) = \sum_{v \in \mathcal{C} \setminus \{c\}} \sum_{q} \mathbf{1}[r_{cv}^{(q)} = c], \qquad L(c) = \sum_{v \in \mathcal{C} \setminus \{c\}} \sum_{q} \mathbf{1}[r_{cv}^{(q)} = v].$$

PairRecon reconstructs a candidate by minimizing its conflicts with the retained judgments. Each opposing relation records a local choice against that candidate, while unresolved comparisons provide no directional evidence. This gives precedence to fewer opposing relations over additional supporting ones. The initial aggregate score resolves ties among candidates with minimum opposition:

$$\mathcal{C}_{\min}(x) = \arg\min_{c \in \mathcal{C}} L(c), \qquad \widetilde{y} = \arg\max_{c \in \mathcal{C}_{\min}(x)} S(c \mid x).$$

Any remaining tie is resolved by a fixed deterministic rule before the revision stage.

Conditional revision. Candidate admission and staged arbitration govern decision updates according to the disagreement structure. Let τ denote the required support threshold. For $K = 2$, a reconstructed candidate $\widetilde{y} \neq y_0$ replaces y_0 if either $W(\widetilde{y}) \geq \tau$ and $L(\widetilde{y}) = 0$, or $\widetilde{y}$ is the initial majority candidate. For multiple-choice inputs, an additional admission condition permits replacement when $W(\widetilde{y}) = 1$, $L(\widetilde{y}) = 0$, and the initial score gap satisfies:

$$S(y_0 \mid x) - S(\widetilde{y} \mid x) \leq \delta,$$

where δ is the score-gap threshold. Otherwise, PairRecon retains y_0. For $K > 2$, PairRecon instead initializes the current decision as $y = \widetilde{y}$ and applies staged arbitration to determine whether an alternative should replace it. If any judgment in the initial stage supports y, PairRecon retains it and stops; otherwise, subsequent judgments are collected. For the collected arbitration judgments $\mathcal{O}$, $\mathrm{supp}_{\mathcal{O}}(c)$ counts those selecting c, and $c^\star$ is the most-supported alternative. The final decision is:

$$\widehat{y} = \begin{cases} c^\star, & \mathrm{supp}_{\mathcal{O}}(c^\star) \geq \tau \quad \text{and} \quad \mathrm{supp}_{\mathcal{O}}(y) = 0, \\ y, & \text{otherwise.} \end{cases}$$

Arbitration may select any original option for multiple-choice inputs but is restricted to the candidate set for open-ended inputs. The staged order retains a supported provisional decision and permits replacement only when sufficient support favors an alternative and none favors the current choice among all the collected judgments. Together, reconstruction and arbitration allow partial relations to guide the final choice without requiring a complete ranking of candidates. Further implementation details and end-to-end examples are provided in Appendices A.4 and C.

4 Experiment

4.1 Experimental Setup

Benchmarks. We evaluate PairRecon on nine benchmarks comprising 12051 examples across six domains: commonsense, social reasoning, science, general knowledge, mathematics, and program reasoning. They span multiple-choice and open-ended formats. The multiple-choice evaluation includes 9732 examples from ARC-Challenge (ARC-C) (Clark et al., 2018), CommonsenseQA (CSQA) (Talmor et al., 2019), HellaSwag (HS) (Zellers et al., 2019), MathQA (Amini et al., 2019), MMLU-Pro (MMLU-P) (Wang et al., 2024), and SocialIQA (SIQA) (Sap et al., 2019). The open-ended evaluation comprises 2319 examples from GSM8K (Cobbe et al., 2021), MATH-500 (Lightman et al., 2024), and MBPP (Austin et al., 2021). We use complete official evaluation splits or fixed, deterministically sampled subsets for consistent method comparisons.

Table 1: Accuracy and inference efficiency on six multiple-choice benchmarks. Best Single denotes Llama 3.3 70B FP8, the individual model with the highest overall accuracy. Direct3 aggregates the three initial model predictions. Direct9 obtains two additional responses per model only on disagreement cases and aggregates all nine responses by voting. Bold and underlined values indicate the best and second-best results, respectively. Efficiency rankings exclude Best Single and Direct3.

Methods	Benchmark Accuracy (%) ↑						Aggregate (%) ↑		Efficiency ↓	
	ARC-C	CSQA	HS	MathQA	MMLU-P	SIQA	Overall	Macro	Tokens	Lat. (s)
Qwen3-32B + Gemma 3 27B + Llama 3.3 70B FP8										
Best Single	95.99	86.00	86.60	87.54	70.14	80.81	84.41	84.51	895	9.71
Direct3	96.25	85.91	87.00	88.98	72.14	<u>81.06</u>	85.25	85.22	2424	18.36
Direct9	96.08	85.34	87.40	**90.95**	75.93	80.25	<u>86.19</u>	<u>85.99</u>	<u>3862</u>	30.38
Agent Forest	96.33	<u>86.08</u>	<u>87.80</u>	89.11	71.93	80.96	85.36	85.37	3942	37.46
MAD	96.25	85.75	86.00	<u>90.75</u>	76.14	79.27	85.89	85.69	6089	<u>29.14</u>
ReConcile	<u>96.50</u>	85.26	86.60	90.22	**76.57**	80.66	86.10	85.97	6278	37.69
MAD-M^2	96.42	**86.24**	86.30	90.42	75.64	79.89	85.95	85.82	4948	30.43
PairRecon	**96.59**	85.42	**89.50**	90.72	<u>76.50</u>	**81.58**	**86.76**	**86.72**	**3822**	**28.15**

Table 2: Performance on the 2118 examples with initial agent disagreement. Relative to Direct3, Δ_{dis} denotes the absolute accuracy gain, while Rescues and Damages count incorrect-to-correct and correct-to-incorrect revisions, respectively. Net is their difference, and Benefit is $\text{Rescues}/(\text{Rescues}+\text{Damages})$, expressed as a percentage. Decision-change rankings exclude Direct3.

Methods	Disagreement Performance (%) ↑				Decision Changes			
	All Acc. $N = 2{,}118$	2–1 Acc. $N = 1{,}807$	1–1–1 Acc. $N = 311$	Δ_{dis}	Rescues ↑	Damages ↓	Net ↑	Benefit ↑
Direct3	54.39	58.66	29.58	–	–	–	–	–
Direct9	<u>58.69</u>	<u>61.82</u>	40.51	<u>+4.30</u>	363	272	<u>+91</u>	<u>57.17</u>
Agent Forest	54.86	58.77	32.15	+0.47	178	<u>168</u>	+10	51.45
MAD	57.32	60.60	38.26	+2.93	<u>412</u>	350	+62	54.07
ReConcile	58.26	61.15	<u>41.48</u>	+3.87	**416**	334	+82	55.47
MAD-M^2	57.60	60.49	40.84	+3.21	373	305	+68	55.01
PairRecon	**61.28**	**64.31**	**43.73**	**+6.89**	285	**139**	**+146**	**67.22**

Models and baselines. Our main experiments use three heterogeneous language models: Qwen3-32B (Yang et al., 2025), Gemma 3 27B (Team et al., 2025), and Llama 3.3 70B FP8 (Grattafiori et al., 2024). For multiple-choice tasks, we compare PairRecon with Best Single, Direct3, Direct9, Agent Forest (Li et al., 2024), MAD (Du et al., 2023), ReConcile (Chen et al., 2024a), and MAD-M^2 (Tian et al., 2026). Best Single denotes Llama 3.3 70B FP8, which achieves the highest overall accuracy among the models on the multiple-choice suite. Direct3 uses the initial aggregation rule in Section 3.1. Direct9 obtains two additional responses per model on disagreement cases and aggregates all nine answers by voting. These methods use the same initial disagreement split, evaluation instances, and scoring protocol. Consensus answers are returned directly, with additional inference restricted to disagreement cases. Results with three smaller heterogeneous models are reported in Appendix B.1.

Evaluation metrics. We report per-benchmark accuracy, overall accuracy across examples, and macro-averaged accuracy across benchmarks within each evaluation suite. Open-ended outputs are assessed through numerical matching, mathematical equivalence, or program execution. Disagreement subsets are defined by the three initial answers, with results reported for all disagreement cases and separately for the 2-1 and 1-1-1 patterns. Relative to Direct3, rescues denote incorrect-to-correct changes, damages denote correct-to-incorrect changes, and net gain is their difference. Efficiency is measured by logical calls, total input and output tokens, and serial-equivalent latency, defined as the summed logical-call duration per example. These costs include both initial and additional inference.

Implementation details. Open-ended candidates are constructed using deterministic, task-specific normalization without reference answers. We use fixed prompts and decoding configurations, with a support threshold of $\tau = 2$ for decision updates and a score-gap threshold of $\delta = 10$ for score-

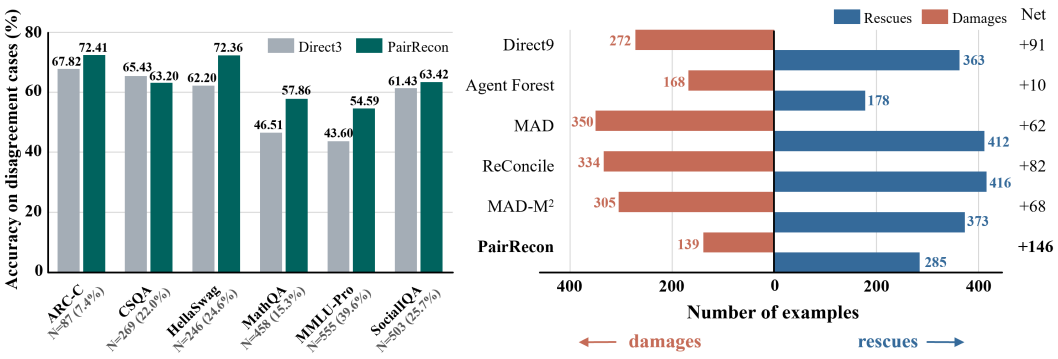


Figure 3: Disagreement resolution across benchmarks and methods. (a) Direct3 and PairRecon accuracy on disagreement cases, with disagreement counts and rates annotated. (b) Rescues (incorrect-to-correct changes) and damages (correct-to-incorrect changes) relative to Direct3, pooled across 2118 disagreement cases from six benchmarks. Net denotes rescues minus damages.

based admission on multiple-choice tasks. Experimental settings, normalization rules, and baseline configurations are provided in Appendix A; prompt templates appear in Appendix D.

4.2 Performance and Efficiency

Table 1 compares methods on six multiple-choice benchmarks under a shared disagreement split and consistent scoring. Overall covers all 9732 examples, while Macro is the unweighted mean of benchmark accuracies. PairRecon achieves the highest overall and macro-averaged accuracies of 86.76% and 86.72%, with an overall gain of 1.50% over Direct3. It exceeds Direct9, the strongest competing baseline, by 0.57% overall and 0.73% macro. PairRecon also leads on ARC-Challenge, HellaSwag, and SocialIQA, with a 1.70% margin over the second-best method on HellaSwag.

PairRecon averages 3822 tokens and 28.15 seconds of serial-equivalent latency per example, both the lowest among methods requiring additional inference. Compared with Direct9, it uses slightly fewer tokens and reduces serial-equivalent latency by 7.3%. Token consumption is 22.8%–39.1% lower than MAD, ReConcile, and MAD-M^2. Relation-level collaboration thus achieves higher aggregate accuracy at lower inference cost than the evaluated sampling and debate baselines.

4.3 Disagreement Resolution

Table 2 compares the methods on examples where the initial predictions are not unanimous. Among the 2118 disagreement cases, PairRecon achieves the highest accuracy of 61.28%, improving over Direct3 and the strongest competing baseline, Direct9, by 6.89% and 2.60%, respectively. PairRecon also achieves the highest accuracies of 64.31% and 43.73% on the 2-1 and 1-1-1 subsets, corresponding to gains of 5.64% and 14.15% over Direct3. These results demonstrate improved collective decisions both when an initial majority exists and when all three agents disagree.

Figure 3(a) shows disagreement rates among the three models, ranging from 7.42% on ARC-Challenge to 39.64% on MMLU-Pro. PairRecon improves disagreement accuracy over Direct3 on five of six benchmarks, with gains exceeding 10% on HellaSwag, MathQA, and MMLU-Pro. The 2.23% decline on CommonsenseQA indicates that these benefits vary across domains.

Table 2 and Figure 3(b) show that PairRecon corrects 285 initial errors while introducing 139 new errors relative to Direct3. These changes yield the highest net gain of 146 examples and the highest Benefit ratio of 67.22%, combining error correction with fewer harmful revisions.

4.4 Ablation and Mechanism Analysis

We examine bidirectional projection, adaptive candidate admission, reliability gating, and relational arbitration through ablations and local judgment analysis. Table 3 and Figure 4 summarize the results. Appendices B.2 and B.3 provide benchmark-level ablations and model-specific analyses, respectively.

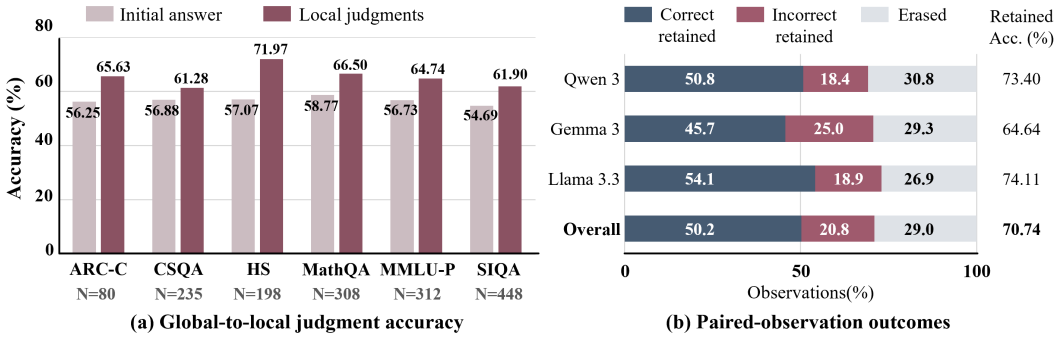


Figure 4: Mechanism analysis on 1581 two-candidate disagreement cases with a correct candidate and complete paired evaluations for all three models. (a) Initial-answer and local-judgment accuracy pooled across models within each benchmark; local accuracy averages both candidate orders. N denotes the number of eligible cases. (b) Fractions of correct retained, incorrect retained, and erased observations; the right column reports accuracy among retained relations.

Table 3: Ablation results on six multiple-choice benchmarks. R/D/Net denotes rescues, damages, and their difference versus Direct3. Δ_{Full} denotes the absolute overall accuracy change versus PairRecon, computed before rounding. Calls/task reports the average logical calls per example. Variants reuse recorded outputs without additional model calls. †Single-order variants modify only the 2-1 route; 1-1-1 outputs remain unchanged because arbitration inputs contain the original bidirectional graph.

Variant	Accuracy (%) ↑				R/D/Net	Δ_{Full}	Calls/task
	Overall	Disagr.	2-1	1-1-1			
Direct3	85.25	54.39	58.66	29.58	0/0/+0	−1.50	3.000
w/o Bidirectional (forward)†	86.33	59.35	62.04	43.73	267/162/+105	−0.42	3.823
w/o Bidirectional (reverse)†	86.63	60.72	63.64	43.73	297/163/+134	−0.12	3.823
w/o Adaptive Admission	86.40	59.63	62.37	43.73	210/99/+111	−0.36	4.380
w/o Reliability Gate	86.68	60.95	63.92	43.73	324/185/+139	−0.07	4.380
w/o 1-1-1 Arbitration	86.66	60.86	64.31	40.84	268/131/+137	−0.09	4.306
PairRecon	**86.76**	**61.28**	**64.31**	**43.73**	**285/139/+146**	–	4.380

Table 4: Reconstruction before admission or arbitration using the same candidate set. Retained is the mean number of retained relations, excluding nine incomplete cases. Changed (%) and C/H report selection changes and corrections/harms relative to Score-only. Δ gives the accuracy gain with 95% paired bootstrap CIs. With erasures denotes cases with unresolved comparisons.

Subset	N	Retained	Accuracy (%) ↑		Changed (%)	Δ	95% CI	C/H
			Score-only	Relational				
2-1	1807	2.10	58.66	63.92	27.67	+5.26	[3.04, 7.47]	256/161
With erasures	1141	1.58	54.34	60.21	29.01	+5.87	[3.07, 8.68]	170/103
1-1-1	311	1.70	29.90	40.84	35.69	+10.93	[6.11, 15.76]	50/16
With erasures	252	1.41	30.95	41.67	31.75	+10.71	[5.56, 15.87]	39/12

Local decision signals. Figure 4(a) compares each model's initial answers with its local judgments on matched examples. Across the 1581 eligible cases, local judgments improve accuracy on all six benchmarks, increasing pooled model-level accuracy from 56.59% to 64.72%. Moreover, 703 of the 2059 initially incorrect model-level answers yield correct retained relations, giving an initial-error recovery rate of 34.14%. These results show that targeted candidate comparisons can recover useful judgments beyond the models' initial answers, providing additional signals for reconstruction.

Order-consistent observations. Figure 4(b) shows that 70.97% of the 4743 paired observations are retained, with an accuracy of 70.74% among retained relations. The remaining 29.03% are erased because the paired candidate choices disagree. In Table 3, retaining only forward or reverse judgments reduces overall accuracy from 86.76% to 86.33% and 86.63%, respectively. On the

Table 5: Results on three open-ended benchmarks. Δ denotes the absolute accuracy gain over Direct3, computed from unrounded scores. Oracle@3 is the percentage of examples with at least one correct answer among the three initial outputs. Mixed N counts candidate sets containing correct and incorrect answers; Mixed Acc. reports PairRecon's selection accuracy on these sets as a percentage. Candidate diagnostics are computed offline using reference outcomes and not used for inference.

Dataset	N	Metric	End-to-End Accuracy (%) ↑			Candidate Diagnostics		
			Direct3	PairRecon	Δ	Oracle@3	Mixed N	Mixed Acc.
GSM8K	1319	Numeric EM	96.36	**96.59**	**+0.23**	97.80	73	78.08
MATH-500	500	Math-Verify Acc.	83.20	**85.80**	**+2.60**	91.40	160	82.50
MBPP	500	Pass@1	73.80	**76.20**	**+2.40**	84.40	113	63.72
Overall	2319	Micro Accuracy	88.66	**89.87**	**+1.21**	93.53	346	75.43
Macro Avg.	–	Dataset Average	84.45	**86.20**	**+1.74**	–	–	–

affected 2-1 subset, accuracy decreases from 64.31% to 62.04% and 63.64%. The difference between the single-order variants reveals dependence on the retained presentation order, while the complete method outperforms both, supporting the use of paired judgments for relation construction.

Candidate admission and reliability. Reverting to strict candidate admission lowers overall accuracy to 86.40%, reduces rescues from 285 to 210, and decreases the net gain from 146 to 111. Adaptive admission enables useful revisions without changing relational observations. Removing the reliability gate directly adopts the reconstructed answer, producing 39 additional rescues but 46 additional damages and reducing the net gain to 139. These components play complementary roles: candidate admission enables beneficial revisions, while reliability gating limits harmful changes.

Relational reconstruction and arbitration. We compare relational reconstruction with score-only selection that maximizes $S(c \mid x)$ over the same $\mathcal{C}(x)$, before admission or arbitration. Table 4 shows that incorporating retained relations increases reconstruction accuracy from 58.66% to 63.92% on 2-1 cases and from 29.90% to 40.84% on 1-1-1 cases. Gains persist in subsets with unresolved comparisons, which retain only 1.58 and 1.41 relations on average, respectively. Subsequent arbitration further raises 1-1-1 accuracy to 43.73%, yielding a net gain of nine correct answers.

4.5 Generalization to Open-Ended Tasks

We evaluate PairRecon on GSM8K, MATH-500, and MBPP using normalized numerical answers, mathematical expressions, and programs. Table 5 shows improvements over Direct3 on all three benchmarks, increasing overall accuracy from 88.66% to 89.87% and macro-averaged accuracy by 1.74%. Gains reach 2.60% on MATH-500 and 2.40% on MBPP, supporting the applicability of decision projection beyond predefined choices.

Candidate diagnostics distinguish coverage from selection. The overall Oracle@3 of 93.53% provides an accuracy upper bound when selecting from initial candidates. Among the 346 mixed candidate sets containing both correct and incorrect answers, PairRecon achieves 75.43% selection accuracy. These sets isolate cases where selection can affect correctness. The lower coverage and mixed-set accuracy on MBPP highlight opportunities to improve both candidate generation and selection.

5 Conclusion

We present PairRecon, a global-to-local decision projection framework for collaborative reasoning among heterogeneous LLMs. PairRecon treats initial disagreement as a structure for subsequent collaboration, using the disputed candidates and their support distribution to elicit targeted pairwise judgments and reconstruct a collective choice from partial relations and initial support. Across nine benchmarks, PairRecon improves collective decision accuracy across model scales and answer formats. In the primary evaluation, it achieves the greatest net gain and lowest inference cost among evaluated methods requiring additional inference. These results show that judgments beyond initial answers can support collective revision without resolving every candidate comparison.

AI USE STATEMENT

We used generative AI tools to assist with method development and implementation, experimental planning and analysis, and manuscript preparation, including literature search, translation, language editing, and figure preparation. The authors have reviewed and verified all AI-assisted materials and take full responsibility for the final paper, code, and reported results.

REPRODUCIBILITY STATEMENT

Appendix A documents dataset selection and evaluation, candidate normalization, model configurations, inference settings, and implementation details for PairRecon and the baselines. These details include random seeds, generation limits, and token and latency accounting. Appendix D provides the prompts used for independent generation, pairwise evaluation, arbitration, and baseline interactions. Appendix B reports benchmark-level ablations and results with smaller models, while Appendix C traces the intermediate judgments and decision updates in two recorded cases.

REFERENCES

Ahmed Abdulaal, Chen Jin, Nina Montaña-Brown, Aryo Pradipta Gema, Daniel Castro, Daniel Alexander, Philip Teare, Tom Diethe, Dino Oglic, and Amrutha Saseendran. Balancing act: Diversity and consistency in large language model ensembles. In *International Conference on Learning Representations*, volume 2025, pp. 66155–66187, 2025.

Aida Amini, Saadia Gabriel, Shanchuan Lin, Rik Koncel-Kedziorski, Yejin Choi, and Hannaneh Hajishirzi. Mathqa: Towards interpretable math word problem solving with operation-based formalisms. 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. 2357–2367, 2019.

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.

Justin Chen, Swarnadeep Saha, and Mohit Bansal. Reconcile: Round-table conference improves reasoning via consensus among diverse llms. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 7066–7085, 2024a.

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, 2023. *URL https://arxiv. org/abs/2308.10848*, 6, 2024b.

Hyeong Kyu Choi, Jerry Zhu, and Sharon Li. Debate or vote: Which yields better decisions in multi-agent large language models? *Advances in Neural Information Processing Systems*, 38: 101732–101764, 2026.

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.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems, 2021. *URL https://arxiv. org/abs/2110.14168*, 9, 2021.

Yilun Du, Shuang Li, Antonio Torralba, Joshua B Tenenbaum, and Igor Mordatch. Improving factuality and reasoning in language models through multiagent debate. *arXiv preprint arXiv:2305.14325*, 2023.

Sugyeong Eo, Hyeonseok Moon, Evelyn Hayoon Zi, Chanjun Park, and Heuiseok Lim. Debate only when necessary: Adaptive multiagent collaboration for efficient llm reasoning. *arXiv preprint arXiv:2504.05047*, 2025.

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, 2026.

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. *arXiv preprint arXiv:2407.21783*, 2024.

Yichong Huang, Xiaocheng Feng, Baohang Li, Yang Xiang, Hui Wang, Ting Liu, and Bing Qin. Ensemble learning for heterogeneous large language models with deep parallel collaboration. *Advances in Neural Information Processing Systems*, 37:119838–119860, 2024.

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.

Lars Benedikt Kaesberg, Jonas Becker, Jan Philip Wahle, Terry Ruas, and Bela Gipp. Voting or consensus? decision-making in multi-agent debate. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 11640–11671, 2025.

Guohao Li, Hasan Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem. Camel: Communicative agents for" mind" exploration of large language model society. *Advances in neural information processing systems*, 36:51991–52008, 2023.

Junyou Li, Qin Zhang, Yangbin Yu, Qiang Fu, and Deheng Ye. More agents is all you need. *arXiv preprint arXiv:2402.05120*, 2024.

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 2024 conference on empirical methods in natural language processing*, pp. 17889–17904, 2024.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *International Conference on Learning Representations*, volume 2024, pp. 39578–39601, 2024.

Bo Lv, Nayu Liu, Chen Tang, Xin Liu, Yue Yu, and Ping Luo. Specem: Training-free llm ensembling via iterative drafting, verification, and online feedback. *Advances in Neural Information Processing Systems*, 38:52571–52595, 2026.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan Le Bras, and Yejin Choi. Social iqa: Commonsense reasoning about social interactions. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pp. 4463–4473, 2019.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. Commonsenseqa: A question answering challenge targeting commonsense knowledge. 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. 4149–4158, 2019.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, et al. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*, 2025.

Hongduan Tian, Xiao Feng, Ziyuan Zhao, Rolan Yan, and Bo Han. Multi-agent debate with memory masking. In *International Conference on Learning Representations*, volume 2026, pp. 5122–5143, 2026.

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, 2025.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*, 2022.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, et al. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. *Advances in Neural Information Processing Systems*, 37: 95266–95290, 2024.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the 57th annual meeting of the association for computational linguistics*, pp. 4791–4800, 2019.

Xiutian Zhao, Ke Wang, and Wei Peng. An electoral approach to diversify llm-based multi-agent collective decision-making. In *Proceedings of the 2024 conference on empirical methods in natural language processing*, pp. 2712–2727, 2024.

Mingchen Zhuge, Wenyi Wang, Louis Kirsch, Francesco Faccio, Dmitrii Khizbullin, and Jürgen Schmidhuber. Language agents as optimizable graphs. *arXiv preprint arXiv:2402.16823*, 2024.

A Experimental and Implementation Details

This appendix supplements the main experimental setup with data processing, inference configurations, baseline settings, and implementation details used throughout our empirical evaluation. Additional results and prompt templates are provided in Appendices B and D, respectively.

A.1 Datasets and Evaluation

Table 6 summarizes the evaluation splits and answer formats. HellaSwag and MMLU-Pro use deterministic hash-based sampling after excluding instances used in earlier development experiments. MMLU-Pro samples are allocated proportionally across subjects. Sampling uses seed `2026082501`, dataset and instance identifiers, and subject labels for stratification, without consulting answer labels.

Table 6: Datasets, evaluation splits, and answer formats. The multiple-choice and open-ended suites contain 9732 and 2319 examples, respectively.

Benchmark	N	**Evaluation split**	**Answer format**
ARC-Challenge	1172	Full official test split	Multiple choice; 3-5 options
CommonsenseQA	1221	Full official validation split	Multiple choice; 5 options
HellaSwag	1000	Fixed validation subset	Multiple choice; 4 options
MathQA	2985	Full official test split	Multiple choice; 5 options
MMLU-Pro	1400	Fixed stratified test subset	Multiple choice; 4, 8, 9, or 10 options
SocialIQA	1954	Full official dev split	Multiple choice; 3 options
GSM8K	1319	Full official test split	Numerical answers
MATH-500	500	Full official test split	Mathematical answers and expressions
MBPP	500	Official test split, task IDs 11-510	Python programs

Candidate construction. Multiple-choice evaluation preserves original option identities, including 44 MathQA instances with duplicated option texts but distinct labels. Open-ended answers are merged when their normalized representations match. GSM8K answers are converted to exact rational values. For MATH-500, numerical values and simple numerical fractions are converted to rational representations, while other expressions are retained as normalized strings. MBPP programs are grouped by abstract syntax trees with source location attributes removed; unparseable programs use normalized code text. Candidates retain their original answer texts and initial support counts, and subsequent judgments use these texts rather than internal identifiers.

Outcome evaluation. Candidate construction does not use reference answers and is separate from correctness evaluation of the generated model outputs. GSM8K uses exact matching after numerical normalization of final answers. MATH-500 applies normalized matching, Math-Verify 0.9.0, and symbolic equivalence checking with SymPy 1.14.0 in sequence. MBPP requires all standard and additional challenge tests to pass in an isolated execution environment, with a six-second timeout per candidate. Public example tests are included in the task specification, but execution outcomes and reference-based equivalence checks are used only for offline evaluation.

A.2 Models and Inference Settings

The main ensemble uses BF16 for Qwen3-32B and Gemma 3 27B and an FP8 checkpoint for Llama 3.3 70B. The smaller ensemble consists of Qwen3-8B, Gemma 3 12B IT, and Llama 3.1 8B Instruct, all in BF16. Models are served using vLLM 0.11.2 on an Ubuntu 22.04 server with four NVIDIA GeForce RTX 5090 GPUs, each with 32 GB of memory. Each main-ensemble model uses four-way tensor parallelism; the smaller models use one, two, and one GPUs, respectively. The maximum context length is 16,384 tokens, with GPU memory utilization set to 0.90.

Generation settings. The sampling temperature is fixed at 0.2. For Qwen3 models, `enable_thinking` is set to `false`, with reasoning elicited through the shared prompting interface. Each logical call comprises reasoning generation followed by structured answer readout, implemented as two model requests. Table 7 lists their output limits for each evaluation setting and stage. Multiple-choice readout requests up to 20 token-level log-probability alternatives to construct option support scores. Request seeds are derived from task and call identifiers through a fixed hashing scheme, using master seeds `20260815` for multiple-choice initialization, `2026081537` for subsequent multiple-choice calls, and `2026082901` for open-ended tasks.

Table 7: Maximum output tokens for the two stages of each logical call. These limits differ from the average token consumption reported in the experiments.

Evaluation setting and stage	Reasoning tokens	Readout tokens
Multiple-choice: initialization, comparisons, arbitration, and baselines	1200	100
Open-ended: initial answers	1600	1400
Open-ended: local judgments and arbitration	900	100

Cost accounting. Tokens include input and output tokens from both stages, while Calls/task counts each two-stage evaluation as one logical call. Technical retries are recorded separately and do not increase the logical call budget. Reused outputs are charged to each method that requires them, including the initial responses used to determine disagreement. Serial-equivalent latency sums logical-call durations per example and is averaged over all evaluation instances; it is not parallel wall-clock latency. These definitions provide consistent accounting across compared methods.

A.3 Baseline Configurations

Table 8 summarizes the baseline implementations used in the multiple-choice experiments. Additional inference is restricted to instances with disagreement among the three initial models.

ReConcile assigns voting weights of 0.1, 0.3, 0.5, 0.8, and 1.0 to confidence values in [0, 0.6], (0.6, 0.8), [0.8, 0.9), [0.9, 1), and $\{1\}$, respectively, without additional convincing demonstrations. MAD-M^2 uses subjective correctness judgments rather than cross-model perplexity comparisons for memory masking. Remaining voting or score ties are resolved deterministically.

A.4 PairRecon Implementation

Evaluator assignment. In the three-model configuration, a 2-1 split assigns each model the same candidate pair in both presentation orders, yielding three paired observations. For a 1-1-1 split, each pair is evaluated in reversed orders by two models: one that proposed a candidate within the pair and one whose initial answer lies outside it. A cyclic assignment gives each model two local judgments,

Table 8: Baseline configurations for the multiple-choice evaluation. The table specifies the model backbones, sampling or interaction budgets, and answer aggregation rules used in this work.

Method	Configuration in this evaluation
Best Single	Uses Llama 3.3 70B FP8, the strongest individual model overall on the main multiple-choice suite, consistently across all six benchmarks.
Direct3	Aggregates the three initial models' support scores in log space over the full option set.
Direct9	Obtains two additional responses per model on disagreement instances and applies plurality voting to all nine answers, including the initial three.
Agent Forest	Applies plurality voting to nine independent responses from one fixed model: Llama 3.3 70B FP8 for the main ensemble and Qwen3-8B for the smaller ensemble.
MAD	Uses the shared initial responses as round one, followed by two rounds of all-to-all interaction. Final-round answers are aggregated by plurality voting.
ReConcile	Generates confidence-bearing initial responses and performs up to two further discussion rounds. It stops at unanimous agreement; otherwise, final-round answers are aggregated by confidence-weighted voting.
MAD-M^2	Uses a fixed cyclic assignment to judge another model's initial solution. Solutions judged correct or uncertain are retained for answer revision, followed by plurality voting.

producing one paired observation for each of the three candidate pairs. Both patterns therefore require six local judgments before the resulting observations are used to construct the graph.

Interpretation of paired observations. In a 2-1 split, the two judgments underlying each paired observation come from the same evaluator, so a retained relation indicates consistency under reversal of candidate order. In a 1-1-1 split, the judgments come from different evaluators. Retention therefore indicates agreement between the assigned models under counterbalanced orders, while an erasure may reflect disagreement between models as well as sensitivity to presentation order. In either route, an unresolved comparison contributes no directed edge and provides no support for either candidate. Reconstruction uses the remaining relations together with initial support. If all comparisons are erased, all opposition counts are zero, and reconstruction reduces to maximizing the initial score over the candidate set, with any remaining score ties resolved deterministically.

Local comparison inputs. Candidates are presented anonymously, and selections are mapped back to their original identities. Multiple-choice comparisons provide the question and two candidate option texts. Open-ended comparisons additionally include the candidates' supporting work from initial generation to provide context for each local judgment.

Arbitration inputs. Multiple-choice arbiters receive the question, all original options, anonymous initial candidates, and relational observations, and may select any original option. Open-ended arbiters receive the question, candidate answers, and their supporting work; their selections are restricted to existing candidates, and graph observations are not directly included in the input. In both settings, the graph-reconstructed answer is the current decision, and independent arbitration outputs are combined using the staged update procedure in Section 3.3.

B Additional Experimental Results

This appendix examines PairRecon with a smaller heterogeneous model ensemble, analyzes component effects across benchmarks, and investigates how individual models contribute useful local judgments for resolving model disagreement through targeted comparisons between candidates.

B.1 Results with Smaller Models

We evaluate PairRecon with Qwen3-8B, Gemma 3 12B IT, and Llama 3.1 8B Instruct on the same six multiple-choice benchmarks. As shown in Table 9, PairRecon achieves the highest overall and macro-averaged accuracies of 82.59% and 81.96%, respectively. It improves overall accuracy

Table 9: Accuracy and inference efficiency with the smaller model ensemble. Tokens and serial-equivalent latency are averaged over 9732 examples. Bold and underlined values indicate the best and second-best results. Efficiency rankings exclude Direct3. †MAD-M^2 accuracy excludes three unavailable MMLU-Pro outputs, using 1397 examples for that benchmark and 9729 overall.

Method	Benchmark Accuracy (%)						Aggregate (%)		Efficiency	
	ARC-C	CSQA	HS	MathQA	MMLU-P	SIQA	Overall	Macro	Tokens	Lat. (s)
Direct3	92.92	78.79	76.40	87.40	66.29	75.33	80.39	79.52	2,341	9.92
Direct9	93.26	80.18	<u>79.80</u>	**89.15**	68.07	**77.89**	82.26	81.39	**4,359**	18.56
Agent Forest	93.52	80.67	77.00	88.44	<u>68.71</u>	76.00	81.57	80.72	<u>4,386</u>	18.96
MAD	<u>94.28</u>	80.59	77.10	88.68	**69.00**	76.77	81.93	81.07	7,942	<u>18.40</u>
ReConcile	**94.88**	**82.15**	77.80	88.88	68.00	77.64	<u>82.36</u>	<u>81.56</u>	8,422	26.70
MAD-M^{2†}	94.11	80.75	78.10	88.61	66.93	76.97	81.76	80.91	6,458	19.86
PairRecon	93.94	<u>81.16</u>	**82.10**	<u>88.94</u>	67.79	<u>77.84</u>	**82.59**	**81.96**	4,390	**18.34**

over Direct3 by 2.20% and exceeds the strongest competing baseline, ReConcile, by 0.23%. On HellaSwag, PairRecon achieves 82.10% accuracy, outperforming the second-best method by 2.30%.

The smaller ensemble produces 3292 initial disagreement cases, representing 33.83% of the evaluation set, compared with 21.76% for the main ensemble. On these cases, PairRecon increases accuracy from 55.77% to 62.27%, yielding 214 additional correct answers relative to Direct3. Because consensus outputs remain unchanged, the higher disagreement prevalence allows a comparable improvement on disagreement cases to translate into a larger overall gain. PairRecon also achieves the lowest mean serial-equivalent latency among methods requiring additional inference, while its token consumption remains close to Direct9 and Agent Forest. These results extend the accuracy and efficiency benefits of decision projection to a different ensemble of smaller heterogeneous models.

B.2 Ablation Results across Benchmarks

Table 10 breaks down the component ablations from Section 4.4 by benchmark. The complete method achieves the highest overall accuracy and the best or tied-best results on ARC-Challenge, HellaSwag, and MathQA. Removing adaptive admission lowers accuracy on five of the six benchmarks, with the largest decrease on MMLU-Pro, from 76.50% to 75.64%. This pattern shows that candidate admission contributes useful revisions across several reasoning domains.

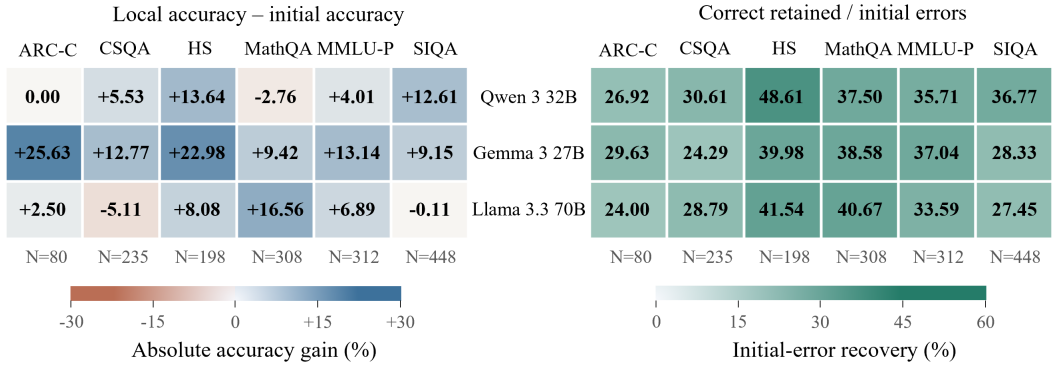
The component effects also vary across benchmarks. On CommonsenseQA, removing adaptive admission increases accuracy from 85.42% to 85.83%, while removing arbitration raises it to 85.59%. Removing the reliability gate instead reduces accuracy to 84.85%. Thus, the admission and arbitration stages contribute to the net degradation on this benchmark, whereas reliability gating remains beneficial. Removing adaptive admission alone does not fully recover the Direct3 accuracy of 85.91%. On MMLU-Pro, removing the reliability gate improves accuracy from 76.50% to 76.86%, although the same ablation reduces accuracy on the other five benchmarks. These results show that the aggregate contribution of a component can coexist with different effects on individual tasks, even within the same evaluation suite. The ablations measure changes to the complete pipeline, so their effects should not be interpreted as independent, additive contributions.

B.3 Detailed Analysis of Local Judgments

Figure 5 analyzes local judgments by model and benchmark on the 1581 eligible 2-1 cases used in the main mechanism analysis. Each case contains a correct candidate and complete paired evaluations for all three models. Panel (a) compares each model's initial-answer accuracy with its local accuracy averaged over both candidate orders on the same examples. Gemma 3 27B improves on all six benchmarks, whereas Qwen3-32B and Llama 3.3 70B FP8 exhibit more task-dependent changes. On MathQA, for example, Qwen3-32B decreases by 2.76%, while Llama 3.3 70B FP8 improves by 16.56%. These contrasting changes show that local comparison affects individual models differently even within the same task domain and on the same evaluation instances.

Table 10: Benchmark-level ablation results on the main model ensemble using shared initial predictions. Bold values indicate the best accuracy in each column, including ties. †Single-order variants modify only the 2-1 route; the original 1-1-1 arbitration outputs are retained.

Variant	Accuracy (%)						
	ARC-C	CSQA	HS	MathQA	MMLU-P	SIQA	Overall
Direct3	96.25	**85.91**	87.00	88.98	72.14	81.06	85.25
w/o Bidirectional (forward)†	96.16	85.18	88.70	90.28	76.21	81.17	86.33
w/o Bidirectional (reverse)†	96.50	85.67	89.00	90.65	76.14	81.47	86.63
w/o Adaptive Admission	96.33	85.83	88.80	90.18	75.64	81.47	86.40
w/o Reliability Gate	96.50	84.85	89.40	90.65	**76.86**	81.53	86.68
w/o 1-1-1 Arbitration	**96.59**	85.59	89.30	90.59	76.07	**81.63**	86.66
PairRecon	**96.59**	85.42	**89.50**	**90.72**	76.50	81.58	**86.76**



(a) From initial answers to local judgments **(b) From errors to correct relations**

Figure 5: Local judgments by model and benchmark on 1581 eligible 2-1 cases. The panels share model rows. (a) Absolute accuracy changes from initial answers to local judgments averaged over both orders. (b) Percentages of initially incorrect answers that yield correct retained relations. N denotes eligible cases per benchmark; recovery denominators are each model's initial errors.

Panel (b) examines whether candidate comparisons recover initially incorrect judgments. Aggregated across benchmarks, the recovery rates are 36.69% for Qwen3-32B, 32.49% for Gemma 3 27B, and 33.73% for Llama 3.3 70B FP8. Each rate divides the number of initially incorrect answers yielding correct retained relations by all initial errors in the corresponding subset, including those followed by incorrect retained or erased observations. Thus, all three models supply correct, order-consistent relations on roughly one-third of their initially incorrect answers within this subset. These recoveries provide useful relational information beyond the initial predictions for global reconstruction and are distinct from the final-answer rescues reported in the main experiments.

C Case Studies

We examine two recorded cases from our experiments that illustrate complementary aspects of PairRecon: recovering a correct minority answer through local comparisons and resolving ambiguity after graph reconstruction through relational arbitration. Qwen, Gemma, and Llama denote Qwen3-32B, Gemma 3 27B, and Llama 3.3 70B FP8, respectively. Questions are lightly edited for readability, and selections are reported using original option identities or answer values.

The two cases trace how an initially available correct answer becomes the final collective decision. Case 1 examines a correct minority candidate that gains stable support when an initially incorrect model revises its judgment during local comparison. In Case 2, the retained relations favor two candidates over a third but leave their mutual preference unresolved. Arbitration then supplies additional support for revising the graph answer. Together, the examples connect model-level error recovery with final-answer correction and distinguish the contributions of local observation, graph reconstruction, and arbitration within the same decision framework.

Case 1: Recovering a Correct Minority Answer

Source. MMLU-Pro, `mmlu_pro:8491`.

Question. Four books on a shelf weigh 2.5, 3.2, 2.7, and 2.3 pounds. Which inequality holds for the weight w of any book chosen from the shelf?

Competing answers. C: $w > 2.3$; I: $w < 3.3$. The correct answer is I. Gemma and Llama initially select C, while Qwen selects I, producing a 2–1 split. Direct3 returns $y_0 = \mathrm{C}$.

Local observations. Each model compares the same two candidates in both presentation orders. The columns below specify the displayed order and report the selected original candidate.

Model	Initial	[C, I]	[I, C]	Observation
Qwen	I	I	I	$\mathrm{I} \succ \mathrm{C}$
Gemma	C	C	I	$\perp$ (erased)
Llama	C	I	I	$\mathrm{I} \succ \mathrm{C}$

Reconstruction and update. Qwen and Llama contribute two stable observations supporting I over C. Gemma's choices disagree under order reversal and contribute no edge. Thus, $W(\mathrm{I}) = 2$, $L(\mathrm{I}) = 0$, and $L(\mathrm{C}) = 2$. PairRecon reconstructs I and accepts it with two stable supports and no opposing observation, satisfying the threshold $\tau = 2$.

$$y_0 = \mathrm{C}\ (w > 2.3) \quad \longrightarrow \quad \hat{y} = \mathrm{I}\ (w < 3.3).$$

Answer check. The book weighing exactly 2.3 pounds violates $w > 2.3$, whereas all four weights satisfy $w < 3.3$.

Case 1 shows how local comparisons recover correct judgments from initially incorrect answers. Llama consistently selects I, while Gemma's order-inconsistent choices are erased. Qwen's stable preference and Llama's recovered judgment jointly support revising the incorrect majority decision, showing that collective correction can proceed despite an unresolved paired observation.

Case 2: Arbitration under Complete Disagreement

Source. MathQA, `math_qa:test:1815`.

Question. A number is "connected with 6" if it is divisible by 6, its digits sum to 6, or it contains the digit 6. How many integers from 1 to 60, inclusive, are not connected with 6?

Options. A: 18; B: 22; C: 42; D: 43; E: 45. The correct answer is D, or 43.

Initial disagreement. Qwen selects 43, Gemma selects 18, and Llama selects 42, producing a 1–1–1 split. Direct3 returns $y_0 = 42$. We use answer values to identify candidates below.

Local observations. Each relation combines judgments from two assigned models under reversed candidate orders.

Pair	Judgment 1	Judgment 2	Observation
$\{43, 18\}$	Qwen: $[43, 18] \to 43$	Llama: $[18, 43] \to 43$	$43 \succ 18$
$\{18, 42\}$	Gemma: $[18, 42] \to 42$	Qwen: $[42, 18] \to 42$	$42 \succ 18$
$\{42, 43\}$	Gemma: $[42, 43] \to 42$	Llama: $[43, 42] \to 43$	$\perp$ (erased)

Here, $\succ$ denotes candidate preference, not numerical comparison.

Graph reconstruction. The retained relations give $L(18) = 2$ and $L(42) = L(43) = 0$. The relation between 42 and 43 is erased. The initial scores, $S(42 \mid x) = -3.26$ and $S(43 \mid x) = -34.04$ after rounding, break the tie in favor of 42, yielding $\widetilde{y} = 42$.

Relational arbitration. The arbiters assess the current decision using the question, candidate answers, and graph observations. Gemma, Llama, and Qwen all select 43. Consequently, $\mathrm{supp}_{\mathcal{O}}(43) = 3 \geq \tau$ and $\mathrm{supp}_{\mathcal{O}}(42) = 0$, allowing 43 to replace the current decision.

$$y_0 = 42 \quad \longrightarrow \quad \widetilde{y} = 42 \quad \xrightarrow{\text{arbitration: } 43,43,43} \quad \hat{y} = 43.$$

Answer check. Exactly 17 integers satisfy at least one of the three conditions, leaving $60 - 17 = 43$ integers not connected with 6.

Case 2 illustrates how arbitration resolves ambiguity after graph reconstruction. The retained relations favor both 42 and 43 over 18 but leave their mutual preference unresolved. Initial scores select 42, while all three arbiters subsequently select 43, providing sufficient unopposed support for correcting the initial decision. Without arbitration, the decision remains incorrectly at 42.

D PROMPT TEMPLATES

This appendix presents the prompts used for independent answer generation, local pairwise evaluation, arbitration, and baseline interactions. The PairRecon templates specify the information available at each decision stage, while the baseline templates record the requests used for additional sampling, inter-agent discussion, and answer revision. Instruction wording is preserved, while repeated instructions are shared and formatting is condensed. Angle brackets denote dynamic inputs or schema placeholders. Bold labels and output schemas are presentation annotations rather than message text.

Each logical call first generates reasoning from the system and user messages. This response is appended as an assistant message before a separate user request for structured readout. Separating reasoning generation from answer readout provides a consistent interface for extracting decisions across models and tasks. Multiple-choice calls use the shared readout below; open-ended readouts are shown with their templates. Output schemas are schematic: all displayed fields are required, additional fields are disallowed, and selections are constrained to the permitted labels.

Shared multiple-choice readout for structured answer extraction. Now output only the sealed JSON decision object, with no reasoning.

Selection fields contain strings. Unless otherwise specified, `option_label` identifies a displayed option, `candidate_label` is `"0"` or `"1"`, and `proposal_number` identifies a displayed proposal. These labels identify the displayed choices and are distinct from the underlying answer content. This separation supports consistent mapping between selections and candidate identities.

D.1 INDEPENDENT ANSWER GENERATION

Direct9 reuses the multiple-choice generation template for additional independent samples, using different seeds. On disagreement cases, each model produces two additional answers to the same question and option set. Together with the three initial predictions, they form the nine-answer vote.

Independent generation: multiple-choice tasks

System. You are an independent expert solver. Follow the requested answer format exactly.

User. Solve the following multiple-choice problem independently. Do not assume that another solver exists. Reason from the problem itself and select exactly one displayed option.

`<question>`
Options: `<option_list>`

Give concise reasoning. A separate sealed readout will request the final option.

Output. `{"selected_option": "<option_label>"}`

Independent generation: open-ended tasks

System. You are an independent expert solver. Follow the requested final-answer format exactly.

User. `<task_instruction>` Work independently and do not assume another solver exists.

`<question>`

Give concise reasoning; a sealed readout will request only the final answer.

Readout. Now output only a sealed JSON object with one string field named final_answer. For MBPP the string is the complete Python program; for math it is only the final value or expression. Include no explanation outside JSON.

Output. `{"final_answer": "<answer_text>"}`

The field <task_instruction> is replaced by the corresponding instruction below:

GSM8K: Solve the word problem carefully. The final answer must be a single exact number.

MATH-500: Solve the mathematics problem rigorously. Preserve an exact symbolic form when appropriate.

MBPP: Write a complete Python solution satisfying the specification and assertions. The final answer must contain only executable Python code.

D.2 Local Pairwise Evaluation

Local queries use anonymous position labels. Comparison inputs include neither model identities nor initial support counts. Reversing candidate order exchanges the complete contents assigned to these labels, including supporting work for open-ended tasks. Readout labels are mapped back to candidate identities before relational observations are constructed. A paired observation is retained only when both mapped selections identify the same candidate; otherwise, it is recorded as an erasure.

Local pairwise evaluation: multiple-choice tasks

System. You are an independent solver in a blinded pairwise-candidate protocol.

User. Solve the following multiple-choice problem independently, but decide only between the two candidate answers shown. Exactly one candidate must be selected. Do not infer why these candidates were paired, do not assume another solver exists, and do not discuss any answer that is not displayed.

<question>

CANDIDATE_0: <candidate_0_text>
CANDIDATE_1: <candidate_1_text>

Give concise private reasoning. A separate sealed readout will request only the selected candidate.

Output. {"selected_candidate": "<candidate_label>"}

Local pairwise evaluation: open-ended tasks

System. You are an independent verifier in a blinded pairwise-candidate protocol.

User. Solve the original task independently, but choose only between the two blinded candidate solutions. Judge correctness, not style or length. Do not invent a third answer and do not infer how the candidates were produced.

ORIGINAL TASK: <question>

CANDIDATE_0:
ANSWER: <candidate_0_answer>
SUPPORTING WORK: <candidate_0_supporting_work>

CANDIDATE_1:
ANSWER: <candidate_1_answer>
SUPPORTING WORK: <candidate_1_supporting_work>

Give concise private verification. A sealed readout will request only candidate 0 or 1.

Readout. Output only a sealed JSON object with selected_candidate equal to "0" or "1".

Output. {"selected_candidate": "<candidate_label>"}

D.3 Relational Arbitration

Multiple-choice arbitration receives the original question, complete option set, anonymous initial proposals, and relational observations. Proposal aliases conceal model identities while preserving the correspondence between candidates and relational observations. Arbiters first solve the problem independently and then assess the anonymous observations as supplementary information. Retained observations identify the favored proposal, whereas erased observations are explicitly described as inconclusive. Arbiters may select any displayed option, including one absent from the initial proposals. Open-ended arbitration instead receives the original task, candidate answers, and their

supporting work, without graph observations. Each arbiter compares the supplied solutions and selects an existing proposal rather than generating a new answer. Returned identifiers are mapped to their corresponding answers before the independent selections are aggregated. In both settings, the resulting structured selections are combined through the staged decision update procedure. The graph-reconstructed answer initializes the current decision before arbitration. A revision requires sufficient support for an alternative and no valid arbitration output supporting the current answer.

Arbitration: multiple-choice tasks

System. You are an impartial final arbiter. Follow the requested answer format exactly.

User. Act as the final arbiter for a difficult multiple-choice problem. First solve the problem independently. Then inspect the anonymous collaboration evidence below as fallible secondary evidence: it may be incomplete or wrong. Select the answer best supported by the problem itself. You may select any displayed option, including one that none of the three anonymous solvers proposed. Do not infer solver identities.

PROBLEM `<question>`

OPTIONS `<option_list>`

ANONYMOUS INITIAL PROPOSALS `<anonymous_proposal_list>`

PAIRWISE PROJECTION EVIDENCE `<relational_observation_list>`

Give concise private reasoning. A separate sealed readout will request only the selected displayed option.

Output. `{"selected_option": "<option_label>"}`

Input entry formats.
Proposal: - `<proposal_alias>`: `<candidate_text>`
Retained observation: - `<left_alias>` versus `<right_alias>`: the counterbalanced comparison stably favored `<winner_alias>`.
Erased observation: - `<left_alias>` versus `<right_alias>`: the counterbalanced comparison was inconclusive.

Arbitration: open-ended tasks

System. You are an independent final verifier in a blinded candidate protocol.

User. Independently verify the original task and select the most correct proposal. Do not create a new answer.

ORIGINAL TASK: `<question>`

`<proposal_blocks>`

Readout. Output only a sealed JSON object with selected_proposal equal to one displayed proposal number.

Output. `{"selected_proposal": "<proposal_number>"}`

Proposal block format.
`PROPOSAL_<proposal_number>`:
ANSWER: `<candidate_answer>`
SUPPORTING WORK: `<candidate_supporting_work>`

One proposal block is inserted for each candidate in its assigned presentation order. Supporting work is taken from the corresponding initial generation for the same example. Each block associates the candidate answer with its supporting work under a single proposal identifier. The returned identifier is mapped back to the selected candidate before arbitration support is counted.

D.4 Baseline Prompts

The field `<question_with_options>` contains the question followed by `Options:` and the option list. Each solution field contains the model's reasoning followed by `Selected answer:`, the selected option label, and its text. For revision stages, these fields supply the earlier solutions that the evaluating model is asked to consider. Placeholders are replaced with the corresponding task

content and recorded responses for each example. The templates below specify which earlier solutions are available to the model at each stage of the corresponding baseline. Different stages within a baseline box represent separate logical calls, each consisting of reasoning generation followed by a structured answer readout requested in a subsequent user message.

Agent Forest, MAD, ReConcile, and the revision stage of MAD-M^2 use the following shared system message. MAD-M^2 memory evaluation uses its own system message, shown below. All stages use the shared multiple-choice readout. Their output schemas distinguish standard option selection, confidence estimation, and the categorical judgments used for memory evaluation.

Shared baseline system for sampling and answer revision across the evaluated methods. You are one independent member of a de-identified multi-agent reasoning team.

Agent Forest: additional independent sampling

User. Solve the following multiple-choice problem independently. Do not assume another solver exists.

`<question_with_options>`

Give concise reasoning. A separate sealed readout will request the final option.

Output. `{"selected_option": "<option_label>"}`

MAD: answer revision

User. These are the recent opinions from other agents:

One peer response: `<peer_solution_1>`

One peer response: `<peer_solution_2>`

This was your most recent opinion: `<own_previous_solution>`

Use these opinions carefully as additional advice. Revise only if warranted by the problem, then give your updated answer with concise reasoning.

Question: `<question_with_options>`

Output. `{"selected_option": "<option_label>"}`

ReConcile: initial generation and discussion

Initial generation — User. Solve the following multiple-choice problem with concise step-by-step reasoning.

`<question_with_options>`

Also estimate the probability that your selected answer is correct as a confidence level from 0.0 to 1.0.

Discussion — User. Carefully review the following solutions from other agents as additional information. State which points you agree or disagree with and why, then provide your own updated answer and concise reasoning.

`<grouped_solutions>`

Question: `<question_with_options>`

Also estimate the probability that your updated answer is correct as a confidence level from 0.0 to 1.0.

Output, both stages.
`{"selected_option": "<option_label>", "confidence_level": <confidence>}`
The field `confidence_level` is numeric in $[0, 1]$.

Answer-group format.
There are `<count>` agents who selected `<option_label>`. `<option_text>`.
One agent solution: `<solution>`

MAD-M^2: memory evaluation and answer revision

Memory evaluation — System. You are a blinded correctness evaluator for one proposed solution.

Memory evaluation — User. Evaluate the proposed solution based only on the question. Return YES if it is completely correct, NO if any material part is incorrect, and NOT_SURE if you cannot determine its correctness.

Question: `<question_with_options>`

Proposed solution: `<proposed_solution>`

Give concise evaluation reasoning. The sealed decision uses 0=YES, 1=NO, 2=NOT_SURE.

Memory evaluation — Output.
`{"selected_option": "<judgment_label>"}`
The field `judgment_label` is `"0"`, `"1"`, or `"2"`.

Answer revision — User, with the shared baseline system. These are the potential solutions retained by a blinded memory-masking step:

`<retained_solution_blocks>`

Use the retained solutions only as additional information. Think step by step and solve the problem.

Question: `<question_with_options>`

Answer revision — Output.
`{"selected_option": "<option_label>"}`

Retained solution block format.
Potential solution `<index>`: `<retained_solution>`

Substitution when no solution is retained.
No proposed solution was retained. Solve independently from the question.