When Debate Helps: Proposal Supply and Verification-Aware Readout in Multi-Agent Reasoning

Zihao Zhao[1,*] **Tunyu Zhang**[2] **Haizhou Shi**[2]
Yusong Zhao[1] **Xinxi Zhang**[2] **Hao Wang**[1,*]
[1]University of Illinois Urbana-Champaign [2]Rutgers University
[*]Corresponding authors: {zihaoz,hwangml}@illinois.edu

Abstract

Multi-agent debate can improve reasoning, yet often fails to beat simple majority voting. We argue that successful debate requires two distinct mechanisms: proposal supply must surface a correct answer, and readout must identify that answer when voting misses it. We formalize the first requirement through recoverable headroom, which measures cases where a correct proposal is available but the majority answer is wrong. For the second, we develop Latent Verification Debate (LVD), an accounting model in which candidate proposals receive answer-specific verification evidence before final generation. Controlled fixed-proposal interventions estimate this latent effect in equivalent peer-support units and show that correct evidence changes answer probabilities and generated decisions while proposal supply remains fixed. To improve proposal supply, we construct societies from neural-thicket agents using labeled and label-free coverage objectives. Across two backbones and matched-budget reasoning benchmarks, coverage-selected societies increase complementary proposal supply and improve aggregate accuracy in repeated stochastic evaluations. Round-level controls further show that interaction provides gains beyond applying the same finalizer directly to the initial proposals. These results identify proposal coverage and truth-sensitive evidence use as complementary conditions for debate to outperform voting. Code is available at https://github.com/Wang-ML-Lab/when-debate-helps.

1 Introduction

Debate lets language-model agents propose, critique, and aggregate alternative reasoning paths [5, 1, 32, 7]. However, it often fails to beat simple majority voting or self-consistency [30, 29]. This raises a basic question: when does interaction help beyond voting?

A prior martingale-null analysis [3] gives a useful baseline: when interaction is symmetric and truth-agnostic, debate can mix proposals without creating positive expected drift toward correctness, and therefore is not guaranteed to improve over majority vote. We use this null as a boundary case: practical debate can succeed only when it adds structure outside neutral proposal mixing.

We argue that two mechanisms determine when this happens. The **proposal** mechanism determines which candidate answers the agent society makes available. The **readout** mechanism distills these candidates into a final answer. For debate to help, the proposal process must surface at least one correct answer, and the readout must select correct proposals rather than merely count how many agents repeat each proposal. When the correct proposal is held by a minority and vote misses it, this becomes a headroom-recovery problem: the proposal stage has hit the answer, but the majority readout has not (Sec. 3). This view connects debate to verifiers, LLM judges, critique, and candidate reranking [4, 14, 13, 38, 15], but treats verification as part of the debate readout rather than only as an external selector.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

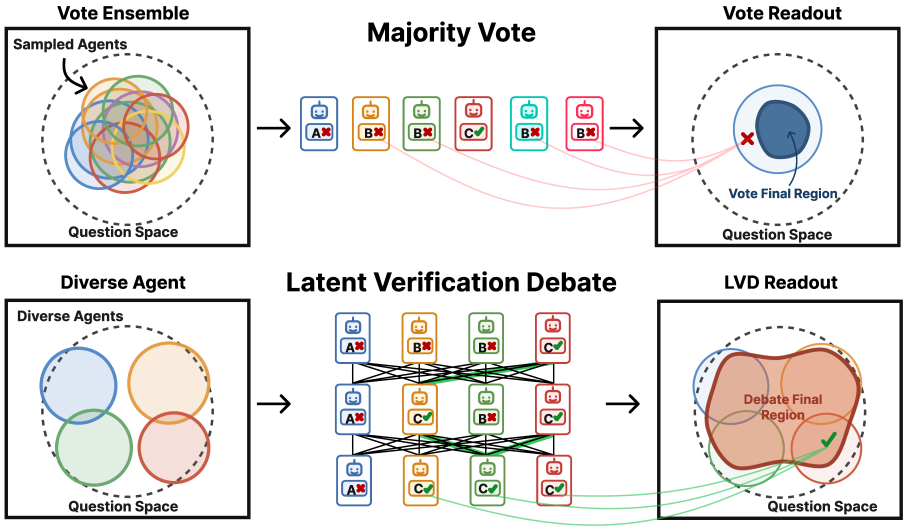

Figure 1: Overview of the supply-readout view. Transparent regions denote questions for which individual sampled agents can surface the correct answer; solid regions denote questions recovered by the final readout. **Top:** Overlapping sampled agents provide redundant proposal coverage, so majority vote keeps a small shared region and can reinforce a repeated wrong answer. **Bottom:** Diverse agents provide complementary proposal coverage; after debate, LVD uses verification-aware evidence to recover a surfaced correct minority answer.

We formalize this framework through recoverable headroom: cases where the proposal stage surfaces the correct answer but majority vote selects a wrong answer. Here, a proposal hit means that at least one surfaced answer matches the correct answer. For readout, Latent Verification Debate (LVD) models candidate-specific verification evidence added to proposal-count dynamics. For supply, we sample neural-thicket agents by applying small perturbations to a pretrained model's weights [8, 37, 20] and use labeled and label-free coverage objectives to select agents with complementary proposal behavior. Fig. 1 summarizes the supply-readout framework.

Across reasoning and multi-discipline benchmarks, we test this framework on Qwen3-4B and OLMo3-7B under matched inference budgets. Coverage-based neural-thicket construction tests the proposal-supply mechanism, while controlled fixed-proposal interventions test whether peer evidence changes readout without changing the candidate set. Repeated stochastic evaluations and round-level controls assess robustness and separate immediate finalizer effects from gains produced by interaction. Our contributions are as follows:

- We introduce recoverable headroom and an exact decomposition that separates recovered headroom, newly correct cases, and damage to vote-correct cases.
- We develop Latent Verification Debate (LVD) as a verification-aware readout model and introduce controlled fixed-proposal probes that operationalize its latent evidence weights without treating them as literal hidden counters.
- We introduce labeled and label-free coverage-based construction for neural-thicket agent pools and show, through matched stochastic replications and mechanism diagnostics, when complementary proposal supply yields gains over Vanilla Debate.

2 Setup and Motivating Observations

We first define the proposal objects shared by vote, debate, and proposal-hit diagnostics, then use two short observations to motivate the theory.

2.1 Proposal Setup and the Martingale Null

We consider reasoning tasks where responses can be compared through a final answer value. Let $\mathcal{X}$ be the input space and let $\mathcal{Y}$ be the decision space. For an input $x \in \mathcal{X}$ with correct answer $y^\star \in \mathcal{Y}$, let $\mathcal{P} = \{e_1, \ldots, e_N\}$ be a candidate agent pool and let $S \subseteq \mathcal{P}$ with $|S| = K$ be the selected society.

We use round $t = 0$ for the initial independent proposals and rounds $t \geq 1$ for subsequent debate updates. At round t, agent e_i produces a response containing a rationale and a normalized answer

$$o_{i,t}(x) = \big(r_{i,t}(x), a_{i,t}(x)\big), \qquad a_{i,t}(x) \in \mathcal{Y}.$$

The rationale $r_{i,t}$ is visible textual evidence. The answer value $a_{i,t}$ is used for vote and proposal-hit diagnostics. Let $\mathcal{M}_t(x, S)$ be the multiset of answer values produced by the selected society at round t. For each candidate answer k, define its proposal count as

$$C_t^{(k)} = \sum\nolimits_{e_i \in S} \mathbf{1}\{a_{i,t}(x) = k\}.$$

Since each selected agent contributes one answer at round t, $\sum_{k \in \mathcal{Y}} C_t^{(k)} = K$. Majority vote operates on $\mathcal{M}_t$:

$$y_t^{\text{vote}}(x, S) \in \arg\max_{k \in \mathcal{Y}} C_t^{(k)}.$$

Ties are resolved by a fixed tie-breaking rule. The proposal-hit diagnostic asks whether the correct answer was surfaced, that is, whether $y^\star$ appears in $\mathcal{M}_t(x, S)$. This diagnostic is not an available readout. It measures whether the proposal supply contains a recoverable correct answer.

In multi-agent debate, agents update from visible prior responses. Let $\mathcal{N}(i) \subseteq S$ be the agents whose previous responses are visible to e_i. At round $t \geq 1$, agent e_i observes

$$\mathcal{F}_{i,t}(x, S) = \{o_{j,t-1}(x) : e_j \in \mathcal{N}(i)\}$$

and updates through a one-round debate operator,

$$o_{i,t}(x) = \mathcal{D}_i(x, \mathcal{F}_{i,t}(x, S)).$$

Here $\mathcal{F}_{i,t}$ is the local context used by agent e_i for its round-t update. We use $\mathcal{F}_t$ to denote the global visible debate history up to round t.

To connect this setup to prior debate-vs-vote theory [3, 21, 35, 12], we use the Dirichlet-categorical-multinomial (DCM) accounting layer as a proposal-count baseline. In this view, a count-based belief state α_t assigns mass to candidate answers. The initial state α_0 represents the model's pre-debate answer belief. Observing proposal counts updates this state symmetrically:

$$\alpha_{t+1}^{(k)} = \alpha_t^{(k)} + C_t^{(k)}. \tag{1}$$

This update records how often answer k is surfaced; it does not know whether k is correct.

Symmetric Truth-Agnostic Null. Under symmetric, truth-agnostic interaction, agents mix surfaced proposals without a signal that preferentially favors the true answer. Prior work formalizes this setting as a martingale over aggregate correctness mass [3]:

$$\mathbb{E}[Q_{t+1} \mid \mathcal{F}_t] = Q_t. \tag{2}$$

Here Q_t denotes the aggregate correctness mass after round t. Thus interaction alone does not create positive expected drift toward correctness. Equivalently, debate that only mixes proposals is not guaranteed to improve over majority vote.

Observation 1: Debate Dynamics Can Depart from the Null. Round-wise debate diagnostics, reported in Fig. 2(a) and detailed in Appendix A.1, Qwen2.5-3B-Instruct [18] shows no positive drift after interaction, which is empirically consistent with null-like proposal mixing. Under the same protocol, Qwen3-4B-Instruct [34] shows persistent gain over vote. We treat this as an empirical boundary: weaker dynamics can look like neutral mixing, while practical debate can contain structure beyond symmetric proposal mixing.

Observation 2: Neural-Thicket Societies Surface Recoverable Expertise. A second observation comes from neural-thicket agent societies [8]. A *neural thicket* is a local pool of models obtained by perturbing pretrained weights θ_0 near a base model, so that each sampled perturbation can behave as a nearby specialist e_m. We use such perturbations as heterogeneous debate agents. Concretely, for base weights θ_0, seed s_m, and noise scale σ_m, a perturbed agent has weights $\theta_m = \theta_0 + \sigma_m \epsilon(s_m)$,. The language-model agent e_m uses the perturbed weights θ_m. Here $\epsilon(s_m)$ is a standard Gaussian noise vector drawn from $\mathcal{N}(0, I)$ using seed s_m; σ_m controls the perturbation scale. The thicket pool is $\mathcal{P}_N(\theta_0) = \{e_m\}_{m=1}^N$. To form a small society, the standard neural-thicket baseline assigns each sampled agent a scalar score v_m on a training set $\mathcal{D}_{\text{train}}$. Typically, v_m is training accuracy (defined

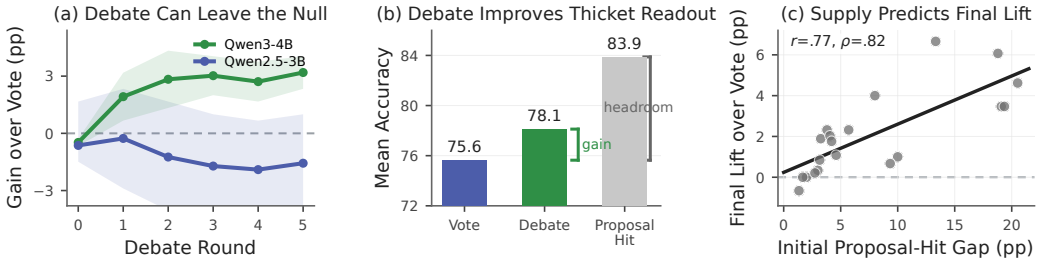

Figure 2: Motivating observations. **(a)** Qwen2.5-3B-Instruct shows null-like debate dynamics with little gain over vote, while Qwen3-4B-Instruct shows positive gain across rounds. **(b)** Under the same neural-thicket supply regime, *Debate* readout improves over *vote*, while the proposal-hit upper bound remains higher, indicating recoverable headroom. **(c)** Final debate lift is positively correlated with initial proposal-hit headroom, suggesting that debate is supply-constrained. Together, these observations motivate a supply-readout account: debate helps when correct minority proposals are surfaced and the readout can recover them through verification-aware dynamics.

in Appendix A.7). The baseline selects the top-K agents by v_m and uses them for thicket vote or debate. Our method keeps the same sample-and-score interface but replaces scalar top-K selection with coverage-based construction (Sec. 4).

These thicket societies allow us to separate proposal supply from final readout. Fig. 2(b) fixes the selected society and compares round-0 vote, final debate readout, and the round-0 proposal-hit upper bound. Debate improves over vote under this fixed supply, so the gain is not merely a larger agent pool or more independent samples. The remaining proposal-hit gap shows that some surfaced correct answers are still missed by the final readout. Fig. 2(c) further shows that round-0 proposal-hit headroom correlates with final lift over vote. This pattern suggests that debate gains are supply-constrained: the readout mostly converts surfaced or early-emerging correct proposals that majority vote does not extract reliably.

Takeaway. The martingale null is a useful boundary case, but not a complete description of practical debate. Under the same supply regime, debate can act as a stronger readout than majority vote; across settings, its lift is tied to round-0 proposal-hit headroom. This motivates our supply-readout view: useful debate requires enough proposal supply to create recoverable headroom and a verification-aware readout to convert that headroom into final accuracy. Sec. 3 formalizes this view.

3 Headroom Recovery Theory

The observations above separate debate into proposal supply and readout. We now formalize this separation using round 0 as the independent proposal stage before interaction. For a selected society S, the round-0 answers define the initial proposal supply, the vote baseline, and the proposal-hit diagnostic. Majority vote is strong because it uses the most common surfaced answer, but it can discard a correct minority proposal. We measure this missed opportunity with recoverable headroom.

3.1 Recoverable Headroom and Gain

For a fixed selected society S, let $\mathcal{M}(x, S) := \mathcal{M}_0(x, S)$ be the round-0 proposal multiset used for headroom accounting. For each instance x, define two indicators:

$$\begin{aligned} V(x, S) &= \mathbf{1}\{\mathrm{Vote}(\mathcal{M}(x, S)) = y^\star\}, \\ P(x, S) &= \mathbf{1}\{\exists a \in \mathcal{M}(x, S) : a = y^\star\}. \end{aligned} \tag{3}$$

Here V records whether majority vote is correct, while P records whether the proposal multiset contains the correct answer. We define society-level recoverable headroom by averaging over the evaluation distribution:

$$\mathcal{H}(S) = \Pr_x(P(x, S) = 1, V(x, S) = 0). \tag{4}$$

Thus $\mathcal{H}(S)$ measures cases where the society contains the correct answer but the majority readout misses it. Let $D_\eta(x, S)$ be the correctness indicator of a final readout with configuration η. We define

the readout's recovery, new-correctness, and damage rates by

$$\begin{aligned} R_\eta(S) &= \Pr(D_\eta = 1 \mid P = 1, V = 0), \\ I_\eta(S) &= \Pr(D_\eta = 1 \mid P = 0, V = 0), \\ L_\eta(S) &= \Pr(D_\eta = 0 \mid V = 1). \end{aligned} \tag{5}$$

Here R_η measures recovery of vote-missed surfaced answers, I_η measures correctness that emerges after the initial proposal stage, and L_η measures harm to vote-correct cases. The net gain over vote is

$$\mathcal{G}_\eta(S) = \mathbb{E}[D_\eta(x, S) - V(x, S)]. \tag{6}$$

Proposition 1 (**Recovery-Damage Gain Decomposition**). *With the initial proposal supply fixed, the gain over vote decomposes as*

$$\mathcal{G}_\eta(S) = \mathcal{H}(S)R_\eta(S) + \Pr(P = 0, V = 0)I_\eta(S) - \Pr(V = 1)L_\eta(S). \tag{7}$$

The proof is in Appendix B.1. This identity is the main accounting result. A readout beats vote when recovered headroom plus new-correctness exceeds damage to vote-correct cases. Our main mechanism claim concerns the first term: LVD should improve the conversion of surfaced correct minorities into final answers. The second term is reported separately because it captures correctness not explained by the initial proposal-hit supply.

3.2 LVD as Verification-Aware Readout

Sec. 2.1 introduced the symmetric truth-agnostic null. We now make the corresponding proposal-count baseline explicit with a Dirichlet-categorical-multinomial (DCM) accounting layer [3]. Note that we use this layer as readout-level bookkeeping over surfaced answer counts, not as a full behavioral model of each debate agent.

More broadly, this probabilistic readout is inspired by the hierarchical Bayesian deep learning perspective [27, 28, 33], in which observed neural signals are coupled with latent probabilistic variables through Bayesian inference. Here, we apply this perspective at the multi-agent readout level, treating surfaced proposals and verification evidence as observations that update a latent distribution over candidate answers.

At round t, let $\mathcal{Y}_t$ be the finite candidate answer set considered by the readout. For multiple-choice tasks, $\mathcal{Y}_t$ is the task-defined option set; for exact-answer tasks, it is the set of surfaced normalized answer clusters in the visible transcript $\mathcal{F}_t$. Let $C_t^{(k)}$ be the count of visible proposals whose normalized answer is k, let $C_t^+ = \sum_{k \in \mathcal{Y}_t} C_t^{(k)}$, and let $q_\theta^0(k \mid x)$ be the readout model θ's pre-debate answer prior over $\mathcal{Y}_t$. A complete single-round readout with one valid answer from each of K agents has $C_t^+ = K$. With prior mass $\kappa > 0$, the proposal-only posterior and predictive are

$$\alpha_t^{\text{prop}}(k) = \kappa q_\theta^0(k \mid x) + C_t^{(k)}, \qquad p_{\text{prop}}(y = k \mid x, \mathcal{F}_t) = \tfrac{\alpha_t^{\text{prop}}(k)}{\kappa + C_t^+}. \tag{8}$$

This proposal-only readout treats surfaced answers symmetrically: an answer gains influence only by appearing more often in the proposal multiset.

Our proposed LVD augments this proposal-count baseline with an effective verified pseudo-count vector $W_{\eta,t}$ [31]. Each $W_{\eta,t}^{(k)} \geq 0$ is the additional verification-aware evidence assigned to answer k by readout configuration η; $W_{\eta,t}$ is a theoretical answer-level abstraction, not an observed hidden state inside the LLM. The LVD posterior is (see the conjugacy argument in Appendix B.2).

$$\begin{aligned} \alpha_{\text{LVD},\eta}^{(k)} &= \kappa q_\theta^0(k \mid x) + C_t^{(k)} + W_{\eta,t}^{(k)}, \\ p_{\text{LVD},\eta}(y = k \mid x, \mathcal{F}_t, W_{\eta,t}) &= \tfrac{\kappa q_\theta^0(k|x) + C_t^{(k)} + W_{\eta,t}^{(k)}}{\kappa + C_t^+ + W_{\eta,t}^+}, \qquad W_{\eta,t}^+ = \sum\nolimits_{k \in \mathcal{Y}_t} W_{\eta,t}^{(k)}. \end{aligned} \tag{9}$$

Following the hierarchical Bayesian deep learning perspective [27, 28, 26], we can further introduce a latent verification variable and marginalize it to obtain the effective verification evidence. One such latent-verification parameterization, proved in Appendix B.3, is $W_{\eta,t}^{(k)} = \tau_{\eta,t} q_\eta^v(k \mid x, \mathcal{F}_t)$, where q_η^v is a soft allocation of verification support and $\tau_{\eta,t}$ is the total effective verification mass. The proposal-only readout is the special case $W_{\eta,t}^{(k)} = 0$ for all k.

Proposition 2 (**Verification-Directed Positive Drift**). *Fix x and $\mathcal{F}_t$, let $y^\star \in \mathcal{Y}_t$ be the correct answer, and define $A_t = \kappa + C_t^+$ and $\alpha_t^\star = \alpha_t^{\mathrm{prop}}(y^\star)$. For a fixed effective pseudo-count vector with $W_{\eta,t}^+ > 0$, let $Q_t^{\mathrm{prop}} := \alpha_t^\star / A_t$. LVD increases posterior mass on $y^\star$ relative to the proposal-count baseline if and only if*

$$\frac{W_{\eta,t}^{(y^\star)}}{W_{\eta,t}^+} > \frac{\alpha_t^\star}{A_t} = Q_t^{\mathrm{prop}}. \tag{10}$$

The proof is in Appendix B.4. This is the readout-level analogue of the martingale-null comparison. Symmetric proposal mixing gives no positive expected drift in correctness mass; LVD creates non-null drift only when verified pseudo-counts allocate more mass to $y^\star$ than the proposal-only posterior already does.

Proposition 3 (**Correct-Minority Recovery**). *On a headroom instance, suppose the proposal-count winner is an incorrect answer $y_v \neq y^\star$, while $y^\star \in \mathcal{Y}_t$. LVD ranks the correct minority above the proposal-count winner if and only if*

$$W_{\eta,t}^{(y^\star)} - W_{\eta,t}^{(y_v)} > \alpha_t^{\mathrm{prop}}(y_v) - \alpha_t^{\mathrm{prop}}(y^\star). \tag{11}$$

The proof is in Appendix B.5. The right side is the proposal-layer disadvantage of the correct minority, including both raw proposal counts and the base prior. The left side is its verified-evidence advantage. Thus LVD recovers a correct minority when verification-aware evidence overcomes the proposal-layer advantage of the wrong winner. This does not make debate universally helpful: gains still require recoverable headroom and enough recovery to outweigh damage to vote-correct cases.

4 Theory-Guided System Design

Proposition 1$\sim$2 suggest two design components: (1) We should increase proposal-hit supply,

$$\mathcal{P}_{\mathrm{hit}}(S) = \Pr(P(x,S)=1) = \Pr(V=1) + \mathcal{H}(S),$$

which can improve vote accuracy and create more vote-missed correct proposals for the readout to recover. (2) When a correct candidate is surfaced, the readout should assign it larger verified evidence; otherwise extra supply need not improve final accuracy. We therefore use coverage-based society construction for supply and a verification-aware readout configuration.

4.1 Coverage-Based Society Construction

The construction step chooses which sampled thicket agents enter the society. Instead of keeping the K agents with the largest individual training-set scores v_m from Eqn. 17, we choose a small society $S \subseteq \mathcal{P}_N(\theta_0)$ whose members cover complementary proposal hits. For a pool $\mathcal{P}_N$ and budget K, the ideal supply target is

$$\mathcal{P}_{\mathrm{hit},K}^\star(\mathcal{P}_N) = \max_{S \subseteq \mathcal{P}_N,\ |S|=K} \Pr(P(x,S)=1). \tag{12}$$

We approximate this target on the construction split, either with labels or with an agreement proxy.

4.2 Answer-Coverage and Support-Coverage Greedy

Directly optimizing proposal-hit supply on the held-out distribution is unavailable during construction because it depends on labels. We instead use a construction-set surrogate. Let $\mathcal{U}_{\mathrm{cal}} = \mathcal{D}_{\mathrm{train}}$ be the construction split, let $\mathcal{P} = \mathcal{P}_N(\theta_0)$ be the sampled agent pool, and let $a_e(u)$ be agent e's normalized initial answer on question u. For a proxy score $\psi(e,u) \in [0,1]$, the selected society maximizes

$$\begin{aligned} S_K &\in \arg\max_{S \subseteq \mathcal{P},\ |S|=K} \widehat{\mathcal{C}}_\psi(S), \\ \widehat{\mathcal{C}}_\psi(S) &= \frac{1}{|\mathcal{U}_{\mathrm{cal}}|} \sum\nolimits_{u \in \mathcal{U}_{\mathrm{cal}}} \max_{e \in S} \psi(e,u). \end{aligned} \tag{13}$$

The max makes this a question-level coverage objective: a question contributes once any selected agent receives a high proxy score. Exact maximization is a cardinality-constrained maximum-coverage problem, so we use the standard greedy approximation [16]. Starting from $S_0 = \emptyset$, step m chooses

$$e_m \in \arg\max_{e \in \mathcal{P} \setminus S_{m-1}} \left[\widehat{\mathcal{C}}_\psi(S_{m-1} \cup \{e\}) - \widehat{\mathcal{C}}_\psi(S_{m-1}) \right], \tag{14}$$

Table 1: Accuracy (%) on Qwen3-4B-Instruct and OLMo-3-7B-Instruct under a unified computation budget. SC and Neural thicket vote rows use `vote20`; debate rows use round-3 (R3) majority vote from matched five-round, four-agent traces. MMLU-Redux is split into five 60-example subjects. Avg is the unweighted mean over the eight displayed columns, and Avg Ranking is the mean within-block method rank. **Boldface** and underline denote the best and second-best result per column within each method block and model. Gray rows mark our proposed agent pool construction methods.

Method	Reasoning			MMLU-Redux					Avg	Avg Ranking
	AMC12	MATH500	GPQA	Moral	Chem.	CS	Acct.	Med.		
Backbone: Qwen3-4B-Instruct										
Label-Free Methods										
SC@20	91.20	92.33	49.13	30.00	70.00	**81.67**	75.00	83.33	71.58	2.38
Vanilla Debate	**92.69**	91.67	**52.31**	50.00	**71.67**	80.00	**76.67**	83.33	74.79	**1.81**
SC-Greedy Debate (Ours)	92.29	**92.67**	51.45	**63.33**	68.33	**81.67**	73.33	**86.67**	**76.22**	**1.81**
Label-Based Methods										
Neural Thicket Vote	90.93	**91.33**	50.00	70.00	63.33	80.00	**76.67**	81.67	75.49	2.19
Top-4 Debate	92.15	90.67	52.31	68.33	66.67	78.33	75.00	**88.33**	76.47	2.25
AC-Greedy Debate (Ours)	**92.29**	**91.33**	**52.60**	**73.33**	**68.33**	**85.00**	73.33	80.00	**77.03**	**1.56**
Backbone: OLMo-3-7B-Instruct										
Label-Free Methods										
SC@20	**87.01**	**88.33**	35.55	25.00	50.00	61.67	56.67	61.67	58.24	2.12
Vanilla Debate	86.33	85.33	**39.31**	23.33	48.33	68.33	**66.67**	63.33	60.12	2.00
SC-Greedy Debate (Ours)	84.03	84.00	37.57	**28.33**	**56.67**	**76.67**	55.00	**70.00**	**61.53**	**1.88**
Label-Based Methods										
Neural Thicket Vote	**86.87**	84.67	**37.57**	28.33	51.67	**78.33**	46.67	**71.67**	60.72	**2.00**
Top-4 Debate	85.93	84.33	36.42	**36.67**	56.67	**78.33**	46.67	**71.67**	62.09	**2.00**
AC-Greedy Debate (Ours)	85.25	**85.00**	35.84	35.00	**58.33**	73.33	**56.67**	**71.67**	**62.64**	**2.00**

and sets $S_m = S_{m-1} \cup \{e_m\}$ until $m = K$.

Answer-Coverage Greedy (AC-Greedy). When labels are available, AC-Greedy uses

$$\psi_{\mathrm{AC}}(e, u) = \mathbf{1}\{a_e(u) = y^\star(u)\}. \tag{15}$$

It chooses agents by marginal increase in labeled answer coverage on the construction split and serves as a label-based construction reference.

Support-Coverage Greedy (SC-Greedy). SC-Greedy removes the need for labels by computing coverage from agent agreement. For each construction question u, we cluster valid extracted answers. Let $c_e(u)$ be the cluster containing agent e's answer, and let $\widehat{p}_u(c)$ be the fraction of valid extracted answers in cluster c:

$$\widehat{p}_u(c) = \frac{|\{e \in \mathcal{P} : c_e(u) = c\}|}{|\{e \in \mathcal{P} : e \text{ has a valid extracted answer on } u\}|}.$$

If no agent has a valid extracted answer on u, all support scores for that question are set to zero. The label-free proxy is

$$\psi_{\mathrm{SC}}(e, u) = \begin{cases} \widehat{p}_u(c_e(u)), & \text{if } e \text{ has a valid extracted answer,} \\ 0, & \text{otherwise.} \end{cases} \tag{16}$$

This score is not a majority-only indicator: it can still select complementary agents.

Verification-Aware Readout. We pair the coverage-selected societies with a fixed debate instruction and a vote-then-judge finalizer. Sec. 5 tests the observable predictions of LVD through controlled interventions that hold proposal supply fixed.

5 Experiments

We test whether the supply-readout view predicts when debate improves over vote. We evaluate two main debate backbones, Qwen3-4B-Instruct and OLMo-3-7B-Instruct [34, 24], on mathematical reasoning (AMC12, MATH500) [11], scientific question answering (GPQA) [19], and multi-discipline reasoning (MMLU-Redux) [10, 9]. Each debate run uses $K = 4$ agents and $T = 5$ rounds. We compare Answer-Coverage Greedy (**AC-Greedy**) and Support-Coverage Greedy (**SC-Greedy**) against baselines including Self-Consistency (**SC**) [30], **Vanilla Debate** [5], **Neural Thicket Vote**,

Table 2: AMC12-selected seed transfer to MMLU-Redux. Init, R3, and R5 report pre-debate, round-3, and round-5 metrics over the same four agents. Final is the R5 vote-then-judge readout; Lift is Final minus Init Vote. Values are percentages except N. Gray rows mark our greedy construction methods.

Method	N	Init Vote	R3 Vote	R5 Vote	Final	Init Hit	R3 Hit	R5 Hit	Lift
Backbone: Qwen3-4B-Instruct, AMC12 → MMLU-Redux									
SC-Greedy Debate (Ours)	300	72.33	75.67	76.33	76.33	**84.00**	77.67	76.33	**+4.00**
Top-4 Debate	300	72.67	75.00	74.67	75.33	**84.00**	77.67	76.67	+2.67
AC-Greedy Debate (Ours)	300	**75.33**	**77.33**	**77.00**	**76.67**	83.67	**78.00**	**78.00**	+1.33
Backbone: OLMo-3-7B-Instruct, AMC12 → MMLU-Redux									
SC-Greedy Debate (Ours)	300	55.33	**56.33**	**58.33**	**58.33**	**70.00**	**59.67**	**59.33**	**+3.00**
Top-4 Debate	300	**56.00**	**56.33**	56.33	56.67	67.00	57.33	57.00	+0.67
AC-Greedy Debate (Ours)	300	54.00	**56.33**	56.67	56.33	69.33	58.00	57.33	+2.33

and **Neural Thicket Debate** [8] under matched vote/debate budgets. Full details on dataset splits, thicket-pool construction, decoding parameters, token budgets, prompts, context visibility, and tie-breaking are in Appendices A.2–A.8; complete round-level statistics are in Appendix A.12.

Diverse Expert Supply for Debate Systems. Table 1 asks whether our proposed coverage-selected thicket agents form stronger debate societies than vote-only scaling, Vanilla Debate, and top-score thicket baselines. We use a unified computation budget: vote rows use `vote20`, while debate rows use the round-3 (R3) majority state from matched four-agent traces. Across different backbones, our Support-Coverage Greedy debate has the best average within the label-free block, and our Answer-Coverage Greedy debate has the best average within the label-based block. The gains are clearest on the MMLU-Redux split, where coverage-based selection improves several subjects even when the aggregate average is close.

Stochastic Robustness. Table 1 retains the submitted method-specific decoding configuration and its fixed global seed. In a separate robustness experiment, we rerun Vanilla Debate, SC-Greedy, and AC-Greedy with a common stochastic configuration (temperature 0.7, top-$p = 0.95$) across five matched generation seeds. Relative to Vanilla Debate, SC-Greedy and AC-Greedy improve the eight-column average by $+1.14$ points (paired 95% CI $[0.60, 1.67]$) and $+1.46$ points ($[0.01, 2.91]$) on Qwen3-4B, and by $+0.96$ points ($[0.02, 1.90]$) and $+2.01$ points ($[1.32, 2.69]$) on OLMo3-7B. These intervals quantify generation-seed variability rather than uncertainty from resampling test questions; complete per-benchmark results and paired tests are reported in Appendix A.5.

Can Coverage-Based Selection Transfer Out of Domain? We next ask whether coverage-selected societies only improve the domain used for selection, or whether their complementarity transfers out of domain. We select agents on AMC12 and evaluate the same selected societies on MMLU-Redux without reselecting agents. This setting tests whether diverse proposal supply selected from one reasoning domain remains useful under a multi-discipline target shift.

Table 2 shows positive transfer for the proposed coverage-selected societies, but the stronger construction varies by backbone. On Qwen3-4B-Instruct, our Answer-Coverage Greedy achieves the best final accuracy, while our Support-Coverage Greedy achieves the largest improvement over the initial vote. On OLMo-3-7B-Instruct, Support-Coverage Greedy achieves both the best final accuracy and the largest improvement. This suggests that coverage selection can capture reusable agent complementarity, although transfer is not uniform across backbones. The headroom, recovery, and damage decomposition for these transfer runs is in Appendix A.11.

Dynamics and Mechanism Analysis. Fig. 3 shows that early rounds can contain more proposal-hit coverage than majority vote uses, while later rounds increase consensus but reduce the alternatives. To isolate interaction, we apply the same vote-then-judge finalizer independently at every saved round. Across four societies and five benchmarks, R0 accuracy rises only from 76.09% to 76.45%, whereas the R2 and R3 states reach 78.34% and 78.52%—1.89 and 2.07 points beyond the R0 readout. An immediate finalizer therefore cannot explain the R2–R3 plateau. On OLMo MATH500, all four societies retain 3.33–6.33 points of headroom but have negative final lift, identifying a conversion bottleneck. Appendix A.10 gives both controls, and Appendix A.12 gives complete recovery and damage results.

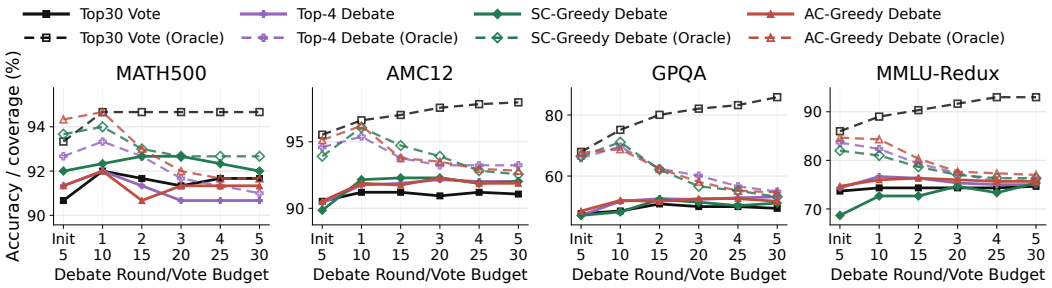

Figure 3: Label-based five-round dynamics. Solid lines show round-level accuracy before finalization; dashed hollow legend entries marked "Oracle" show proposal-hit coverage. Later rounds can improve vote accuracy while reducing answer diversity, so final gains depend on recovering correct proposals before coverage collapses. The label-free counterpart is in Fig. 4.

Controlled Fixed-Proposal Readout Probes. We directly test whether evidence assigned to surfaced candidates changes answer selection while proposal supply remains fixed. For each backbone–dataset pair, we uniformly sample 50 traces at random from the saved traces whose initial proposal set contains at least one correct and one incorrect answer. Within each trace, we hold the question, target agent, initial response, prompt style, and candidate set fixed. We vary only the peer evidence supplied to the target agent.

We first estimate the effect of an original correct peer response in equivalent-support units. Let N denote a neutral peer context, G_d a context containing $d \in \{1, 2, 3\}$ short messages supporting the gold candidate y_g, and T the original context containing a correct peer response from the saved trace. For a designated wrong candidate y_w, we measure

$$m(x) = \log p(y_g \mid x) - \log p(y_w \mid x).$$

The controlled shift is $D_d = m(G_d) - m(N)$, and the original trace-response shift is $\Delta m_{\text{trace}} = m(T) - m(N)$. For each recovery case, where the target initially answers incorrectly while a correct peer proposal is available, we interpolate the observed $d = 0, 1, 2, 3$ dose-response curve to find $\widehat{W}$, the controlled-support dose that matches Δm_{trace}. We treat $\widehat{W}$ as an operational equivalent-support measure, not a literal internal counter.

We next test whether the effect depends on evidential content rather than repetition of an answer. From the saved traces, we construct matched correct and wrong peer rationales, remove all stated answers and option references, and match each pair in length. We then compare the gold-versus-other-answer margin after the correct rationale, T, and the wrong rationale, F. A positive $T - F$ means that the answer-stripped correct reasoning leaves the model more favorable to the gold answer than equally long wrong reasoning.

Finally, we test whether these probability shifts alter generated decisions. From the same fixed mixed-proposal states, we generate fresh answers after $d \in \{1, 2, 3\}$ peer messages under correct-support and matched control conditions. The controls provide no candidate evidence, withhold the supported answer, or insert an answer-stripped rationale from another question. Table 3 reports the original trace effect, equivalent support, answer-stripped correct-over-wrong gap, and free-generation accuracy gain at $d = 3$.

Original correct peer responses have positive margin effects in all four backbone–dataset combinations. Their median effects correspond to roughly two controlled messages on GPQA and more than three on MMLU-Redux. The answer-stripped $T - F$ gap is also positive in every backbone–dataset combination. At $d = 3$, correct support improves accuracy over neutral context in all four combinations; averaged across them, accuracy rises from .325 to .655. Because the proposal set never changes, these effects isolate truth-sensitive readout rather than proposal supply. Full intervention tables and configuration controls are in Appendix A.9 and Appendix A.9.

Table 3: Controlled fixed-proposal evidence for verification-aware readout. Trace effect is the answer-margin shift from an original correct peer response relative to neutral context; $\widehat{W}$ is its median equivalent controlled support over recovery cases. $T - F$ is the gold-margin difference between answer-stripped correct and wrong rationales. ΔAcc is correct-support accuracy minus neutral accuracy at $d = 3$. All entries use the main readout configuration; secondary configuration controls are reported in the appendix.

Backbone	Dataset	Δm_{trace} [95% CI]	Median $\widehat{W}$	Answer-stripped $T - F$	ΔAcc at $d = 3$
Qwen3-4B	GPQA	8.15 [4.96, 11.41]	2.23	4.59	+.14
Qwen3-4B	MMLU-Redux	11.40 [9.02, 13.97]	> 3	5.50	+.04
OLMo3-7B	GPQA	4.57 [3.65, 5.50]	1.92	2.01	+.56
OLMo3-7B	MMLU-Redux	5.47 [4.52, 6.43]	> 3	2.10	+.58

6 Related Work

Debate, Vote, and Null Models. Multi-agent debate asks several language-model agents to propose, critique, and revise answers before a final decision [5, 1, 32, 2]. Empirical results vary across tasks and aggregation rules [22, 29, 6, 36], and majority vote remains a strong baseline [30, 12]. Recent theory formalizes a symmetric, truth-agnostic martingale null for debate-like interaction [3]. We use that result as a proposal-count baseline and show how verification-directed pseudo-counts can create readout-level positive drift.

Verification and Judging. Our readout model connects to work on verifiers, LLM-as-a-judge methods [23, 39], critique, and reranking [4, 14, 13, 38, 15]. OrchRM uses execution artifacts to learn rewards for orchestrator training and test-time scaling [25]. LVD places answer-specific verification within debate, where it changes proposal pseudo-counts before final generation. Our probabilistic treatment also draws on hierarchical Bayesian deep learning [27, 28], while placing the latent probabilistic structure at the multi-agent readout level.

Neural Thickets and Coverage Selection. Neural thickets show that nearby perturbations of a base model can form an agent society (pool) with heterogeneous strengths [8, 17]. We use this pool as the proposal-supply side of debate. Our coverage constructor selects complementary specialists from the thicket rather than ranking agents only by individual average accuracy. This connects to coverage objectives for budgeted subset selection [16]. Our coverage signal can be supervised, using true correctness on a construction set, or label-free, using answer-cluster support.

7 Conclusion

Debate helps when proposal supply and readout succeed together. Coverage-based neural-thicket construction increases complementary correct proposals, while LVD uses answer-specific evidence to improve their readout. Controlled experiments confirm truth-sensitive readout effects, and matched stochastic evaluations show gains over Vanilla Debate across both backbones, though headroom conversion remains task- and backbone-dependent.

These results do not imply that debate is generally better than voting. Interaction can provide additional gains but may also discard useful alternatives or damage vote-correct cases. Its practical value therefore depends on proposal headroom, the backbone's ability to evaluate competing evidence, and inference cost. See Appendix C for detailed limitations.

Acknowledgments and Disclosure of Funding

We thank all reviewers, the area chair, and the senior area chair for their valuable comments.

Funding. HW is supported by the Amazon Faculty Research Award, Microsoft AI & Society Fellowship, Nokia Faculty Research Award, NSF CAREER Award IIS-2340125, NIH grant R01CA297832, and NSF grant IIS-2127918.

Competing interests. The authors declare no competing interests.

References

[1] C.-M. Chan, W. Chen, Y. Su, J. Yu, W. Xue, S. Zhang, J. Fu, and Z. Liu. Chateval: Towards better llm-based evaluators through multi-agent debate. In *The Twelfth International Conference on Learning Representations*, 2024.

[2] J. Chen, S. Saha, and M. 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)*, pages 7066–7085, 2024.

[3] H. K. Choi, J. Zhu, and S. Li. Debate or vote: Which yields better decisions in multi-agent large language models? In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2026.

[4] K. Cobbe, V. Kosaraju, M. Bavarian, M. Chen, H. Jun, L. Kaiser, M. Plappert, J. Tworek, J. Hilton, R. Nakano, C. Hesse, and J. Schulman. Training verifiers to solve math word problems, 2021.

[5] Y. Du, S. Li, A. Torralba, J. B. Tenenbaum, and I. Mordatch. Improving factuality and reasoning in language models through multiagent debate. In *International Conference on Machine Learning*, pages 11733–11763. PMLR, 2024.

[6] A. Estornell and Y. Liu. Multi-llm debate: Framework, principals, and interventions. *Advances in Neural Information Processing Systems*, 37:28938–28964, 2024.

[7] A. Estornell, J.-F. Ton, Y. Yao, and Y. Liu. Acc-debate: An actor-critic approach to multi-agent debate. In *The Thirteenth International Conference on Learning Representations*, 2025.

[8] Y. Gan and P. Isola. Neural thickets: Diverse task experts are dense around pretrained weights, 2026.

[9] A. P. Gema, J. O. J. Leang, G. Hong, A. Devoto, A. C. M. Mancino, R. Saxena, X. He, Y. Zhao, X. Du, M. R. Ghasemi Madani, C. Barale, R. McHardy, J. Harris, J. Kaddour, E. Van Krieken, and P. Minervini. Are we done with MMLU? In L. Chiruzzo, A. Ritter, and L. Wang, editors, *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 5069–5096, Albuquerque, New Mexico, Apr. 2025. Association for Computational Linguistics.

[10] D. Hendrycks, C. Burns, S. Basart, A. Zou, M. Mazeika, D. Song, and J. Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations*, 2021.

[11] D. Hendrycks, C. Burns, S. Kadavath, A. Arora, S. Basart, E. Tang, D. Song, and J. Steinhardt. Measuring mathematical problem solving with the math dataset, 2021.

[12] L. B. Kaesberg, J. Becker, J. P. Wahle, T. Ruas, and B. Gipp. Voting or consensus? decision-making in multi-agent debate. In W. Che, J. Nabende, E. Shutova, and M. T. Pilehvar, editors, *Findings of the Association for Computational Linguistics: ACL 2025*, pages 11640–11671, Vienna, Austria, July 2025. Association for Computational Linguistics.

[13] Y. Li, Z. Lin, S. Zhang, Q. Fu, B. Chen, J.-G. Lou, and W. Chen. Making language models better reasoners with step-aware verifier. In A. Rogers, J. Boyd-Graber, and N. Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5315–5333, Toronto, Canada, July 2023. Association for Computational Linguistics.

[14] H. Lightman, V. Kosaraju, Y. Burda, H. Edwards, B. Baker, T. Lee, J. Leike, J. Schulman, I. Sutskever, and K. Cobbe. Let's verify step by step, 2023.

[15] A. Madaan, N. Tandon, P. Gupta, S. Hallinan, L. Gao, S. Wiegreffe, U. Alon, N. Dziri, S. Prabhumoye, Y. Yang, S. Gupta, B. P. Majumder, K. Hermann, S. Welleck, A. Yazdanbakhsh, and P. Clark. Self-refine: Iterative refinement with self-feedback. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 46534–46594. Curran Associates, Inc., 2023.

[16] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher. An analysis of approximations for maximizing submodular set functions—i. *Mathematical programming*, 14(1):265–294, 1978.

[17] X. Qiu, Y. Gan, C. F. Hayes, Q. Liang, Y. Xu, R. Dailey, E. Meyerson, B. Hodjat, and R. Miikkulainen. Evolution strategies at scale: Llm fine-tuning beyond reinforcement learning, 2026.

[18] Qwen, A. Yang, B. Yang, B. Zhang, B. Hui, B. Zheng, B. Yu, C. Li, D. Liu, F. Huang, H. Wei, H. Lin, J. Yang, J. Tu, J. Zhang, J. Yang, J. Yang, J. Zhou, J. Lin, K. Dang, K. Lu, K. Bao, K. Yang, L. Yu, M. Li, M. Xue, P. Zhang, Q. Zhu, R. Men, R. Lin, T. Li, T. Tang, T. Xia, X. Ren, X. Ren, Y. Fan, Y. Su, Y. Zhang, Y. Wan, Y. Liu, Z. Cui, Z. Zhang, and Z. Qiu. Qwen2.5 technical report, 2025.

[19] D. Rein, B. L. Hou, A. C. Stickland, J. Petty, R. Y. Pang, J. Dirani, J. Michael, and S. R. Bowman. GPQA: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*, 2024.

[20] H. Shi, Y. Wang, L. Han, H. Zhang, and H. Wang. Training-free bayesianization for low-rank adapters of large language models. In *ICLR Workshop: Quantify Uncertainty and Hallucination in Foundation Models: The Next Frontier in Reliable AI*, 2025.

[21] A. Smit, P. Duckworth, N. Grinsztajn, K. ab Tessera, T. Barrett, and A. Pretorius. Are we going MAD? benchmarking multi-agent debate between language models for medical q&a. In *Deep Generative Models for Health Workshop NeurIPS 2023*, 2023.

[22] A. P. Smit, N. Grinsztajn, P. Duckworth, T. D. Barrett, and A. Pretorius. Should we be going mad? a look at multi-agent debate strategies for llms. In *International Conference on Machine Learning*, pages 45883–45905. PMLR, 2024.

[23] S. Tan, S. Zhuang, K. Montgomery, W. Y. Tang, A. Cuadron, C. Wang, R. A. Popa, and I. Stoica. Judgebench: A benchmark for evaluating llm-based judges, 2025.

[24] Team Olmo, A. Ettinger, A. Bertsch, B. Kuehl, D. Graham, D. Heineman, D. Groeneveld, F. Brahman, F. Timbers, H. Ivison, J. Morrison, J. Poznanski, K. Lo, L. Soldaini, M. Jordan, M. Chen, M. Noukhovitch, N. Lambert, P. Walsh, P. Dasigi, R. Berry, S. Malik, S. Shah, S. Geng, S. Arora, S. Gupta, T. Anderson, T. Xiao, T. Murray, T. Romero, V. Graf, A. Asai, A. Bhagia, A. Wettig, A. Liu, A. Rangapur, C. Anastasiades, C. Huang, D. Schwenk, H. Trivedi, I. Magnusson, J. Lochner, J. Liu, L. J. V. Miranda, M. Sap, M. Morgan, M. Schmitz, M. Guerquin, M. Wilson, R. Huff, R. L. Bras, R. Xin, R. Shao, S. Skjonsberg, S. Z. Shen, S. S. Li, T. Wilde, V. Pyatkin, W. Merrill, Y. Chang, Y. Gu, Z. Zeng, A. Sabharwal, L. Zettlemoyer, P. W. Koh, A. Farhadi, N. A. Smith, and H. Hajishirzi. Olmo 3, 2026.

[25] K. Y. Tsang, Z. Zhao, V. Venkataramani, H. Shi, Z. Ke, S. Yavuz, S. Joty, and H. Wang. Reward modeling for multi-agent orchestration, 2026.

[26] H. Wang, S. Xingjian, and D.-Y. Yeung. Natural-parameter networks: A class of probabilistic neural networks. In *NIPS*, pages 118–126, 2016.

[27] H. Wang and D.-Y. Yeung. Towards bayesian deep learning: A framework and some existing methods. *TDKE*, 28(12):3395–3408, 2016.

[28] H. Wang and D.-Y. Yeung. A survey on bayesian deep learning. *CSUR*, 53(5):1–37, 2020.

[29] Q. Wang, Z. Wang, Y. Su, H. Tong, and Y. Song. Rethinking the bounds of llm reasoning: Are multi-agent discussions the key? In *62nd Annual Meeting of the Association for Computational Linguistics, ACL 2024*, pages 6106–6131. Association for Computational Linguistics (ACL), 2024.

[30] X. Wang, J. Wei, D. Schuurmans, Q. V. Le, E. H. Chi, S. Narang, A. Chowdhery, and D. Zhou. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*, 2023.

[31] Z. Wang and H. Wang. Variational imbalanced regression: Fair uncertainty quantification via probabilistic smoothing. In *NeurIPS*, 2023.

[32] K. Xiong, X. Ding, Y. Cao, T. Liu, and B. Qin. Examining inter-consistency of large language models collaboration: An in-depth analysis via debate. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 7572–7590, 2023.

[33] Z. Xu, G. Hao, H. He, and H. Wang. Domain indexing variational bayes: Interpretable domain index for domain adaptation. In *ICLR*, 2023.

[34] A. Yang, A. Li, B. Yang, B. Zhang, B. Hui, B. Zheng, B. Yu, C. Gao, C. Huang, C. Lv, C. Zheng, D. Liu, F. Zhou, F. Huang, F. Hu, H. Ge, H. Wei, H. Lin, J. Tang, J. Yang, J. Tu, J. Zhang, J. Yang, J. Yang, J. Zhou, J. Zhou, J. Lin, K. Dang, K. Bao, K. Yang, L. Yu, L. Deng, M. Li, M. Xue, M. Li, P. Zhang, P. Wang, Q. Zhu, R. Men, R. Gao, S. Liu, S. Luo, T. Li, T. Tang, W. Yin, X. Ren, X. Wang, X. Zhang, X. Ren, Y. Fan, Y. Su, Y. Zhang, Y. Zhang, Y. Wan, Y. Liu, Z. Wang, Z. Cui, Z. Zhang, Z. Zhou, and Z. Qiu. Qwen3 technical report, 2025.

[35] H. Zhang, Z. Cui, J. Chen, X. Wang, Q. Zhang, Z. Wang, D. Wu, and S. Hu. Stop overvaluing multi-agent debate – we must rethink evaluation and embrace model heterogeneity, 2025.

[36] H. Zhang, Z. Cui, X. Wang, Q. Zhang, Z. Wang, D. Wu, and S. Hu. If multi-agent debate is the answer, what is the question? *arXiv preprint arXiv:2502.08788*, 2025.

[37] T. Zhang, H. Shi, Y. Wang, H. Wang, X. He, Z. Li, H. Chen, L. Han, K. Xu, H. Zhang, D. N. Metaxas, and H. Wang. TokUR: Token-level uncertainty estimation for large language model reasoning. In *The Fourteenth International Conference on Learning Representations*, 2026.

[38] L. Zheng, W.-L. Chiang, Y. Sheng, S. Zhuang, Z. Wu, Y. Zhuang, Z. Lin, Z. Li, D. Li, E. P. Xing, H. Zhang, J. E. Gonzalez, and I. Stoica. Judging llm-as-a-judge with mt-bench and chatbot arena, 2023.

[39] Y. Zhou, A. Xu, P. Wang, C. Xiong, and S. Joty. Evaluating judges as evaluators: The jetts benchmark of llm-as-judges as test-time scaling evaluators, 2025.

A Additional Details

A.1 Implementation Details for Fig. 2

Panel (a) uses four-agent, five-round `standard` Vanilla Debate and scores every saved round with the same `vote_then_judge_tiebreak` readout. For Qwen2.5-3B-Instruct, the curve averages over AMC12, MATH500, GPQA, and MMLU-Redux. For Qwen3-4B-Instruct, the curve uses the matched MMLU-Redux and GPQA diagnostics. The plotted value is the per-round readout accuracy minus the R0 majority-vote accuracy, in percentage points.

Panel (b) reports the unweighted average over AMC12, MATH500, GPQA, and MMLU-Redux for the label-free thicket society. The Vote bar is R0 majority-vote accuracy, the Debate readout bar is the final five-round debate accuracy, and the Proposal-hit bar is the R0 proposal-hit@4 rate for the same selected society.

Panel (c) uses the matched Qwen3-4B-Instruct five-round cells for sample, answer-coverage, support-coverage, and top-accuracy societies across the same five benchmarks. The x-axis is $P_0 - V_0$, where P_0 is the R0 proposal-hit rate and V_0 is the R0 majority-vote accuracy. The y-axis is $F - V_0$, where F is the final debate readout accuracy. The reported correlations are Pearson $r = 0.77$ and Spearman $\rho = 0.82$ across the 20 cells.

A.2 Evaluation Details

The main runs use data-specific generation budgets: 8192 tokens for AMC12 and MATH500, and 4096 tokens for GPQA and MMLU-Redux. We keep decoding fixed within each matched comparison. Self-consistency and Vanilla Debate sample the base model with temperature 0.7 and top-p 0.95. Thicket agents and the tie-break judge use deterministic decoding with temperature 0 and top-p 1.0. All runs use global seed 42; agent-level sampling seeds are deterministic offsets from that seed.

The main experiments use the verification debate prompt. Each debate-round prompt shows an agent its own previous response and the clipped previous-round responses from the other agents, asks it to

verify the peer answers, and then asks for a new final answer in the dataset's required format. The standard prompt-control ablation instead asks agents to compare peer solutions and revise if needed. The tie-break judge prompt sees the final candidate responses and is used only when the final vote has no unique winner.

A.3 Compute Resources

All experiments are inference-only: we do not train or fine-tune either backbone. We use BF16 inference with vLLM on an internal cluster. The reported runs used NVIDIA A100 GPUs with 80 GB memory and NVIDIA RTX PRO 6000 Blackwell Server Edition GPUs with 96 GB memory. Each downstream evaluation or diagnostic uses one GPU, while seed profiling distributes independent perturbed models over 6–8 GPUs. Table 4 reports observed runtime from successful-run logs; GPU-hours are the number of allocated GPUs multiplied by wall-clock time.

Table 4: Compute used by the reported experiment families. Profiling times vary with dataset size and output length; downstream times vary with the backbone, dataset, and number of debate rounds.

Experiment family	Workers per run	Observed wall time and compute
Qwen seed profiling (500 agents per dataset)	6–8 GPUs	3.3–18.4 hours per dataset; approximately 27–110 GPU-hours
OLMo seed profiling (300 retained agents per dataset)	8 GPUs	1.3–6.6 hours per dataset; about 10–53 GPU-hours
Five-round downstream and prompt-control evaluation	1 GPU per job	0.2–4.9 hours per job
Five-run stochastic robustness experiment	1 GPU per job; 120 jobs	156.8 GPU-hours in total
Controlled mechanism, readout, and transfer diagnostics	1 GPU per job	Within the same 0.2–4.9-hour single-job envelope

Summing successful-run wall times and GPU allocations gives an estimated **0.9k GPU-hours** for the experiments reported in this paper, with a plausible range of 0.8–1.0k GPU-hours. The range reflects early Qwen profiling jobs that were paused and resumed before scheduler-level accounting was retained; the 156.8 GPU-hours for the stochastic robustness experiment are measured directly from complete logs. Queue time and CPU-only analysis are not included. Retained run artifacts occupy approximately 12 GB, excluding cached model weights. The broader research project used additional compute for preliminary sweeps, failed or resumed jobs, and experiments on backbones not reported here; that exploratory compute is not included in this estimate.

A.4 Existing-Asset Licenses

Table 5 records the versions and terms of the model and dataset assets used in the paper. We use these assets only for research evaluation and do not redistribute model weights or raw benchmark questions.

Table 6: Five-run stochastic robustness under a common decoding configuration. Panel (a) reports mean ± SD across five generation seeds; bold marks the highest mean within each backbone and column, not statistical significance. Panel (b) reports paired comparisons of the five matched run-level eight-column averages. Effects and confidence intervals are in percentage points. Comparisons near the nominal $p = .05$ threshold should be interpreted cautiously.

(a) Per-benchmark accuracy and eight-column average.

Backbone	Method	AMC12	MATH500	GPQA	Chem.	CS	Med.	Moral	Acct.	8-col. Avg
Qwen3-4B	Vanilla Debate	**92.56 ± 0.21**	91.40 ± 0.55	50.40 ± 1.01	**71.00 ± 3.65**	81.00 ± 1.49	83.33 ± 0.00	67.67 ± 1.49	76.33 ± 1.39	76.71 ± 1.10
	SC-Greedy	**92.56 ± 0.17**	**92.27 ± 0.43**	51.62 ± 0.48	70.33 ± 3.80	84.00 ± 0.91	**86.00 ± 0.91**	69.67 ± 1.39	76.33 ± 2.17	77.85 ± 1.19
	AC-Greedy	92.21 ± 0.23	91.93 ± 0.49	**52.20 ± 0.90**	70.00 ± 1.18	**84.33 ± 2.24**	84.67 ± 1.39	**73.00 ± 1.39**	**77.00 ± 1.39**	**78.17 ± 1.08**
OLMo3-7B	Vanilla Debate	**86.82 ± 0.68**	85.80 ± 1.02	35.90 ± 0.24	60.67 ± 2.24	71.67 ± 4.08	72.67 ± 0.91	28.67 ± 3.80	57.00 ± 1.83	62.40 ± 1.72
	SC-Greedy	86.60 ± 0.59	85.60 ± 0.86	**36.99 ± 1.78**	59.00 ± 1.90	74.33 ± 2.24	74.33 ± 0.91	32.67 ± 4.35	57.33 ± 1.90	63.36 ± 1.69
	AC-Greedy	86.39 ± 0.89	**86.13 ± 0.38**	36.71 ± 0.87	**62.33 ± 1.49**	**77.33 ± 1.90**	**75.00 ± 1.18**	**33.00 ± 3.61**	**58.33 ± 2.04**	**64.40 ± 1.39**

(b) Paired comparisons of run-level averages.

Backbone	Comparison	Δ Avg	Paired 95% CI	p-value
Qwen3-4B	SC-Greedy − Vanilla Debate	$+1.135$	$[0.599, 1.671]$	0.0042
	AC-Greedy − Vanilla Debate	$+1.455$	$[0.005, 2.905]$	0.0495
	AC-Greedy − SC-Greedy	$+0.320$	$[-1.225, 1.865]$	0.5961
OLMo3-7B	SC-Greedy − Vanilla Debate	$+0.960$	$[0.022, 1.898]$	0.0468
	AC-Greedy − Vanilla Debate	$+2.005$	$[1.320, 2.691]$	0.0012
	AC-Greedy − SC-Greedy	$+1.045$	$[0.209, 1.881]$	0.0256

Table 5: Licenses and stated terms for existing model and dataset assets.

Asset and version	License or terms	Use in this work
Qwen3-4B-Instruct-2507	Apache-2.0	Main 4B backbone; research evaluation only
OLMo-3-7B-Instruct	Apache-2.0; Ai2 Responsible Use Guidelines	Main 7B backbone; research evaluation only
Qwen2.5-3B-Instruct	Qwen Research License	Motivating experiment; non-commercial research evaluation
GPQA	CC BY 4.0	Gated benchmark; examples are not redistributed
MMLU-Redux 2.0	CC BY 4.0	Benchmark evaluation
OlympiadBench	Apache-2.0	Benchmark evaluation
MATH-500	Mirror declares no license; source MATH and PRM800K repositories are MIT	We use the published split without redistributing it
AMC12-full	No open license declared; original problems are copyright MAA	Dataset card states research/educational use; questions are not redistributed

A.5 Five-Run Stochastic Robustness

We repeat the three main-table methods involved in the coverage comparison— Vanilla Debate, SC-Greedy, and AC-Greedy—for five matched generation runs on both backbones and all eight benchmark columns. The original full test sets, model, society, prompts, and finalizer remain fixed. For this robustness experiment only, all three methods use temperature 0.7 and top-$p = 0.95$; the generation seed varies across runs and is shared across methods within each run. This common stochastic configuration differs from the submitted configuration in Table 1, where thicket agents use deterministic decoding and all methods use one fixed global seed.

Table 6 reports mean ± standard deviation across the five runs. We compute the eight-column average within each run before summarizing across runs. Paired 95% confidence intervals and two-sided p-values use paired t-tests on these five run-level averages. Thus, the intervals measure sensitivity to generation randomness under stochastic decoding; they do not include uncertainty from resampling test questions.

Both coverage-selected methods have higher aggregate accuracy than Vanilla Debate on both backbones in these matched runs. All four comparisons against Vanilla Debate have nominal $p < .05$, although the Qwen AC-Greedy interval and the OLMo SC-Greedy interval lie close to zero. AC-Greedy and SC-Greedy are not distinguishable on Qwen3-4B, while AC-Greedy has the higher run-level average on OLMo3-7B.

Table 7: Seed-quality diagnostics for the main-table backbones. Values are percentages on the profiling split. Base ref. is the same-split unperturbed pass@1 reference. The Used column marks scales retained for downstream construction.

Backbone	Profiling split	σ	Seeds	Base ref.	Seeds $>$ base	Mean	Best	Used
Qwen3-4B	MATH500 first200	0.0005	160	90.00	59/160	89.82	92.50	yes
Qwen3-4B	MATH500 first200	0.001	174	90.00	51/174	89.58	92.50	yes
Qwen3-4B	MATH500 first200	0.002	166	90.00	11/166	88.10	91.00	yes
Qwen3-4B	AMC12 train360	0.0005	160	84.17	5/160	82.40	85.83	yes
Qwen3-4B	AMC12 train360	0.001	174	84.17	7/174	81.96	86.11	yes
Qwen3-4B	AMC12 train360	0.002	166	84.17	0/166	78.97	83.33	yes
Qwen3-4B	GPQA train200	0.0005	160	48.00	79/160	48.13	53.00	yes
Qwen3-4B	GPQA train200	0.001	174	48.00	69/174	47.73	54.50	yes
Qwen3-4B	GPQA train200	0.002	166	48.00	13/166	44.64	52.00	yes
Qwen3-4B	MMLU-Redux train200	0.0005	160	71.00	32/160	70.00	74.00	yes
Qwen3-4B	MMLU-Redux train200	0.001	174	71.00	23/174	68.92	76.50	yes
Qwen3-4B	MMLU-Redux train200	0.002	166	71.00	28/166	67.47	75.00	yes
OLMo-3-7B	MATH500 random200	0.0005	100	78.50	91/100	80.22	82.50	yes
OLMo-3-7B	MATH500 random200	0.001	100	78.50	87/100	80.31	83.50	yes
OLMo-3-7B	MATH500 random200	0.002	100	78.50	83/100	80.07	83.00	yes
OLMo-3-7B	AMC12 train360	0.0005	100	75.56	6/100	72.89	76.67	yes
OLMo-3-7B	AMC12 train360	0.001	100	75.56	6/100	72.90	76.67	yes
OLMo-3-7B	AMC12 train360	0.002	100	75.56	0/100	71.01	75.56	yes
OLMo-3-7B	GPQA train200	0.0005	100	33.00	9/100	30.26	36.00	yes
OLMo-3-7B	GPQA train200	0.001	100	33.00	17/100	30.71	36.50	yes
OLMo-3-7B	GPQA train200	0.002	100	33.00	13/100	30.46	37.00	yes
OLMo-3-7B	MMLU-Redux train200	0.0005	100	50.00	4/100	48.09	52.00	yes
OLMo-3-7B	MMLU-Redux train200	0.001	100	50.00	8/100	47.92	52.50	yes
OLMo-3-7B	MMLU-Redux train200	0.002	100	50.00	9/100	47.16	53.00	yes

A.6 Dataset and Split Details

The main call-aligned table evaluates AMC12, MATH500, GPQA, and MMLU-Redux. AMC12 uses 360 training questions for construction and 739 held-out test questions. MATH500 uses 200 training questions and 300 held-out test questions. GPQA uses 200 training questions and 346 held-out test questions. MMLU-Redux uses five subjects, with 40 training questions and 60 held-out test questions per subject, for 300 test questions in total. The five-round dynamics in the diagnostic appendix also include OlympiadBench, with 200 training questions and 474 held-out test questions.

A.7 Seed Profiling Details

Table 7 records the perturbation scales used to construct the main-table thicket pools. Qwen3-4B-Instruct uses a 500-seed legacy pool. A fixed global random seed sampled both the perturbation seeds and their scales, so the realized counts across the three retained scales are 160, 174, and 166. OLMo samples 100 seeds at each listed scale. Its diagnostic sweeps also included smaller and larger scales, but the final downstream experiments use the middle three scales: 5×10^{-4}, 10^{-3}, and 2×10^{-3}. For top-K thicket baselines, the scalar score is training accuracy:

$$v_m = \frac{1}{|\mathcal{D}_{\text{train}}|} \sum_{(x,y^\star)\in\mathcal{D}_{\text{train}}} \mathbf{1}\{\text{ans}(e_m, x) = y^\star\}, \tag{17}$$

where $\text{ans}(e_m, x)$ is the normalized answer from agent e_m.

A.8 Prompt Templates

Initial Answer Prompt. For MATH500, the runs use the following system instruction and append the same answer-format requirement to the user question:

MATH500 Initial Prompt

System message. Let's think step by step. End with exactly one final answer in the form `\boxed{your_answer}` and nothing else after the boxed answer.

User message. *[Problem text]*

Let's think step by step. End with exactly one final answer in the form `\boxed{your_answer}` and nothing else after the boxed answer.

For AMC12, the answer space is A–E:

AMC12 Initial Prompt

System message. Let's think step by step. End with exactly one final answer tag in the form `<answer>X</answer>`, where X is A, B, C, D, or E. Do not write anything after the final answer tag.

User message. *[Problem text]*
Return the single best answer as `<answer>X</answer>` where X is one of A, B, C, D, E.

For GPQA and MMLU-Redux, the answer space is A–D:

GPQA and MMLU-Redux Initial Prompt

System message. Let's think step by step. End with exactly one final answer tag in the form `<answer>X</answer>`, where X is A, B, C, or D. Do not write anything after the final answer tag.

User message. *[Problem text]*
Think briefly and solve the multiple-choice question. On the final line, write exactly one answer in the form `\boxed{X}`, where X is one of A, B, C, D. Do not write anything after the boxed answer.

For multiple-choice datasets, the extractor maps answer tags, boxed letters, parenthesized letters, and trailing single-letter answers to the same option key.

Verification Debate Prompt. The verification prompt is appended after the agent's previous response. It includes clipped peer responses and then asks:

Verification Debate Prompt

Other assistants answered the same question. Verify your previous answer against the peer answers below. First write a short verification note that identifies the single most important agreement or disagreement that affects correctness. Then, on the final line only, give your final answer in the same required format.

This is the main debate prompt used in the experiments.

Standard Debate Prompt. The standard prompt-control ablation replaces the verification instruction with:

Standard Debate Prompt

Other assistants answered the same question. Compare their solutions with yours. Revise your answer if needed. Return only your final solution in the same required format.

Tie-Break Judge Prompt. The judge receives the final candidate responses only for tied final votes. In the verification-style runs, it is asked:

Tie-Break Judge Prompt

Review the candidate solutions below and verify which answer is best supported. First write a short verification note explaining the decisive evidence. Then, on the final line only, give the final answer in the same required format.

A.9 Controlled Readout Probe Details

For the answer-stripped rationale intervention, let T, F, and N denote the correct-rationale, wrong-rationale, and neutral contexts. We measure the gold answer against all wrong-answer probability mass,

$$m_a(x) = \log p(y_g \mid x) - \log \sum_{y \neq y_g} p(y \mid x).$$

The correct-over-wrong gap is $m_a(T) - m_a(F)$. We also report the help-over-harm asymmetry

$$A = [m_a(T) - m_a(N)] - [m_a(N) - m_a(F)],$$

which tests whether the helpful shift from correct reasoning exceeds the reverse harmful shift from wrong reasoning.

Table 8: Original peer-response effects and equivalent conclusion-only support over recovery cases. Trace effects are means with 95% confidence intervals; $\widehat{W}$ is the median interpolated equivalent support. The final column reports the correct-support minus wrong-support answer margin under the Verify prompt at each dose.

Backbone	Dataset	Δm_{trace} [95% CI]	Median $\widehat{W}$	Correct−wrong margin, $d = 1/2/3$
Qwen3-4B	GPQA	8.15 [4.96, 11.41]	2.23	2.59 / 5.94 / 13.63
Qwen3-4B	MMLU-Redux	11.40 [9.02, 13.97]	> 3	1.03 / 5.36 / 11.76
OLMo3-7B	GPQA	4.57 [3.65, 5.50]	1.92	3.33 / 5.06 / 5.79
OLMo3-7B	MMLU-Redux	5.47 [4.52, 6.43]	> 3	3.59 / 5.18 / 5.53

Table 9: Truth sensitivity of answer-stripped trace rationales. Entries are mean margin changes with 95% confidence intervals, shown as Standard / Verify.

Backbone / dataset	Help-over-harm asymmetry A	Correct-over-wrong gap $T - F$
Qwen3-4B / GPQA	7.34 [2.14, 12.45] / 6.84 [1.29, 12.33]	1.90 [.54, 3.39] / 4.59 [2.45, 6.96]
Qwen3-4B / MMLU-Redux	8.70 [3.50, 13.72] / 6.36 [2.34, 10.42]	5.98 [3.78, 8.32] / 5.50 [3.46, 7.69]
OLMo3-7B / GPQA	1.92 [.46, 3.35] / 1.39 [.17, 2.58]	1.70 [1.16, 2.31] / 2.01 [1.35, 2.77]
OLMo3-7B / MMLU-Redux	2.84 [.87, 4.85] / 2.24 [.78, 3.78]	1.93 [1.26, 2.63] / 2.10 [1.41, 2.83]

Table 10: Free-generation accuracy after three peer messages. Each entry evaluates the same 50 questions. Correct support is bolded.

Backbone	Dataset	Prompt	Neutral	Correct support	Answer withheld	Shuffled rationale
Qwen3-4B	GPQA	Standard	.32	**.50**	.32	.28
Qwen3-4B	GPQA	Verify	.32	**.46**	.34	.14
Qwen3-4B	MMLU-Redux	Standard	.28	**.58**	.26	.28
Qwen3-4B	MMLU-Redux	Verify	.38	**.42**	.32	.26
OLMo3-7B	GPQA	Standard	.30	**.80**	.30	.30
OLMo3-7B	GPQA	Verify	.30	**.86**	.30	.30
OLMo3-7B	MMLU-Redux	Standard	.30	**.80**	.30	.28
OLMo3-7B	MMLU-Redux	Verify	.30	**.88**	.30	.20
Average			.313	**.663**	.305	.255

Table 11: Round-level prompt-style dynamics averaged over the 15 matched dataset-method cells. R0 is identical because both controls use the same seed answers before debate. Δ is Standard minus Verify. All values are percentages.

Round	Verify Vote	Standard Vote	Δ Vote	Verify Hit	Standard Hit	Δ Hit
R0	76.04	76.04	+0.00	83.95	83.95	+0.00
R1	77.95	77.81	-0.14	85.38	84.89	-0.49
R2	78.19	78.12	-0.08	82.20	83.38	+1.18
R3	78.37	78.16	-0.21	80.44	82.98	+2.54
R4	78.10	78.43	+0.32	79.56	82.04	+2.48
R5	78.21	78.13	-0.08	79.26	81.62	+2.37

Table 12: Prompt-style control for the five-round new-seed debate runs. Verify Final is the completed `verify_then_answer` debate prompt; Standard Final repeats the same datasets, seed banks, verifier settings, and `vote_then_judge_tiebreak` finalizer with the normal `standard` debate prompt. Δ is Standard Final minus Verify Final. All accuracy values are percentages.

Dataset	Method	N	Verify Final	Standard Final	Δ	Standard R5 Vote	Standard R5 Hit
AMC12	Vanilla Debate	739	92.96	92.42	-0.54	92.29	93.78
	AC-Greedy Debate	739	91.61	91.34	-0.27	91.75	94.18
	SC-Greedy Debate	739	91.88	92.02	+0.14	91.88	93.50
MATH500	Vanilla Debate	300	91.33	92.00	+0.67	91.67	93.00
	AC-Greedy Debate	300	91.67	92.67	+1.00	93.33	94.00
	SC-Greedy Debate	300	92.00	91.67	-0.33	91.67	93.33
OlympiadBench	Vanilla Debate	474	81.22	83.76	+2.53	83.97	85.86
	AC-Greedy Debate	474	81.43	81.86	+0.42	81.65	84.39
	SC-Greedy Debate	474	80.80	81.22	+0.42	81.43	83.33
GPQA	Vanilla Debate	346	51.73	52.31	+0.58	51.45	60.98
	AC-Greedy Debate	346	52.02	49.71	-2.31	50.58	60.98
	SC-Greedy Debate	346	50.58	50.58	+0.00	50.29	58.96
MMLU-Redux	Vanilla Debate	300	72.33	70.00	-2.33	70.67	73.33
	AC-Greedy Debate	300	75.67	74.33	-1.33	74.33	76.00
	SC-Greedy Debate	300	75.33	75.67	+0.33	75.00	78.67
Average	Vanilla Debate	–	77.92	78.10	+0.18	78.01	81.39
Average	AC-Greedy Debate	–	78.48	77.98	-0.50	78.33	81.91
Average	SC-Greedy Debate	–	78.12	78.23	+0.11	78.05	81.56
Overall	All	–	78.17	78.10	-0.07	78.13	81.62

Prompt-Control Tables. Tables 11 and 12 report the completed control that replaces the verification debate prompt with the normal `standard` debate prompt. The control uses the same five datasets, seed banks, verifier settings, debate rounds, saved traces, and finalizer as the main five-round runs. The standard prompt is slightly slower in the mean vote trajectory: it trails `verify_then_answer` in R1–R3, reaches its mean vote peak one round later, and ends essentially tied at R5. It preserves more proposal-hit coverage after R2, but the finalizer does not convert that extra coverage into higher final accuracy.

These tables provide a secondary prompt-style analysis for the controlled readout probes in Sec. 5. In the averaged dynamics, the standard prompt retains 2.37 points more R5 proposal-hit coverage than the verification prompt, while their final accuracies differ by only 0.07 points overall. The aggregate comparison therefore does not identify a uniform prompt advantage or directly estimate verification mass.

The effect is not uniform across datasets or selection methods. The standard prompt helps Vanilla Debate on MATH500, OlympiadBench, and GPQA, while the verification prompt is better for MMLU-Redux Vanilla Debate and for AC-Greedy Debate on GPQA and MMLU-Redux. We therefore use this control as a secondary configuration diagnostic rather than evidence that one prompt dominates the other. The controlled fixed-proposal interventions in Sec. 5 provide the direct test of answer-specific evidence use.

A.10 Dynamics and Conversion Controls

Immediate-finalizer control. A shrinking proposal-hit rate does not distinguish correct minorities that were recovered from those that were simply lost. We therefore apply the same `vote_then_judge_tiebreak` finalizer independently to the proposals saved at R0 through R5. This changes only the stage at which the finalizer is applied; no debate generation is rerun. The analysis covers four societies and five benchmarks, with equal weight assigned to each benchmark.

Table 13: Immediate-finalizer and interaction control. Panel (a) compares the R0 and R5 states by method. The final column is R5 Vote+Judge minus R0 Vote+Judge; confidence intervals resample questions within each benchmark before recomputing the unweighted macro average. Panel (b) applies the same finalizer at every saved stage and averages over all four societies and five benchmarks. All accuracy, hit, gain, tie, and unanimity values are percentages.

(a) Method-specific R0 and R5 comparison.

Method	Vote R0 → R5	Hit R0 → R5	R0 V+J	R0 judge	R5 V+J	Additional R5 gain [95% CI]	Tie R0 → R5	Unanimous R0 → R5
Vanilla Debate	75.65 → 77.99	83.03 → 78.98	76.03	+0.39	77.89	+1.85 [0.54, 3.18]	7.27 → 0.83	70.50 → 95.86
Top-4 Debate	76.25 → 78.16	84.11 → 79.27	76.84	+0.59	78.42	+1.58 [0.37, 2.79]	8.90 → 0.78	67.78 → 95.84
AC-Greedy Debate	76.84 → 78.41	84.93 → 79.60	77.59	+0.75	78.51	+0.92 [−0.39, 2.27]	9.58 → 0.91	64.61 → 95.45
SC-Greedy Debate	75.64 → 78.22	83.88 → 79.19	75.33	−0.31	78.08	+2.74 [1.40, 4.11]	9.56 → 0.63	63.07 → 96.11

(b) Full stage trajectory.

Stage	Vote	Hit	Headroom	Vote+Judge	Gain over R0 vote	Tie	Unanimous
R0	76.09	83.99	7.90	76.45	+0.36	8.83	66.49
R1	78.09	85.32	7.23	77.99	+1.89	6.74	72.13
R2	78.32	82.20	3.88	78.34	+2.25	3.64	86.14
R3	78.35	80.56	2.21	78.52	+2.42	1.96	91.71
R4	78.10	79.58	1.48	78.12	+2.02	1.45	94.30
R5	78.20	79.26	1.06	78.22	+2.13	0.79	95.81

At R0, the finalizer contributes only 0.36 points on average. The R2 and R3 states reach 78.34% and 78.52% after the same finalizer, respectively 1.89 and 2.07 points above the R0 Vote+Judge result. This rise occurs while ties fall from 8.83% to 3.64% at R2 and 1.96% at R3, and unanimity rises from 66.49% to 86.14% and 91.71%. The trajectory therefore shows rapid interaction gains followed by a plateau, not a claim that the final round is always optimal. At the method level, the additional R5 gain beyond the R0 readout ranges from 0.92 to 2.74 points; its interval excludes zero for Vanilla, Top-4, and SC-Greedy.

Backbone-dependent conversion. We next ask whether initial headroom predicts final lift on OLMo as it does on Qwen. For the pooled comparison, we use the four datasets and four society constructions available for both backbones. The MMLU analysis pools the five predefined Table 1 subjects, and its Pearson interval is a paired question-bootstrap interval.

Table 14: Association between initial headroom $H = P_0 - V_0$ and final lift $G = F - V_0$ on the two backbones. The MMLU row reports a 95% paired question-bootstrap interval for Pearson r.

Scope	Qwen3-4B	OLMo3-7B
Four shared datasets, pooled	$r = .770,\ \rho = .787$	$r = .548,\ \rho = .648$
Five MMLU subjects, pooled	$r = .568\ [.125, .787]$	$r = .635\ [.219, .805]$

The positive OLMo associations show that proposal supply remains informative, especially on MMLU. The failure mode is nevertheless dataset-dependent. On OLMo MATH500, all four societies retain 3.33–6.33 points of initial headroom, yet every final lift is negative, ranging from -1.67 to -0.33 points. This localizes the failure to conversion rather than missing proposals: useful answers are surfaced, but the backbone and readout do not reliably turn them into a better final decision.

A.11 Secondary Construction and Transfer Diagnostics

Transfer decomposition. The transfer experiment in Table 2 can be read through the exact gain decomposition in Proposition 1. Recovered, New correct, and Damage below are percentage-point contributions and satisfy Gain = Recovered + New correct − Damage.

Table 15: AMC12-to-MMLU-Redux transfer decomposition. Recovery rate is conditional on R0 headroom; the contribution columns are unconditional percentage points.

Backbone	Method	R0 vote	Headroom	Recovery rate	Recovered	New correct	Damage	Gain	Final
Qwen3-4B	SC-Greedy	72.33	11.67	45.71	5.33	0.67	2.00	+4.00	76.33
Qwen3-4B	AC-Greedy	75.33	8.33	48.00	4.00	0.33	3.00	+1.33	76.67
OLMo3-7B	SC-Greedy	55.33	14.67	36.36	5.33	0.67	3.00	+3.00	58.33
OLMo3-7B	AC-Greedy	54.00	15.33	26.09	4.00	0.33	2.00	+2.33	56.33

On Qwen, AC-Greedy begins with the higher vote and therefore retains the best final accuracy. SC-Greedy has more headroom, recovers 5.33 rather than 4.00 points, and incurs less damage, explaining its larger lift. On OLMo, the two methods have similar headroom, but SC-Greedy has the higher recovery rate and again recovers 5.33 rather than 4.00 points. Transfer therefore depends on both proposal supply and its subsequent conversion.

Complementary-coverage diagnostics. We use expertise in a behavioral, task-level sense: an agent is useful when it solves construction examples that its teammates miss and this complementarity improves coverage on unseen questions. Let C_1 and C_2 be the construction questions correctly solved by the first two selected agents. Lower $|C_1 \cap C_2|/|C_1 \cup C_2|$ means less redundant coverage.

Table 16: Complementarity and held-out coverage on Qwen3-4B MATH500. Panel (a) uses the 200-question construction split. Panel (b) freezes each four-agent society and evaluates cumulative R0 proposal-hit coverage on 300 held-out questions; agents are listed in selection order, and tied construction accuracies share the minimum rank.

(a) Correct-solution overlap of the first two agents.

Method	Intersection	Union	IoU ↓
Top-4 Debate	182	188	96.8%
AC-Greedy	177	190	93.2%
SC-Greedy	176	186	94.6%

(b) Team composition and held-out cumulative R0 coverage.

Method	Construction rank (accuracy), in team order	Held-out cumulative coverage	R5 final
Top-4 Debate	1 (92.5%), 1 (92.5%), 3 (92.0%), 3 (92.0%)	88.33 → 91.33 → 92.00 → 92.67	90.67
AC-Greedy	1 (92.5%), 25 (91.0%), 25 (91.0%), 8 (91.5%)	90.00 → 92.67 → 93.33 → 94.33	91.67
SC-Greedy	8 (91.5%), 181 (89.5%), 8 (91.5%), 25 (91.0%)	90.67 → 92.67 → 93.67 → 93.67	92.00

Both coverage objectives select less-overlapping first pairs than Top-4. When the four-agent societies are evaluated out of sample, AC-Greedy covers 283 of 300 questions, SC-Greedy covers 281, and Top-4 covers 278. The selected agent identities need not encode fixed semantic roles; the evidence supports the narrower claim of transferable, task-level complementary proposal behavior.

Pool- and construction-size sensitivity. We run 200 paired subsampling replications across four Qwen benchmarks, with four selected agents and 50 held-out questions per replication. Entries in Table 17 are equal-dataset macro held-out proposal-hit@4.

Table 17: Sensitivity to candidate-pool size N and construction-set size m. Each arrow follows the increasing values shown in the column header.

Selector	Pool $N = 50 \to 100 \to 200 \to 500$	Construction $m = 25 \to 50 \to 100 \to 150$
AC-Greedy	$84.06 \to 84.08 \to 84.29 \to 84.12$	$83.22 \to 83.59 \to 84.12 \to 84.38$
SC-Greedy	$83.95 \to 84.03 \to 84.01 \to 83.81$	$83.30 \to 83.36 \to 83.81 \to 83.85$

Coverage is effectively saturated over $N = 50$–500, while increasing m from 25 to 150 improves AC-Greedy by 1.16 points (95% CI $[0.87, 1.43]$) and SC-Greedy by 0.56 points ($[0.26, 0.86]$). The useful perturbation scale remains task- and backbone-dependent. We swept a broader range on construction data, retained $\{5 \times 10^{-4}, 10^{-3}, 2 \times 10^{-3}\}$ because it contained a useful concentration of high-quality seeds, and froze that band before test evaluation; the per-scale profiles are in Table 7.

Hidden-Prefix construction. To test whether construction requires labels or discrete answer identities, we remove final-answer tokens, keep the first 64 reasoning tokens, and record the ℓ_2-normalized final-layer state at the prefix endpoint. For each question, Hidden-Prefix clusters the 500-agent pool with k-means ($k = 4$, 20 restarts) in cosine geometry and selects four agents to maximize inverse-support representation-cluster coverage. Neither answer identity nor correctness enters selection.

Selection uses 200 construction questions and evaluation uses a disjoint 300-question MMLU-Redux test set. We fix one Hidden-Prefix society and one TopAcc society and evaluate each with 10 paired generation seeds, temperature 0.7, top-$p = 0.95$, four agents, five verification-prompt rounds, an 800-character peer context, and the same vote-then-judge finalizer.

Table 18: Fixed-society Qwen3-4B MMLU-Redux results. Hidden-Prefix and TopAcc report mean $\pm$ SD across 10 paired generation seeds; AC-Greedy and SC-Greedy are the submitted fixed runs and are included only as reference points.

Selector	Construction signal	Requires	R0 hit@4	R0 vote	R5 final
AC-Greedy	Correct-answer coverage	Labels	84.67	74.67	75.67
SC-Greedy	Answer-cluster coverage	Clusters	82.00	68.67	75.33
Hidden-Prefix	Representation-cluster coverage	Neither	84.30 ± 0.69	75.07 ± 1.03	77.00 ± 1.25
TopAcc	Individual accuracy	Labels	83.60 ± 0.62	73.93 ± 1.07	75.40 ± 0.80

Hidden-Prefix reaches 84.30 hit@4 without labels or answer identities, within 0.37 points of supervised AC-Greedy. Against its paired TopAcc control, it improves R0 vote by 1.13 points (95% CI $[0.20, 2.07]$) and R5 final accuracy by 1.60 points ($[0.32, 2.88]$); a two-way seed-and-question interval for the final gain is $[0.07, 3.10]$. This establishes an answer-string-free construction route on a multiple-choice task. It does not yet validate semantic equivalence classes or evaluation for open-ended generation.

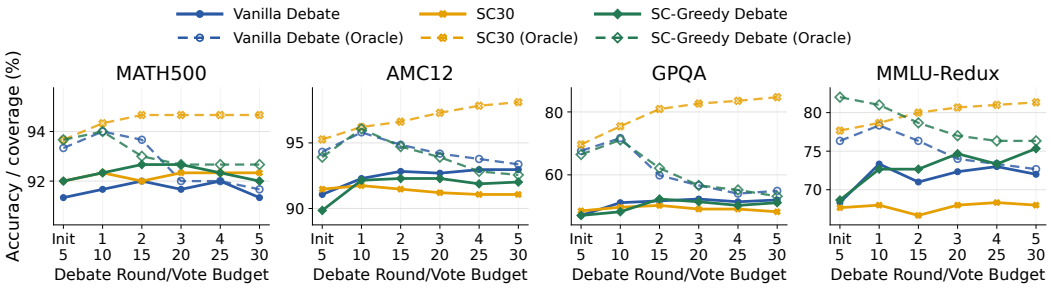

Figure 4: Label-free five-round dynamics. This appendix counterpart to Fig. 3 shows methods that do not use labeled training diagnostics for society selection. Solid lines show round-level majority-vote accuracy before finalization; dashed hollow legend entries marked "Oracle" show proposal-hit coverage.

A.12 Complete Round-Level Statistics

This appendix collects the full round-level tables behind Fig. 3 and Fig. 4. The tables expand the mechanism diagnostics used in Sec. 5 and Sec. 4. They ask two questions: whether the selected agents supply correct proposals that vote misses, and whether the matched readout converts that supply into final accuracy without excessive damage.

Metrics and Reading Guide. Vote is the majority-vote accuracy on the proposal multiset at a given round. Proposal Hit is the probability that at least one visible proposal at that round contains the correct answer. Headroom H is the gap between these two events: it counts cases where the proposal multiset contains the correct answer but majority vote misses it. Final is the end-of-run `vote_then_judge_tiebreak` accuracy. Gain G is Final minus Vote. Recovery R is the final accuracy on headroom instances. New Correctness I is the final accuracy on instances where the visible proposal multiset did not contain the correct answer and Vote was wrong. Damage L is the failure rate on instances where Vote is already correct. Thus Proposal Hit and Headroom diagnose proposal supply, while Recovery, New Correctness, and Damage diagnose readout conversion.

These definitions make the appendix a supply-readout analysis rather than a second main-results table. Higher Proposal Hit or Headroom means that the selected agents surface more correct proposals. Higher Recovery means that the readout converts more surfaced correct minorities into final answers. Lower Damage means that the readout preserves cases already solved by majority vote. New Correctness tracks gains not explained by the proposal-hit supply at the chosen round. Final accuracy combines these effects, so it can hide whether a gain comes from better proposal supply, better readout conversion, or lower damage.

The main pattern is that useful proposal supply appears early and then contracts as debate moves toward consensus. At R0, all four debate societies have 7–8 points of recoverable headroom on average. By R5, the remaining headroom falls to about one point. This supports the fixed-initial-proposal view in Sec. 3: the society often surfaces correct minorities before interaction, and later rounds can reduce the set of alternatives even when vote accuracy improves. Debate gains therefore depend on recovering useful minority answers before proposal-hit coverage collapses.

The recovery and damage columns show why higher proposal-hit supply is not enough by itself. For example, AC-Greedy Debate has the highest average R0 proposal-hit rate, 84.93%, but its final lift over the R0 vote is 1.64 points. SC-Greedy Debate has lower average R0 proposal-hit rate, 83.88%, but larger average lift, 2.48 points, because its starting vote is lower and more of its gain comes from converting missed proposals. These differences are why the main text reports headroom, recovery, and damage rather than final accuracy alone.

A.12.1 Average Recovery Diagnostics

Interpretation. Table 19 separates proposal supply from readout conversion. Initial headroom is substantial for every method, which means the selected societies surface correct answers that majority vote does not extract. However, recovery falls sharply after the first few rounds because

Table 19: Round-level recovery diagnostics averaged over the five benchmarks. For each round, Vote and Hit are the round-specific majority-vote and proposal-hit accuracies, while Final is the end-of-run `vote_then_judge_tiebreak` accuracy. We define $H = P - V$, $G = D - V$, $\rho = G/H$, $R = \Pr(D = 1 \mid P = 1, V = 0)$, $I = \Pr(D = 1 \mid P = 0, V = 0)$, and $L = \Pr(D = 0 \mid V = 1)$, where D is the final decision. All values except ρ are percentages; averages are unweighted over benchmarks.

Method	Round	Vote	Hit	Final	H	G	ρ	R	I	L
Vanilla Debate	R0	75.65	83.03	77.92	7.39	2.27	0.31	53.49	9.94	5.00
	R1	77.85	85.05	77.92	7.20	0.07	0.01	43.10	1.23	4.25
	R2	77.84	81.86	77.92	4.02	0.08	0.02	24.33	0.14	1.58
	R3	78.05	80.07	77.92	2.03	-0.13	-0.06	43.39	0.60	1.37
	R4	78.13	79.04	77.92	0.92	-0.21	-0.23	38.89	0.13	1.08
	R5	77.99	78.98	77.92	0.99	-0.08	-0.08	10.00	0.13	0.33
Top-4 Debate	R0	76.25	84.11	78.28	7.87	2.03	0.26	40.88	7.26	4.53
	R1	78.51	85.15	78.28	6.64	-0.23	-0.03	29.46	0.38	3.67
	R2	78.72	82.22	78.28	3.51	-0.44	-0.13	28.94	0.48	2.16
	R3	78.30	80.93	78.28	2.63	-0.02	-0.01	30.09	0.00	1.18
	R4	78.11	79.66	78.28	1.56	0.17	0.11	42.93	0.00	0.73
	R5	78.16	79.27	78.28	1.11	0.11	0.10	17.78	0.00	0.25
AC-Greedy Debate	R0	76.84	84.93	78.48	8.09	1.64	0.20	48.94	7.70	6.01
	R1	78.63	85.72	78.48	7.09	-0.15	-0.02	34.76	1.68	4.23
	R2	78.43	82.49	78.48	4.06	0.06	0.01	45.25	0.40	2.46
	R3	78.73	80.84	78.48	2.10	-0.25	-0.12	37.24	0.00	1.58
	R4	78.50	79.97	78.48	1.47	-0.02	-0.01	39.44	0.00	0.64
	R5	78.41	79.60	78.48	1.19	0.07	0.06	32.00	0.00	0.11
SC-Greedy Debate	R0	75.64	83.88	78.12	8.23	2.48	0.30	48.52	7.20	5.34
	R1	77.37	85.35	78.12	7.98	0.75	0.09	34.76	1.45	3.55
	R2	78.32	82.25	78.12	3.93	-0.20	-0.05	25.29	0.00	2.58
	R3	78.33	80.42	78.12	2.09	-0.21	-0.10	29.56	0.00	1.37
	R4	77.69	79.65	78.12	1.97	0.43	0.22	24.71	0.00	0.42
	R5	78.22	79.19	78.12	0.97	-0.10	-0.10	6.67	0.00	0.26

debate compresses the proposal set. Damage also falls over rounds, since fewer vote-correct cases are changed once the agents converge. New Correctness is largest at R0 and becomes small after the debate trace has mostly converged, which means most final gains are explained by surfaced proposal supply rather than late invention of new correct answers. The tradeoff is therefore not monotone: later rounds can make the majority more stable, but they can also remove the minority alternatives that the readout needs to recover.

A.12.2 Five-Round New-Seed Debate Results

Interpretation. Table 20 shows that the same mechanism appears at the dataset level. On GPQA and MMLU-Redux, initial proposal-hit rates are much higher than initial vote accuracy, and debate produces the largest lifts. On MATH500 and AMC12, the initial vote is already high, so the available room for final improvement is smaller even when proposal-hit coverage is high. This is consistent with Proposition 1: final gain is largest when there is recoverable headroom and the readout converts it without large damage.

The construction methods also behave differently across datasets. AC-Greedy Debate gives the best average final accuracy, but SC-Greedy Debate gives the largest average lift from its own initial vote. This supports the role of label-free coverage as a practical supply constructor, while also showing why the supervised construction remains a useful reference.

Table 20: Five-round debate results for the new seed teams. Final is accuracy after `vote_then_judge_tiebreak`; Lift is Final minus Init Vote. Bold marks the best final accuracy within each dataset. Average rows report the unweighted mean across the five benchmarks.

Dataset	Method	N	Init Vote	R5 Vote	Final	Init Hit	R5 Hit	Lift	Judge Ties
MATH500	Vanilla Debate	300	91.33	91.33	91.33	93.33	91.67	+0.00	1
MATH500	AC-Greedy Debate	300	91.33	91.33	91.67	94.33	91.67	+0.33	2
MATH500	SC-Greedy Debate	300	92.00	92.00	**92.00**	93.67	92.67	+0.00	0
OlympiadBench	Vanilla Debate	474	80.38	81.65	81.22	83.54	82.28	+0.84	7
OlympiadBench	AC-Greedy Debate	474	79.11	81.43	**81.43**	82.91	81.86	+2.32	7
OlympiadBench	SC-Greedy Debate	474	80.59	80.59	80.80	83.33	81.22	+0.21	5
AMC12	Vanilla Debate	739	91.07	92.96	**92.96**	94.32	93.37	+1.89	4
AMC12	AC-Greedy Debate	739	90.53	91.88	91.61	95.13	92.83	+1.08	9
AMC12	SC-Greedy Debate	739	89.85	92.02	91.88	93.91	92.56	+2.03	2
GPQA	Vanilla Debate	346	47.11	52.02	51.73	67.63	54.91	+4.62	4
GPQA	AC-Greedy Debate	346	48.55	51.73	**52.02**	67.63	54.62	+3.47	3
GPQA	SC-Greedy Debate	346	47.11	51.16	50.58	66.47	53.18	+3.47	4
MMLU-Redux	Vanilla Debate	300	68.33	72.00	72.33	76.33	72.67	+4.00	2
MMLU-Redux	AC-Greedy Debate	300	74.67	75.67	**75.67**	84.67	77.00	+1.00	1
MMLU-Redux	SC-Greedy Debate	300	68.67	75.33	75.33	82.00	76.33	+6.67	2
Average	Vanilla Debate	–	75.65	77.99	77.92	83.03	78.98	+2.27	3.6
Average	AC-Greedy Debate	–	76.84	78.41	**78.48**	84.93	79.60	+1.64	4.4
Average	SC-Greedy Debate	–	75.64	78.22	78.12	83.88	79.19	+2.48	2.6

B Proofs for Sec. 3

B.1 Proof of Proposition 1

Proof of Proposition 1. By definition,

$$\mathcal{G}_\eta(S) = \mathbb{E}[D_\eta - V] = \Pr(D_\eta = 1, V = 0) - \Pr(D_\eta = 0, V = 1).$$

We decompose the positive term by whether the initial proposal supply hit the correct answer:

$$\Pr(D_\eta = 1, V = 0) = \Pr(D_\eta = 1, P = 1, V = 0) + \Pr(D_\eta = 1, P = 0, V = 0).$$

Using the definitions of R_η and I_η gives

$$\Pr(D_\eta = 1, P = 1, V = 0) = \Pr(P = 1, V = 0)\Pr(D_\eta = 1 \mid P = 1, V = 0) = \mathcal{H}(S)R_\eta(S),$$

and

$$\Pr(D_\eta = 1, P = 0, V = 0) = \Pr(P = 0, V = 0)I_\eta(S).$$

The negative term is

$$\Pr(D_\eta = 0, V = 1) = \Pr(V = 1)\Pr(D_\eta = 0 \mid V = 1) = \Pr(V = 1)L_\eta(S).$$

Substituting these three identities proves Eqn. 7. □

B.2 Proof of Lemma 1

Lemma 1 (Verification-Weighted Dirichlet Conjugacy)**.** *Let $\pi = (\pi^{(k)})_{k\in\mathcal{Y}_t}$ be an answer-probability vector over $\mathcal{Y}_t$, and let $\pi \mid x \sim \mathrm{Dirichlet}(\alpha_0)$ with $\alpha_0^{(k)} = \kappa q_\theta^0(k \mid x)$. Conditional on an effective verified pseudo-count vector $W_{\eta,t}$, use the verification-weighted pseudo-likelihood*

$$p(\mathcal{F}_t, W_{\eta,t} \mid \pi) \propto \prod_{k\in\mathcal{Y}_t} \left(\pi^{(k)}\right)^{C_t^{(k)} + W_{\eta,t}^{(k)}}.$$

Then the posterior remains Dirichlet, and its posterior predictive is Eqn. 9.

Proof of Lemma 1. The Dirichlet prior has density

$$p(\pi \mid x) \propto \prod_{k\in\mathcal{Y}_t} \left(\pi^{(k)}\right)^{\alpha_0^{(k)} - 1}.$$

Multiplying by the verification-weighted pseudo-likelihood gives

$$p(\pi \mid x, \mathcal{F}_t, W_{\eta,t}) \propto p(\pi \mid x)\, p(\mathcal{F}_t, W_{\eta,t} \mid \pi) \propto \prod_{k \in \mathcal{Y}_t} \left(\pi^{(k)}\right)^{\alpha_0^{(k)} - 1 + C_t^{(k)} + W_{\eta,t}^{(k)}}.$$

This is the kernel of a Dirichlet distribution with parameters

$$\alpha_0^{(k)} + C_t^{(k)} + W_{\eta,t}^{(k)} = \kappa q_\theta^0(k \mid x) + C_t^{(k)} + W_{\eta,t}^{(k)}.$$

The posterior predictive for a categorical draw from a Dirichlet posterior is the posterior mean:

$$\mathbb{E}[\pi^{(k)} \mid x, \mathcal{F}_t, W_{\eta,t}] = \frac{\kappa q_\theta^0(k|x) + C_t^{(k)} + W_{\eta,t}^{(k)}}{\sum_{j \in \mathcal{Y}_t} \left(\kappa q_\theta^0(j|x) + C_t^{(j)} + W_{\eta,t}^{(j)}\right)}.$$

Since $\sum_j q_\theta^0(j \mid x) = 1$, $\sum_j C_t^{(j)} = C_t^+$, and $\sum_j W_{\eta,t}^{(j)} = W_{\eta,t}^+$, this is exactly Eqn. 9. □

B.3 Proof of Lemma 2

Lemma 2 (LVD Marginalization)**.** *Let*

$$q_\eta^v(z \mid x, \mathcal{F}_t) = \mathrm{softmax}_{z \in \mathcal{Y}_t}\big(a_\eta(z; x, \mathcal{F}_t)\big) \tag{18}$$

be a soft allocation of verification support. If a latent verification draw $v = z$ adds mass $\tau_{\eta,t} \geq 0$ to answer z, define

$$p_\eta(y = k \mid v = z, x, \mathcal{F}_t) = \frac{\kappa q_\theta^0(k|x) + C_t^{(k)} + \tau_{\eta,t} \mathbf{1}\{z=k\}}{\kappa + C_t^+ + \tau_{\eta,t}}. \tag{19}$$

If $v \sim q_\eta^v(\cdot \mid x, \mathcal{F}_t)$, marginalizing v gives Eqn. 9 with $W_{\eta,t}^{(k)} = \tau_{\eta,t} q_\eta^v(k \mid x, \mathcal{F}_t)$.

Proof of Lemma 2. For fixed $v = z$, the conditional readout is

$$p_\eta(y = k \mid v = z, x, \mathcal{F}_t) = \frac{\kappa q_\theta^0(k|x) + C_t^{(k)} + \tau_{\eta,t} \mathbf{1}\{z=k\}}{\kappa + C_t^+ + \tau_{\eta,t}}.$$

Marginalizing over $v \sim q_\eta^v(\cdot \mid x, \mathcal{F}_t)$ gives

$$\begin{aligned} p_\eta(y = k \mid x, \mathcal{F}_t) &= \sum_{z \in \mathcal{Y}_t} q_\eta^v(z \mid x, \mathcal{F}_t) p_\eta(y = k \mid v = z, x, \mathcal{F}_t) \\ &= \frac{\kappa q_\theta^0(k|x) + C_t^{(k)} + \tau_{\eta,t} \sum_{z \in \mathcal{Y}_t} q_\eta^v(z|x,\mathcal{F}_t) \mathbf{1}\{z=k\}}{\kappa + C_t^+ + \tau_{\eta,t}} \\ &= \frac{\kappa q_\theta^0(k|x) + C_t^{(k)} + \tau_{\eta,t} q_\eta^v(k|x,\mathcal{F}_t)}{\kappa + C_t^+ + \tau_{\eta,t}}. \end{aligned}$$

Thus the marginal readout has the LVD form with

$$W_{\eta,t}^{(k)} = \tau_{\eta,t} q_\eta^v(k \mid x, \mathcal{F}_t), \qquad W_{\eta,t}^+ = \tau_{\eta,t}.$$

□

B.4 Proof of Proposition 2

Proof of Proposition 2. Let $Q_t^{\mathrm{prop}} = \alpha_t^\star / A_t$ and $\widetilde{Q}_{\eta,t}^{\mathrm{LVD}} = (\alpha_t^\star + W_{\eta,t}^{(y^\star)})/(A_t + W_{\eta,t}^+)$. We first compute the difference between the LVD correctness mass and the proposal-only correctness mass:

$$\begin{aligned} \widetilde{Q}_{\eta,t}^{\mathrm{LVD}} - Q_t^{\mathrm{prop}} &= \frac{\alpha_t^\star + W_{\eta,t}^{(y^\star)}}{A_t + W_{\eta,t}^+} - \frac{\alpha_t^\star}{A_t} \\ &= \frac{A_t(\alpha_t^\star + W_{\eta,t}^{(y^\star)}) - \alpha_t^\star(A_t + W_{\eta,t}^+)}{A_t(A_t + W_{\eta,t}^+)} \\ &= \frac{A_t W_{\eta,t}^{(y^\star)} - \alpha_t^\star W_{\eta,t}^+}{A_t(A_t + W_{\eta,t}^+)}. \end{aligned}$$

If $W_{\eta,t}$ is fixed conditional on $\mathcal{F}_t$ and $W^+_{\eta,t} > 0$, then the denominator is positive, and positive drift is equivalent to

$$A_t W^{(y^\star)}_{\eta,t} > \alpha^\star_t W^+_{\eta,t}.$$

Dividing by $A_t W^+_{\eta,t}$ gives

$$\frac{W^{(y^\star)}_{\eta,t}}{W^+_{\eta,t}} > \frac{\alpha^\star_t}{A_t} = Q^{\mathrm{prop}}_t,$$

which proves the fixed-vector condition. If $W_{\eta,t}$ is stochastic, the corresponding expected-drift condition is

$$\mathbb{E}\left[\frac{A_t W^{(y^\star)}_{\eta,t} - \alpha^\star_t W^+_{\eta,t}}{A_t(A_t + W^+_{\eta,t})} \mid \mathcal{F}_t\right] > 0.$$

□

B.5 Proof of Proposition 3

Proof of Proposition 3. The LVD posterior predictive uses the same denominator for every candidate answer. Therefore LVD ranks $y^\star$ above y_v if and only if its numerator is larger:

$$\alpha^{\mathrm{prop}}_t(y^\star) + W^{(y^\star)}_{\eta,t} > \alpha^{\mathrm{prop}}_t(y_v) + W^{(y_v)}_{\eta,t}.$$

Rearranging terms gives

$$W^{(y^\star)}_{\eta,t} - W^{(y_v)}_{\eta,t} > \alpha^{\mathrm{prop}}_t(y_v) - \alpha^{\mathrm{prop}}_t(y^\star),$$

which is Eqn. 11. □

C Limitations

The method is a headroom-recovery method, not a general claim that debate beats vote. LVD can only recover answers that have already been surfaced by the agent society. If the society contains no correct proposal, the readout has no headroom to recover. LVD can also damage cases where vote is already correct, which is why the experiments report damage rate in addition to final accuracy.

The label-free construction proxy is intentionally simple, but its reliability depends on answer clustering. It is most natural in exact-answer or near-exact-answer settings where valid answers can be normalized and grouped. Open-ended tasks may require a different proxy and a different way to define proposal-hit coverage. Appendix A.11 shows that a Hidden-Prefix representation can construct a competitive society without labels or answer identities. This is evidence for answer-string-free construction, but it remains a multiple-choice experiment and does not by itself validate semantic equivalence or evaluation for open-ended outputs.

The experiments use extra test-time compute. We therefore treat budget alignment as part of the empirical claim, not an implementation detail. We report matched or explicitly stated budgets for vote, plain debate, and LVD.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction state the supply-readout claim, the recoverable-headroom framework, and the LVD readout theory. The theory is developed in Sec. 3, experiments are reported in Sec. 5, and limitations are discussed in Appendix C.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Appendix C discusses scope limits, answer-clustering assumptions, possible damage to vote-correct cases, and the extra test-time compute required by debate.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: Sec. 3 states the definitions and conditions for the three propositions. Complete proofs are provided in Appendix B.1, Appendix B.4, and Appendix B.5. The DCM conjugacy and LVD marginalization arguments are given in Appendix B.2 and Appendix B.3.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: Sec. 5 describes the main evaluation. Appendices A.2–A.8 give data splits, decoding settings, perturbation scales, construction details, prompts, context visibility, and tie-breaking rules.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The paper uses public benchmarks and provides reproduction details in Appendices A.2–A.8. Code and reproduction instructions are available at `https://github.com/Wang-ML-Lab/when-debate-helps`.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Sec. 5 gives the main setup. Appendices A.2–A.8 specify data splits, perturbation pools, decoding parameters, token budgets, prompts, and aggregation rules.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: The main paper reports paired effects and 95% confidence intervals across five matched stochastic runs. Appendix A.5 reports per-benchmark means and standard deviations, paired confidence intervals, and two-sided paired tests, and states that these intervals quantify generation-seed variability rather than test-set resampling.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.

- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Appendix A.3 reports GPU types and memory, workers per run, observed wall-clock ranges, an exact 156.8 GPU-hour total for the five-run robustness experiment, an estimated 0.8–1.0k GPU-hours for all reported experiments, artifact storage, and the additional exploratory compute excluded from that estimate.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The study uses public benchmarks and public or publicly available language models, does not involve human subjects or private data, and reports limitations and broader impacts.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [N/A]

Justification: The work focuses on public LLMs and public reasoning data, and does not introduce a deployed system, new model, or new dataset with direct societal impact.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: The paper does not release a new pretrained model, image generator, scraped dataset, or deployed system. It evaluates existing public models and public reasoning benchmarks.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: Appendix A.4 lists the exact model and dataset versions with URLs, licenses or stated use terms, and the handling of assets whose mirrors do not declare an open license. We do not redistribute model weights or raw benchmark questions.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: The accompanying code repository documents the released implementation, dependencies, and commands needed to reproduce the reported experiments. The paper does not release a new dataset or model checkpoint.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: The paper does not involve crowdsourcing or human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: The paper does not involve crowdsourcing or human subjects, so IRB approval is not applicable.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: The experiments use Qwen and OLMo language models as the agents whose proposal, debate, and readout behavior is studied. Section 5 reports the evaluated models and protocol, while Appendices A.2 and A.8 provide the decoding configuration and complete prompts.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.