TRUST BUT VERIFY: PROVER-VERIFIER DELIBERATION FOR SELECTIVE LLM PREDICTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Reliably knowing when a language model is correct is almost as important as being correct. We introduce prover-verifier deliberation (PVD), an inference-time protocol grounded in interactive proof theory, as a mechanism for selective prediction: a prover defends a candidate answer through checkable sub-claims while a verifier issues targeted challenges and returns ACCEPT, CHALLENGE, or REJECT, so that a system can report high-confidence answers and abstain on the rest. Because frozen language models are imperfect provers and verifiers, formal soundness and completeness guarantees do not transfer; we characterize the protocol empirically through its coverage-precision behavior. Questions accepted with no answer revision — Accept + No Change (ANC) — form the reported high-confidence subset. With Claude Sonnet 4.6 as prover and Claude Haiku 4.5 as verifier on GPQA Diamond, PVD achieves HC-Prec 84.2% at HC-Cov 77% with roughly three LLM calls per question and a $\sim$30pp precision gap over the non-ANC complement. GPT and Gemini pairings show that high precision transfers across model families while verifier strictness and domain competence largely determine the size of the gap; on Humanity's Last Exam, weaker pairings collapse or invert the ANC signal, a practical failure mode of a verifier operating outside its effective region. Comparisons with self-consistency, universal self-consistency, multi-agent debate, and Reflexion suggest that PVD supplies a distinct argument-defensibility signal for selective prediction rather than reproducing sample agreement or debate consensus.[1]

1 INTRODUCTION

Reliably knowing when a language model is correct is often as important as being correct: a system that answers 90% of questions accurately but cannot tell its correct answers from its errors is less useful in deployment than one that answers 60% with near-certainty and abstains otherwise. This motivates selective prediction (Geifman & El-Yaniv, 2017) — trading coverage for precision by reporting answers only where confident — which remains underexplored for LLMs relative to the large literature on improving raw accuracy through inference-time compute.

Inference-time reasoning methods have advanced rapidly since chain-of-thought prompting (Wei et al., 2022): self-consistency (Wang et al., 2023) aggregates independent samples by majority vote, Reflexion (Shinn et al., 2023) refines answers through verbal self-reflection, Universal Self-Consistency (Chen et al., 2024) selects the most coherent candidate with an LLM aggregator, and multi-agent debate (Du et al., 2024) argues toward consensus. These methods improve accuracy, but their confidence signals are indirect proxies for correctness, and none was designed to produce a principled abstain/report decision; indeed, Maryanskyy (2026) shows that aggregation in multi-agent pipelines helps only above a selector-quality threshold s^*, below which adding diverse agents *hurts*, and PVD's empirical Effective Verifier condition (Section 6.2) is the multi-round, interactive counterpart of that static threshold.

A more structured approach comes from interactive proof theory (Goldwasser et al., 1989): a computationally powerful **prover** convinces a skeptical **verifier** through a structured challenge-response dialogue whose final verdict carries formal soundness and completeness guarantees. Recent

[1] Code, prompts, configurations, and per-call result logs are provided as anonymized supplementary material and will be released publicly with the camera-ready version.

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

work brings this paradigm to learning systems — Neural Interactive Proofs (Hammond & Adam-Day, 2025), human-LLM interactive proofs (Zhang & Sedoc, 2025), and self-proving models (Amit et al., 2025) — with trained agents and formal verifiability. We take a complementary direction: we ask whether the **structure** of interactive proof dialogue, without training, without verifiable ground truth, and over open-domain knowledge questions, can serve as a practical calibration mechanism for frozen LLMs.

We introduce Prover-Verifier Deliberation (PVD), a challenge-guided selective-prediction protocol in which a prover defends a candidate answer and a verifier issues targeted challenges before returning an Accept, Challenge, or Reject verdict. The dialogue terminates when the verifier reaches a verdict or a fatigue limit is hit. Because language models are imperfect agents operating over a noisy channel rather than computationally bounded reasoners, hard soundness and completeness guarantees do not transfer; instead, we characterize the protocol empirically as a selective predictor. We introduce Accept + No Change (ANC) — questions where the prover's answer is accepted without revision — as a coverage-precision operating point, and evaluate it against standard baselines across two benchmarks and multiple hosted model pairings.

Our main findings are: (i) on GPQA Diamond (Rein et al., 2024), ANC questions achieve HC-Prec 84–98% at HC-Cov 43–77%, a +6.6–34.8pp gap over the non-ANC complement; (ii) the strongest signal comes from a within-family capability-asymmetric pairing, GPT-5.4 with a GPT-5.4-mini verifier (HC-Prec 97.6% at HC-Cov 43%, +34.8pp over the complement); (iii) on Humanity's Last Exam (HLE) (Center for AI Safety et al., 2026), weak pairings collapse or invert the Accept/Reject gap, and because measuring the gap requires ground-truth labels it is a diagnostic for *evaluation* rather than a signal a deployed system can monitor unaided (Section 6.2; Appendix A.8 tests a stronger model as a stand-in for the labels); (iv) on GPQA, simpler signals from the same prover are competitive or stronger: self-consistency (k=8) exceeds PVD's precision at matched coverage, and verbalized confidence has AURC 7.7% (Table 7) against 11.8% for standard PVD and 7.6% with a same-model verifier (Figure 3); what PVD adds is a verdict the prover must earn from a separate verifier, which an instructed deceiver obtains on only 6–9% of questions (Section 6.4).

2 Background & Related Work

Our work connects five lines of research: selective prediction and abstention, calibration and uncertainty estimation for language models (Appendix A.1.1), inference-time reasoning and aggregation (Appendix A.1.2), verification and multi-agent deliberation (Appendix A.1.3), and interactive proofs and prover-verifier games. We position prover-verifier deliberation (PVD) as a selective-prediction protocol: the goal is not only to improve the final answer, but to produce a structured report/abstain signal from the interaction itself. In this sense, PVD draws conceptual structure from interactive proof systems while remaining an empirical inference-time method for frozen, general-purpose LLMs.

2.1 Interactive Proofs and Prover-Verifier Games

Interactive proof systems formalize verification as a multi-round protocol between a powerful prover and a computationally bounded probabilistic verifier (Goldwasser et al., 1989). Classical results such as IP = PSPACE show that interaction and randomness can greatly expand what can be efficiently verified (Shamir, 1992). These results provide the conceptual template for PVD: a claim is not accepted merely because it is asserted, but because it survives a challenge-response process.

Neural Interactive Proofs (Hammond & Adam-Day, 2025) study learned prover-verifier protocols; human-LLM interactive proofs (Zhang & Sedoc, 2025) ask when a human verifier can adjudicate LLM-generated arguments; scalable oversight with weak judges (Kenton et al., 2024) asks whether a weaker judge reaches the *right answer* under debate and consultancy; and prover-verifier games (Kirchner et al., 2024) train provers for legibility to a smaller verifier. We share the capability-asymmetric structure but ask a different question: whether the interaction's terminal verdict *partitions* the answers into a reliable reported subset and a withheld complement, using frozen models and no training. The interaction itself has a lineage from Goldwasser et al. (1989) through Hammond & Adam-Day (2025), Amit et al. (2025), and Kenton et al. (2024); our contribution is the use of unchanged acceptance (ANC) as a confidence signal, its coverage-precision characterization, and the identification of its failure regime (Appendix A.3 expands this positioning).

PVD occupies a complementary niche: we train nothing, assume no ground-truth oracle during interaction, and claim no formal soundness or completeness, since frozen LLMs are imperfect provers and verifiers that may hallucinate, miss valid challenges, reject correct arguments, or accept incorrect ones. We instead ask an empirical selective-prediction question: **when the interactive-proof structure is instantiated with off-the-shelf LLMs on open-domain expert questions, does the verifier's verdict separate reliable answers from answers that should be withheld?**

2.2 Soundness, Completeness, and the Effective Verifier

Interactive proof systems carry two formal guarantees: **completeness** — an honest prover of a true claim $s \in L$ is accepted with probability at least $1 - p^T$ after T rounds, where $p < 1$ is the per-round probability of challenge — and **soundness** — a false claim $s \notin L$ is accepted with probability at most q^T, where $q < 1$ is the per-round slip-through probability — with both error rates decreasing exponentially in interaction depth. One qualification matters for what follows: the q^T bound presumes that a false claim must survive all T rounds, whereas a protocol that may accept at round $t < T$ — as PVD does — is bounded only by q^t on that question, so early acceptance trades the depth of the theoretical bound for inference cost; our challenge-first variant, which forbids acceptance before at least one challenge, restores one round of this decay by construction.

Zhang & Sedoc (2025) formalize the conditions under which these guarantees hold when the verifier is human and the prover is an LLM. Their central requirement is the **Effective Verifier** assumption: the verifier must be able to confirm valid responses and identify errors within a finite fatigue limit T. One structural condition makes this assumption plausible in practice, and one competence condition determines whether it actually holds.

Asymmetric cognitive load. The protocol is feasible precisely because the prover bears the burden of full proof generation while the verifier only checks individual steps locally — confirming that each sub-claim is valid and follows from the prior exchange. This asymmetry keeps the verification task tractable even for resource-bounded agents: step-wise evaluation is far less demanding than global correctness judgment.

Sufficient verifier competence. Local checking is cheaper than global judgment only if the verifier can actually evaluate the steps it is shown. Writing I for the set of sub-claims the verifier is competent to assess and c_i for its probability of judging sub-claim i correctly, the exponential bounds above require c_i bounded away from chance across a large enough I. When the verifier's domain knowledge is thin, $|I|$ shrinks and c_i drifts toward $\frac{1}{2}$, and neither bound survives.

These conditions do not automatically hold for a frozen-LLM verifier, which may issue spurious rejections (violating completeness) or fail to detect a flawed argument (violating soundness) whenever it operates near its own competence ceiling. Hence our empirical framing: rather than claiming formal soundness and completeness, we ask whether the interactive-proof structure, instantiated with frozen LLMs, produces verdicts that *empirically* separate reliable answers from unreliable ones. The collapse of the ACCEPT/REJECT accuracy gap on Humanity's Last Exam is the signature of an empty effective region — the verifier lacks the domain knowledge $|I|$ to maintain c_i at levels sufficient for either property, and its verdict degrades to near-chance — whereas the strong GPQA results arise in Chemistry and Physics, where the verifier's knowledge sustains adversarial challenge across many rounds.

3 Prover-Verifier Deliberation

3.1 Protocol

Prover-Verifier Deliberation (PVD) is a multi-round challenge-response protocol that produces both an answer and a structured confidence verdict. Two participants, a **Prover** $\mathcal{P}$ and a **Verifier** $\mathcal{V}$, each a frozen language model, engage over a multiple-choice question q with candidate answers $\mathcal{A}$.

Initialization and information. $\mathcal{P}$ selects a candidate answer $\hat{a} \in \mathcal{A}$ and produces a structured proof: a *statement* asserting why $\hat{a}$ is correct, 3–5 *atomic sub-claims* (each independently verifiable), and a brief *reasoning* paragraph. $\mathcal{V}$ sees q, $\mathcal{A}$, and this proof, *but not the ground-truth answer*: its verdict

reflects only what can be inferred from the proof itself, which distinguishes PVD from LLM-as-judge and reward-modelling settings where the judge has the target label.

Challenge-response loop. At each round $t \in \{1, \ldots, T\}$, $\mathcal{V}$ examines the current proof transcript and issues one of three verdicts.

- ACCEPT: every sub-claim is specific, verifiable, and mutually consistent. The protocol terminates with acceptance.
- REJECT: a clear logical flaw, factual error, or internal contradiction has been identified. The attempt terminates in rejection.
- CHALLENGE: $\mathcal{V}$ identifies the single most suspicious or least-justified sub-claim $\tilde{c}$ and poses a targeted question. $\mathcal{P}$ responds by justifying $\tilde{c}$ more rigorously; it may revise $\hat{a}$ if it detects an error. The dialogue proceeds to round $t{+}1$.

Fatigue, retry, and fallback. If $\mathcal{V}$ issues T challenges without a terminal verdict, the attempt ends in a *fatigue rejection*. An attempt that ends in Reject, in fatigue, or in an Accept that followed an answer change triggers a new attempt (up to K total), with a brief templated summary of the prior attempts given to $\mathcal{P}$ as adversarial context; if no attempt reaches ACCEPT with no answer revision, the final answer is a majority vote over all K attempts' final answers.

Accept + No Change (ANC). An attempt is ANC if $\mathcal{V}$ issues ACCEPT and $\mathcal{P}$'s answer was never revised across the challenge rounds. ANC is the primary selective-prediction signal: ANC questions form the reported subset, the remainder are withheld or answered by majority vote, and an ACCEPT that followed a revision — the prover corrected itself under pressure, evidence of prior uncertainty — is not treated as high-confidence.

Coverage and precision. We evaluate ANC as a binary classifier that predicts "the answer is correct." Treating ANC as the positive prediction, the two metrics are: **HC-Cov** $= \Pr[\text{ANC}]$ (coverage: the fraction of questions reported) and **HC-Prec** $= \Pr[\text{correct} \mid \text{ANC}]$ (precision: accuracy on the reported subset). HC-Prec is the standard precision of the ANC classifier; its complement is the *selective risk* of Geifman & El-Yaniv (2017). We report HC-Cov rather than recall because the non-ANC complement is intended for downstream remediation (escalation to a more capable model, extended compute, or human review) rather than counted as a missed prediction. The **Gap** between HC-Prec and accuracy on the non-ANC complement quantifies the information the ANC verdict carries beyond raw accuracy; the same definitions apply to each baseline's natural high-confidence subset (full consensus for SC/USC; agent consensus for Debate; stability for Reflexion).

3.2 ALGORITHM

Algorithm 1 (Appendix A.4) gives the pseudocode. Separate histories $h_{\mathcal{P}}$ and $h_{\mathcal{V}}$ are maintained for prover and verifier respectively. Two information-hiding constraints hold throughout: $\mathcal{V}$ does not observe $\mathcal{P}$'s internal reasoning, only its structured outputs (statement, sub-claims, reasoning paragraph); and $\mathcal{V}$ does not observe the ground-truth answer, only the candidate set $\mathcal{A}$ and $\mathcal{P}$'s defence of one of its elements.

4 EXPERIMENTS

4.1 BENCHMARKS AND EVALUATION SETTING

We evaluate PVD on benchmarks that stress advanced reasoning and expert knowledge: GPQA Diamond, a graduate-level, domain-expert benchmark in biology, physics, and chemistry designed to be difficult for non-experts even with web access (Rein et al., 2024), and Humanity's Last Exam, a broad expert-level benchmark that probes frontier academic knowledge and calibration (Center for AI Safety et al., 2026). Both expose the distinction between being often correct and knowing when an answer is reliable, and when prover and verifier operate near their competence ceiling the collapse of the ACCEPT–REJECT accuracy gap is itself informative: the protocol no longer provides a trustworthy selection signal.

4.2 Baselines and Evaluation Protocol

Baselines. We compare PVD against a one-call reference point and four inference-time reasoning methods that represent the dominant approaches to improving LLM accuracy through additional computation.

A *direct* chain-of-thought call is the one-call reference point (the Sonnet 4.6 and Gemini 3.1 Pro rows are new single-call runs with the same decoding settings as their PVD provers, Appendix A.5; the GPT-5.4 row is from the original runs); for Sonnet 4.6 we also report a *single context play* ablation in which one response contains both prover and verifier blocks, exposing an ANC signal without separating the two histories. *Self-Consistency* (Wang et al., 2023) draws $k{=}8$ independent chains and returns the majority answer, with full consensus as its confidence signal; *Universal Self-Consistency* (Chen et al., 2024) replaces the vote with an LLM aggregator over the same samples and uses the same signal; *Multi-agent Debate* (Du et al., 2024) runs three agents that revise over two rounds, with unanimous agreement as the signal; *Reflexion* (Shinn et al., 2023) refines one answer over up to five trials, with a final answer stable across trials and unchanged from the initial response as the signal and, because its ground-truth failure oracle is unavailable at inference time, a self-consistency recheck (a second independent sample at the same temperature) as the failure detector.

Model configurations and protocol parameters. We evaluate six prover-verifier configurations along two axes: **within-family capability-asymmetric pairings**, with a smaller and more skeptical verifier from the same family (Claude Sonnet 4.6 / Haiku 4.5; GPT-5.4 / GPT-5.4-mini), and **cross-family pairings** (Gemini 3.1 Pro / GPT-5.5-pro on GPQA; GPT-5.5 / Gemini 3.1 Pro on HLE), which test whether family-diverse verification improves calibration; a single context play variant with one model in both roles is the separation ablation (§6). Two open-weight in-family pairings run under the identical protocol — Gemma-4-31b-it with Gemma-4-26b-a4b-it (a 4B-active mixture-of-experts verifier) and DeepSeek-v4-pro with DeepSeek-v4-flash. The fatigue limit T caps verifier turns per attempt (reaching T without a verdict counts as a rejection) and the retry limit K caps independent attempts before the majority-vote fallback. Unless noted, two-model single-attempt configurations use $K{=}1$ and $T{=}12$; the single context play ablation uses one call with an internal $T{=}6$; the strongest GPQA cross-family retry configuration uses $K{=}6$, $T{=}12$ and is examined separately as a cost-performance tradeoff. Prompts are in Appendix A.13.

Cost, evaluation passes, and decoding. We measure cost in LLM calls per question (a single PVD attempt that terminates on verifier turn t uses $2t$ calls; single-attempt PVD averages 3–6 calls per question and the $K{=}6$ retry protocol $\approx$29) and in provider-logged tokens converted to USD at list prices. Table 1 reports our original runs; every confidence interval, risk-coverage curve, and significance test in the paper comes from a second, fully logged re-execution of the Sonnet 4.6 suite whose base configuration is the strict verifier prompt, so its "standard" PVD is the analogue of Table 1's PVD† rather than of its PVD. The two passes, the decoding settings, and the model identifiers are documented in Appendix A.5.

5 Results

5.1 GPQA Diamond

Table 1 reports overall accuracy, the coverage and precision of each method's high-confidence (HC) subset, the resulting gap, and the mean number of LLM calls per question; the **Acc** column reports the prover's final answer on *all* questions, Appendix A.6 explains why full-population accuracy is not the protocol's objective, and Appendix A.9 quantifies the escalation that a non-ANC verdict enables. For PVD, the HC subset is ANC; for baselines, it is the method's natural confidence signal (full consensus for SC/USC; agent consensus for Debate; stability for Reflexion). An example PVD conversation is in Appendix A.14.

Per-domain breakdowns of the ANC signal for three representative configurations are given in Appendix A.6 (Table 4); Biology is the smallest domain ($n{=}19$) and its numbers should be read accordingly.

Table 1: GPQA Diamond results grouped by prover. **HC-Cov**: fraction of questions flagged high-confidence. **HC-Prec**: accuracy on that subset. **Gap**: HC-Prec minus accuracy on the complement. **Calls**: mean LLM calls per question. *: SC with extended thinking (Epoch AI benchmark); not directly comparable to PVD runs (standard API). †: challenge-first verifier prompt. ‡: HC coverage >90%, leaving n<20 in the complement; gap estimate unreliable. ¶: open-weight in-family pairings, reported over attempted questions (n=189 Gemma, 193 DeepSeek) after excluding items on which the prover exhausted its token budget without answering (Appendix A.12).

Method	Verifier	HC Signal	Acc	HC-Cov	HC-Prec	Gap	Calls
Sonnet 4.6 as prover							
Direct	—	—	76.3%	—	—	—	1
Single context play	Self	ANC	78.8%	63%	83.9%	+13.6	1
SC (k=8)	—	Full consensus	83.3%	72%	91.5%	+29.0	8
USC (k=8)	Sonnet 4.6	Full consensus	81.8%	72%	91.5%	+29.0	9
Debate (3×2)	Sonnet 4.6	Agent consensus	83.3%	95%	85.7%	+52.4‡	9
Reflexion	—	Stable, unchanged	82.3%	93%	82.6%	+4.0‡	∼2.5
PVD	Haiku 4.5	ANC	76.8%	77%	84.2%	+32.0	∼3
PVD†	Haiku 4.5	ANC	79.8%	65%	89.9%	+29.1	∼6
GPT-5.4 as prover							
Direct	—	—	72.7%	—	—	—	1
SC* (k=8)	—	Full consensus	94.9%	90%	97.2%	+23.5‡	8
PVD	GPT-5.4-mini	ANC	77.8%	43%	97.6%	+34.8	∼3
Gemini 3.1 Pro as prover							
Direct	—	—	93.4%	—	—	—	1
SC* (k=8)	—	Full consensus	93.9%	94%	97.3%	+60.9‡	8
PVD	Flash-Lite	ANC	94.4%	57%	97.3%	+6.6	∼4
PVD+retry	GPT-5.5-pro	ANC	92.9%	75%	97.3%	+17.3	∼29
Gemma-4-31b-it as prover¶							
PVD	Gemma-4-26b-a4b-it	ANC	86.8%	70%	90.2%	+11.2	∼2
DeepSeek-v4-pro as prover¶							
PVD	DeepSeek-v4-flash	ANC	89.6%	87%	91.7%	+15.7	∼3

5.2 HUMANITY'S LAST EXAM

Table 5 (Appendix A.6) reports results for the GPT-5.5 prover / Gemini 3.1 Pro verifier configuration on the 513 multiple-choice questions of HLE — every item whose `answer_type` is `multipleChoice` in the release we used; exact-match and free-response items are excluded because the protocol assumes a fixed answer alphabet — broken down by domain. This configuration uses the strongest available models and represents the frontier capability ceiling in our evaluation. Table 2 summarizes the ANC gap for all three HLE pairings; Section 6.2 discusses its dependence on capability.

Table 2: HLE ANC gap by model pairing. **HC-Cov** = $\Pr[\text{ANC}]$; **HC-Prec** = $\Pr[\text{correct} \mid \text{ANC}]$; **Gap**: HC-Prec minus non-ANC accuracy.

Prover	Verifier	Acc	HC-Cov	HC-Prec	Gap
Sonnet 4.6	Haiku 4.5	20.1%	31%	15.2%	−7.1
Opus 4.6	Sonnet 4.6	40.0%	67%	41.4%	+4.3
GPT-5.5	Gemini 3.1 Pro	45.6%	52%	59.0%	+27.9

5.3 VERIFIER CHOICE AND STRICTNESS

Table 6 (Appendix A.6) holds the prover fixed and varies the verifier to isolate the contribution of verifier strictness and model pairing. Two provers are shown: Sonnet 4.6 and Gemini 3.1 Pro, each paired with verifiers of increasing independence and strictness.

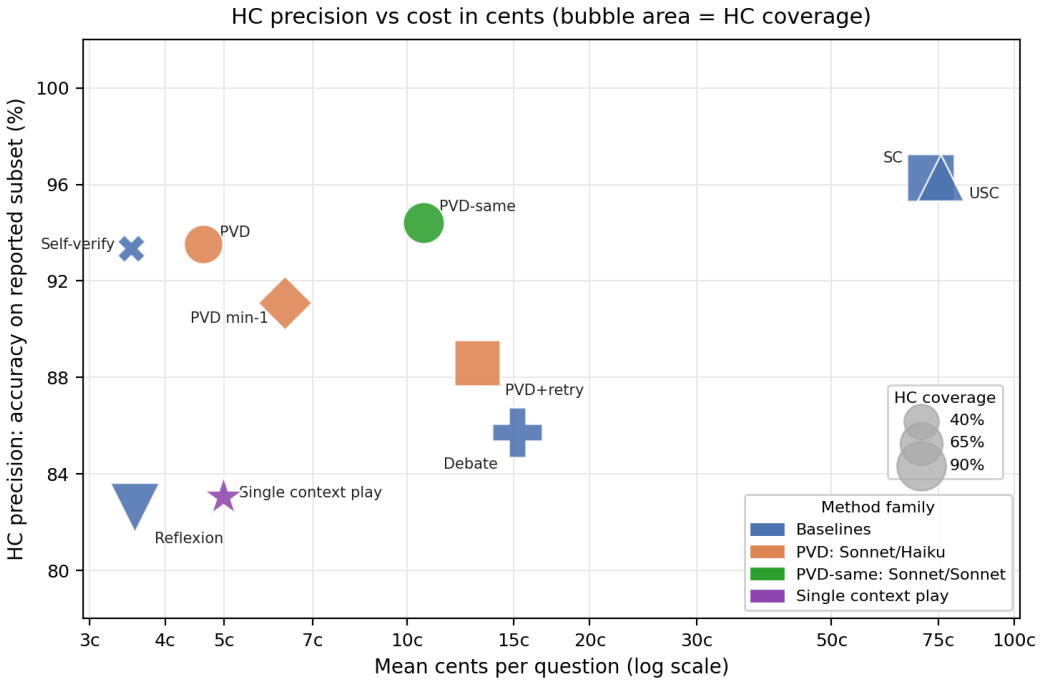

Figure 1: Cost–precision tradeoff for the re-executed Sonnet 4.6 suite on GPQA Diamond (Table 3; Appendix A.5): y is HC-Prec, x mean cents per question (log scale), bubble area HC-Cov. PVD variants occupy the high-precision region with a controllable cost–coverage tradeoff.

6 Discussion

6.1 Argument Defensibility as a Correctness Signal

The ANC signal is not merely a restatement of the prover's prior confidence. On GPQA Diamond, HC-Prec ranges from 84–98% across all configurations, consistently above both overall accuracy and the non-ANC complement. But the more telling evidence comes from the structure of errors.

Logical and reasoning errors. The verifier can identify internal inconsistencies or invalid inferential steps without domain expertise: on GPQA Chemistry the GPT-5.4-mini verifier reaches HC-Prec 100% with a $+46.2$pp gap (Table 4) despite being much smaller than the prover, because multi-step derivations break on a single wrong sign, stoichiometric ratio, or mislabelled intermediate, and asking "how does step 2 follow from step 1?" catches that without knowing the answer. **Insufficient knowledge.** A confident but wrong prior leaves no traceable reasoning error, so catching it requires independent domain knowledge; the same verifier's modest Physics gap ($+9.9$pp) reflects this asymmetry. **Effective Verifier.** A verifier that issues challenges but accepts any confident-sounding reply is not verifying. Gemini Flash-Lite reaches HC-Prec 97.3% with a gap of only $+6.6$pp not because it fails to discriminate — it rejects 8 of the prover's 11 incorrect answers (72.7%) but only 78 of 187 correct ones (41.7%) — but because the prover is right 94.4% of the time: high prover accuracy caps the achievable gap, and the verifier's contribution is better read from the rejection rates. Together these modes predict when PVD works. A verifier that can identify reasoning errors or has relevant domain knowledge yields a large gap; one that is easily satisfied or lacks the knowledge to sustain adversarial pressure collapses it, as HLE confirms (§6.2).

Further analyses (Appendix A.7). A single context play ablation with one model in both roles keeps HC-Prec at 84% but narrows the gap by $\sim$15pp, so separation matters beyond the sub-claim decomposition. Stability under re-sampling (Reflexion's signal) is only marginally informative; one round of self-verification with sub-claim checking (Weng et al., 2023) matches PVD's precision (93.3% vs. 93.5%) at under half its coverage (22.7% vs. 54.5%), so interactive challenge buys

coverage rather than precision. At a matched call budget extended-thinking SC@4 beats single-attempt PVD on the gap ($+52.8$ vs. $+42.4$pp); at a matched dollar budget every PVD configuration dominates. Coverage and gap saturate by $T{\approx}6$, and the three verdict tiers are strictly ordered in accuracy. Finally, two label-free self-report baselines on the same prover's *direct* answers — stated 0–100 confidence (Lin et al., 2022; Tian et al., 2023) and sampled P(True) (Kadavath et al., 2022) — are compared with ANC at matched coverage in Table 7: for Sonnet 4.6 verbalized confidence is a strong selective predictor (AURC 7.7%, matching or exceeding single-attempt ANC precision at PVD's coverage), so what PVD adds is verdict information that self-report lacks rather than precision at a fixed coverage, while P(True) saturates and ranks poorly.

6.2 Verifier Capability and the Effective Verifier Condition

Table 2 shows how the gap responds to prover and verifier capability. The most striking entry is not the GPT-5.5 / Gemini 3.1 Pro gap of $+27.9$pp but the Sonnet 4.6 / Haiku 4.5 gap of -7.1pp: the ANC questions are *less* accurate than the rest. This inversion is the empirical signature of an empty effective region in the sense of Zhang & Sedoc (2025): Haiku can construct a non-trivial challenge only where it has some domain knowledge, so it challenges what it dimly recognizes and accepts what it cannot evaluate at all, inverting the intended selection — the interactive fingerprint of a verifier below the selector-quality threshold s^* of Maryanskyy (2026).

The inversion is domain-localized. Under the strongest HLE pairing (aggregate $+27.9$pp), seven of eight domains show a positive gap — Math $+49.1$pp, Chemistry $+48.4$pp, Computer Science/AI $+36.3$pp, Other $+34.4$pp, Humanities/Social Science $+24.1$pp, Biology/Medicine $+17.1$pp, Physics $+14.6$pp — and exactly one inverts: **Engineering**, at -24.3pp (ANC precision 31.2% [14.2, 55.6] against 55.6% on the complement, $n{=}25$). The same domain inverts under the weakest pairing (ANC $0/8$ correct, gap -11.8pp), and the verifier *over*-accepts exactly there ($16/25$, 64%, against $268/513$, 52%, overall) while its accepted subset is less accurate than its rejected one: the confidently-wrong pattern the Effective Verifier account predicts, localized to a single domain, and suggestive rather than decisive at $n{=}25$.

A related pattern appears within GPQA when the verifier is varied with the prover held fixed (Table 6); the GPT-5.5-pro row is a $K{=}6$ retry run, but its first attempt gives the same gap ($+17.7$pp), so the widening is the verifier's, not the compute's (Appendix A.6).

6.3 PVD and SC Are Not Measuring the Same Thing

Self-Consistency and PVD identify high-confidence subsets from the same base model (Sonnet 4.6) that overlap only partially (Table 10 in Appendix A.10; the PVD column is the challenge-first configuration PVD† of Table 1, at 65% coverage). The 54% overlap at 96.3% precision is the shared easy set; the disagreements are more informative. Where SC is confident but PVD rejects (35 questions), SC's accuracy falls to 77.1%, and PVD rejected 8 of the 12 cases where SC was confident yet wrong; conversely, SC had flagged non-consensus on 9 of the 13 questions PVD accepted wrongly. PVD's weak spot is the sharper: on the 22 questions PVD accepted without SC consensus its accuracy is 59.1% against SC's majority vote of 72.7%, so a solo ANC verdict that sample consistency does not corroborate carries a roughly 40% error rate.

The mechanisms differ: SC measures **output consistency**, which correlates with correctness but can amplify confident hallucinations, whereas PVD measures **argument defensibility**, which is sensitive to reasoning structure but blind to consistency across samples. The practical implication is an ensemble signal: questions flagged by *both* methods reach 96.3% precision, the highest subset in Table 10, at $\sim$54% coverage. A "PVD-consistency" variant that aggregates independent deliberations, and how its false-acceptance behaviour differs from retry, is discussed in Appendix A.10.

6.4 Soundness Under a Deceptive Prover

We test the assumption that an Accept reflects an argument surviving scrutiny by instructing the prover (Sonnet 4.6) to argue convincingly for a wrong answer against an unchanged, unaware verifier (Haiku 4.5), naively and in a *strategic* condition coached to exploit the capability asymmetry (all of GPQA Diamond, $T{=}12$). The verifier accepts the deception on 6.1% and 9.1% of questions respectively, against 87.2% acceptance of honest correct answers, a 10–$14\times$ separation; every catch

was an outright REJECT, reached faster than honest acceptance (1.88 and 1.77 vs. 3.0 mean rounds), the prover never abandoned the lie under challenge ($0/396$), and coaching bought only $+3$pp. This is the empirical analogue of soundness: it bounds, without eliminating, the shared-misconception concern of Section 7, for one instructed adversary on one pairing rather than as a general guarantee (Appendix A.11).

7 LIMITATIONS

Several limitations bound these conclusions. First, all experiments use English-language multiple-choice benchmarks with a fixed answer alphabet, an unambiguous correctness criterion that may not transfer to open-ended generation, multilingual tasks, long-form synthesis, or partially correct answers.

Second, most agents are proprietary hosted models; two open-weight pairings (Table 1) reproduce the signal at reduced magnitude, but the other results are evaluations of the specified models at the time they were run and are sensitive to provider-side updates and decoding defaults.

Third, we use the protocol's native verdict signals rather than a learned or distribution-free selector; conformal risk control and a learned confidence head over protocol features (ANC, rounds, answer changes, SC agreement) would need calibration splits that our benchmarks are too small to supply, and are deferred to future work.

Fourth, we measure cost in calls and tokens rather than wall-clock latency; one latency cost is intrinsic, since deliberation needs three to six *dependent* round trips (up to 28 under retry) where SC's $k{=}8$ samples need a single parallel round trip.

Fifth, comparisons at native operating points can conflate a better selector with a more conservative one, so we report risk-coverage curves where possible, with two consequences. PVD-min1, which forbids first-round acceptance and which we had presented as a calibration improvement, is an operating-point control: at matched coverage the same strict-prompt configuration *without* the enforcement lies above it on the curve. And the retry protocol's false-acceptance risk grows with K: the gap peaks at $K{=}2$, consistent with additional attempts eventually purchasing acceptances rather than correctness.

Sixth, gaps estimated against very small complements are unreliable and are marked as such: Debate at 95.5% coverage leaves a nine-question complement and a bootstrap interval roughly 68 points wide, and Reflexion's gap is not significant ($+4.0$pp, $[-17.7, +29.1]$, $p{=}0.72$); neither is comparable evidence to the well-estimated rows.

Finally, PVD carries no adversarial robustness guarantee and is neither an AI-control protocol nor a formal interactive-proof system: a colluding model pair, or a verifier outside its domain competence, can collapse or invert the ANC signal, as the HLE results show. Section 6.4 bounds one instructed attack on one pairing (6.1–9.1% acceptance against 87.2% for honest correct answers) but not adaptive or collusive strategies; PVD remains useful only where the verifier's targeted challenges are informative.

8 CONCLUSION

We introduced Prover-Verifier Deliberation (PVD), which instantiates the challenge-response structure of interactive proofs with frozen LLMs as a selective-prediction signal: a prover decomposes a candidate answer into atomic sub-claims and defends them under adversarial challenge, a verifier terminates with Accept, Challenge, or Reject, and questions accepted without revision (ANC) form the high-confidence reported subset, which we characterize empirically rather than through formal soundness and completeness.

The broader implication is architectural: a two-model deliberation at three to six calls per question yields a high-precision reported subset without trained reward models or SC-scale ensembles, with a qualitatively different error profile, provided the verifier is inside its effective region. The ANC gap collapse on HLE shows what happens outside it, and because detecting that collapse requires labels, a deployment must characterize its verifier on a labeled slice of the target domain before trusting the signal, as it would for any other selective predictor.

Reproducibility Statement

All protocol code, configuration files, prompts, and result files are provided as anonymized supplementary material and will be released publicly with the camera-ready version. Each result file from the second evaluation pass carries a header record with the git commit, configuration, model identifiers, dataset seed, and per-call token counts (Appendix A.5). The two evaluation passes, and the verifier-prompt difference between them, are described in Appendix A.5; all system prompts and user-turn templates appear verbatim in Appendix A.13, and the HLE subset is defined in Section 5.2 (the 513 text-only multiple-choice items; questions with images were skipped at load time). Every table in the paper is generated by a script in `tables/` from the released result files, and the statistical procedures are specified in Appendix A.2.2. Benchmark data are not redistributed: GPQA Diamond and Humanity's Last Exam are loaded from their public releases under their original terms. Because all evaluated models are hosted and not versioned beyond their public identifiers, token-level outputs may not replicate exactly; the reproducibility claim is for the protocol, the analysis pipeline, and the released logs.

Ethics Statement

This work evaluates inference-time protocols on public academic benchmarks with hosted commercial models; it involves no human subjects, no personal data, and no data collection. The deceptive-prover experiment (Section 6.4) instructs a model to argue for a known-wrong answer in order to measure a verifier's resistance; it produces no artifact intended to mislead people and is reported to characterize a failure mode. Two deployment cautions follow from the results: a selective-prediction signal whose reliability depends on verifier competence can invert outside that competence (Section 6.2), and it should be characterized on a labeled slice of any new domain before its acceptances are trusted.

AI Use Statement

Generative AI tools (Anthropic's Claude models, used through an agentic coding assistant) were used in this work for the following tasks. *Implementing methods:* the evaluation harness—protocol drivers, baselines, table and figure scripts, and statistical analyses—was written with AI assistance and was reviewed, tested, and executed by the authors. *Designing and providing feedback on methodology and experiments:* the assistant was consulted on the design of the re-execution pass, the matched-budget comparison, and the confidence-baseline analyses; the authors made all design decisions. *Interpreting results and cleaning and reformatting data:* AI assistance was used to aggregate result logs, compute summary statistics, and reconcile the two evaluation passes described in Section 4. Generative AI was not used to generate synthetic datasets, to develop the theoretical framing, to formulate or prove mathematical claims, to propose the central hypotheses, for translation, or for qualitative thematic analysis. Additionally, AI tools were used for paper editing, LaTeX formatting, figure preparation, and literature lookup. We have reviewed all AI-assisted work: code was unit-tested and re-run end to end, statistics were recomputed from the raw per-call logs, and citations were checked against their sources. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI. The language models evaluated in this paper are the object of study rather than writing assistance; their use is described in Section 4.

References

Noga Amit, Shafi Goldwasser, Orr Paradise, and Guy Rothblum. Models that prove their own correctness. In *Advances in Neural Information Processing Systems*, 2025. arXiv:2405.15722.

Center for AI Safety, Scale AI, and HLE Contributors Consortium. A benchmark of expert-level academic questions to assess AI capabilities. *Nature*, 649:1139–1146, 2026. doi: 10.1038/s41586-025-09962-4. URL https://arxiv.org/abs/2501.14249.

Jiefeng Chen, Jie Ren, Xinyun Chen, Chengrun Yang, Ruoxi Sun, Jinsung Yoon, and Sercan O Arik. SETS: Leveraging self-verification and self-correction for improved test-time scaling. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856. URL https://openreview.net/forum?id=Wv9NMJoKww.

Xinyun Chen, Renat Aksitov, Uri Alon, Jie Ren, Kefan Xiao, Pengcheng Yin, Sushant Prakash, Charles Sutton, Xuezhi Wang, and Denny Zhou. Universal self-consistency for large language models. In *ICML 2024 Workshop on In-Context Learning*, 2024. URL https://openreview.net/forum?id=LjsjHF7nAN.

C. Chow. On optimum recognition error and reject tradeoff. *IEEE Transactions on Information Theory*, 16(1):41–46, 1970. doi: 10.1109/TIT.1970.1054406.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Shehzaad Dhuliawala, Mojtaba Komeili, Jing Xu, Roberta Raileanu, Xian Li, Asli Celikyilmaz, and Jason Weston. Chain-of-verification reduces hallucination in large language models. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 3563–3578, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.212. URL https://aclanthology.org/2024.findings-acl.212/.

Yilun Du, Shuang Li, Antonio Torralba, Joshua B. Tenenbaum, and Igor Mordatch. Improving factuality and reasoning in language models through multiagent debate. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=zj7YuTE4t8.

Yonatan Geifman and Ran El-Yaniv. Selective classification for deep neural networks. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/4a8423d5e91fda00bb7e46540e2b0cf1-Paper.pdf.

Yonatan Geifman and Ran El-Yaniv. SelectiveNet: A deep neural network with an integrated reject option. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2151–2159. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/geifman19a.html.

Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proof systems. *SIAM Journal on Computing*, 18(1):186–208, 1989. doi: 10.1137/0218012. URL https://doi.org/10.1137/0218012.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In Doina Precup and Yee Whye Teh (eds.), *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1321–1330. PMLR, 06–11 Aug 2017. URL https://proceedings.mlr.press/v70/guo17a.html.

Lewis Hammond and Sam Adam-Day. Neural interactive proofs. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=R2834dhBlo.

Zhengbao Jiang, Jun Araki, Haibo Ding, and Graham Neubig. How can we know when language models know? on the calibration of language models for question answering. *Transactions of the Association for Computational Linguistics*, 9:962–977, 2021. doi: 10.1162/tacl_a_00407. URL https://aclanthology.org/2021.tacl-1.57/.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, et al. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*, 2022.

Amita Kamath, Robin Jia, and Percy Liang. Selective question answering under domain shift. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault (eds.), *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 5684–5696, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.503. URL https://aclanthology.org/2020.acl-main.503/.

Zachary Kenton, Noah Y. Siegel, János Kramár, Jonah Brown-Cohen, Samuel Albanie, Jannis Bulian, Rishabh Agarwal, David Lindner, Yunhao Tang, Noah D. Goodman, and Rohin Shah. On scalable oversight with weak LLMs judging strong LLMs. In *International Conference on Machine Learning (ICML)*, 2024.

Akbir Khan, John Hughes, Dan Valentine, Laura Ruis, Kshitij Sachan, Ansh Radhakrishnan, Edward Grefenstette, Samuel R. Bowman, Tim Rocktäschel, and Ethan Perez. Debating with more persuasive LLMs leads to more truthful answers. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 23662–23733. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/khan24a.html.

Jan Hendrik Kirchner, Yining Chen, Harri Edwards, Jan Leike, Nat McAleese, and Yuri Burda. Prover-verifier games improve legibility of llm outputs. *arXiv preprint arXiv:2407.13692*, 2024.

Tian Liang, Zhiwei He, Wenxiang Jiao, Xing Wang, Yan Wang, Rui Wang, Yujiu Yang, Shuming Shi, and Zhaopeng Tu. Encouraging divergent thinking in large language models through multi-agent debate. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 17889–17904, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.992. URL https://aclanthology.org/2024.emnlp-main.992/.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 39578–39601, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/aca97732e30bcf1303bc22ac3924fd16-Paper-Conference.pdf.

Stephanie Lin, Jacob Hilton, and Owain Evans. Teaching models to express their uncertainty in words. *Transactions on Machine Learning Research*, 2022. ISSN 2835-8856. URL https://openreview.net/forum?id=8s8K2UZGTZ.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. Self-refine: Iterative refinement with self-feedback. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 46534–46594. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/91edff07232fb1b55a505a9e9f6c0ff3-Paper-Conference.pdf.

Potsawee Manakul, Adian Liusie, and Mark Gales. SelfCheckGPT: Zero-resource black-box hallucination detection for generative large language models. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 9004–9017, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.557. URL https://aclanthology.org/2023.emnlp-main.557/.

Artem Maryanskyy. When agents disagree: The selection bottleneck in multi-agent LLM pipelines. *Applied Sciences*, 16(10):4914, 2026. arXiv:2603.20324.

Julian Michael, Salsabila Mahdi, David Rein, Jackson Petty, Julien Dirani, Vishakh Padmakumar, and Samuel R Bowman. Debate helps supervise unreliable experts. *arXiv preprint arXiv:2311.08702*, 2023.

Alicia Parrish, Harsh Trivedi, Nikita Nangia, Vishakh Padmakumar, Jason Phang, Amanpreet Singh Saimbhi, and Samuel R Bowman. Two-turn debate doesn't help humans answer hard reading comprehension questions. *arXiv preprint arXiv:2210.10860*, 2022.

Andrea Pugnana, Lorenzo Perini, Jesse Davis, and Salvatore Ruggieri. Deep neural network benchmarks for selective classification. *Journal of Data-centric Machine Learning Research*, 2024. ISSN XXXX-XXXX. URL https://openreview.net/forum?id=xDPzHbtAEs. Reproducibility Certification.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. GPQA: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=Ti67584b98.

William Saunders, Catherine Yeh, Jeff Wu, Steven Bills, Long Ouyang, Jonathan Ward, and Jan Leike. Self-critiquing models for assisting human evaluators. *arXiv preprint arXiv:2206.05802*, 2022.

Adi Shamir. Ip = pspace. *J. ACM*, 39(4):869–877, October 1992. ISSN 0004-5411. doi: 10.1145/146585.146609. URL https://doi.org/10.1145/146585.146609.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik R Narasimhan, and Shunyu Yao. Reflexion: language agents with verbal reinforcement learning. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=vAElhFcKW6.

Katherine Tian, Eric Mitchell, Allan Zhou, Archit Sharma, Rafael Rafailov, Huaxiu Yao, Chelsea Finn, and Christopher Manning. Just ask for calibration: Strategies for eliciting calibrated confidence scores from language models fine-tuned with human feedback. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 5433–5442, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.330. URL https://aclanthology.org/2023.emnlp-main.330/.

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

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 24824–24837. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/9d5609613524ecf4f15af0f7b31abca4-Paper-Conference.pdf.

Yixuan Weng, Minjun Zhu, Fei Xia, Bin Li, Shizhu He, Shengping Liu, Bin Sun, Kang Liu, and Jun Zhao. Large language models are better reasoners with self-verification. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 2550–2575, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.167. URL https://aclanthology.org/2023.findings-emnlp.167/.

Miao Xiong, Zhiyuan Hu, Xinyang Lu, YIFEI LI, Jie Fu, Junxian He, and Bryan Hooi. Can LLMs express their uncertainty? an empirical evaluation of confidence elicitation in LLMs. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=gjeQKFxFpZ.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 11809–11822. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/271db9922b8d1f4dd7aaef84ed5ac703-Paper-Conference.pdf.

Baotong Zhang and João Sedoc. Human-llm deliberation as an interactive zero-knowledge proof protocol. In *ICIS 2025 Proceedings*, number 28, 2025. URL https://aisel.aisnet.org/icis2025/gen_ai/gen_ai/28.

Zhi Zhou, Yuhao Tan, Zenan Li, Yuan Yao, Lan-Zhe Guo, Yu-Feng Li, and Xiaoxing Ma. A theoretical study on bridging internal probability and self-consistency for LLM reasoning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2026. URL https://openreview.net/forum?id=E0PaeSszLz.

A Technical appendices and supplementary material

A.1 Further Related Work

A.1.1 Selective Prediction, Abstention, and Calibration

Selective prediction studies systems that may abstain on uncertain inputs in order to improve accuracy on the accepted subset. The classical reject-option view formalizes a tradeoff between error and rejection rate (Chow, 1970). Modern selective classification casts this as a risk–coverage problem: a model should minimize error on the covered region while abstaining elsewhere (Geifman & El-Yaniv, 2017; Pugnana et al., 2024). Subsequent work trains selective neural networks end-to-end with an integrated reject option (Geifman & El-Yaniv, 2019), and extends selective prediction to question answering under distribution shift (Kamath et al., 2020). This literature provides the evaluation lens for our work: the relevant object is not only full-coverage accuracy, but the precision achieved at a given coverage level.

Calibration asks whether a model's confidence corresponds to empirical correctness. Neural networks are often miscalibrated (Guo et al., 2017), and generative language models introduce additional calibration difficulties because probability mass is distributed over many possible completions (Jiang et al., 2021). In the LLM setting, several papers ask whether models "know what they know" (Kadavath et al., 2022; Lin et al., 2022; Tian et al., 2023; Xiong et al., 2024). Other approaches estimate reliability from disagreement among generations. SelfCheckGPT detects hallucination by checking whether sampled responses remain factually consistent (Manakul et al., 2023).

These methods are close in spirit to PVD because they treat inconsistency as evidence of unreliability. The distinction is that PVD does not rely only on independent samples or scalar confidence. Instead, the confidence signal is procedural: a prover decomposes and defends an answer, while a verifier searches for weaknesses and issues an explicit Accept, Challenge, or Reject verdict. Thus, the selector is not a thresholded probability, but the outcome of a structured verification dialogue.

A.1.2 Inference-Time Reasoning and Self-Aggregation

A large body of work improves LLM performance by spending additional computation at inference time. Chain-of-thought prompting elicits intermediate reasoning steps and improves accuracy on multi-step reasoning tasks (Wei et al., 2022). Self-Consistency samples multiple independent reasoning paths and aggregates by majority vote (Wang et al., 2023). Universal Self-Consistency extends this idea to free-form outputs by using an LLM to select the most consistent answer among sampled candidates (Chen et al., 2024). Reflexion, Self-Refine, and Self-Enhanced Test-Time Scaling (SETS) use verbal feedback loops to critique and revise model outputs (Shinn et al., 2023; Madaan et al., 2023; Chen et al., 2025), while Tree of Thoughts structures inference as search over possible reasoning states (Yao et al., 2023).

Verification-oriented inference-time methods are also relevant. Chain-of-Verification asks models to generate and answer verification questions about their own outputs (Dhuliawala et al., 2024), and self-verification methods ask models to check or re-derive candidate answers before selecting a final output (Weng et al., 2023). Recent theoretical work studies sampling-based test-time scaling through the lens of confidence estimation, relating self-consistency and internal probability estimates and proposing hybrid approaches that reduce sampling cost (Zhou et al., 2026).

PVD differs from these methods in objective. Most inference-time reasoning methods optimize full-coverage answer accuracy; any reliability signal is a byproduct of agreement, convergence, or

successful refinement. PVD instead makes the reliability signal an explicit output of the protocol. The central empirical question is whether the verifier's verdict defines a high-precision accepted subset at useful coverage and cost. This makes PVD closer to selective prediction than to ordinary test-time ensembling: the protocol is evaluated not only by the answer it returns, but also by whether it knows when not to report.

A.1.3 Verification, Critique, Judges, and Debate

A separate literature uses verifiers or critics to evaluate generated reasoning. Cobbe et al. (2021) train verifiers to score candidate solutions to math word problems, showing that learned verification can improve sample efficiency. Lightman et al. (2024) extend this idea to process supervision, where intermediate reasoning steps are scored rather than only final answers. Saunders et al. (2022) study self-critique as a way to help humans identify flaws in model outputs.

PVD also uses a verifier, but not as a static reward model, reranker, or post-hoc judge. The verifier actively participates in the dialogue: it examines the prover's decomposition, challenges weak or unsupported subclaims, and terminates with a structured verdict. This distinction matters because the verdict is not merely an evaluation score; it is the mechanism by which the system chooses between reporting and abstaining.

Multi-agent debate is another closely related direction. Debate methods cast inference as an exchange among multiple agents, often with answers revised after peer arguments (Du et al., 2024). Some work finds that debate can help weaker or non-expert judges adjudicate difficult questions (Khan et al., 2024; Michael et al., 2023), while other work finds that short debate protocols do not reliably improve judge accuracy on hard reading comprehension tasks (Parrish et al., 2022). Recent work also studies how to encourage diverse perspectives and avoid premature convergence in multi-agent settings (Liang et al., 2024).

Our protocol is deliberately asymmetric. In debate, agents usually argue toward a shared answer or persuade a judge among competing claims. In PVD, the prover commits to a candidate answer and the verifier's role is exclusively skeptical: to identify whether the argument is sufficient, incomplete, or wrong. This asymmetry sacrifices some collaborative refinement, but yields a cleaner selective-prediction signal: accepted unchanged answers are treated as a high-confidence subset, while challenged or rejected answers naturally induce abstention.

A.2 Implementation Details

A.2.1 Baselines

For the primary Sonnet GPQA comparison, baselines use Claude Sonnet 4.6 to control for model capability. Self-Consistency and USC use $k{=}8$; Debate uses 3 agents and 2 rounds; Reflexion uses a maximum of 5 trials. Additional GPT-5.4 and Gemini 3.1 Pro rows are included as robustness comparisons where matching external or direct baselines are available.

A.2.2 Statistical Significance

Table 3 reports the results on GPQA Diamond ($n{=}198$) with 95% confidence intervals and a formal test of the selection gap. Accuracy, HC-Cov, and HC-Prec use Wilson score intervals; the Gap (HC-Prec minus non-HC accuracy) uses a percentile bootstrap over $10{,}000$ resamples. We additionally test each gap with a two-sided Fisher's exact test of H_0: the high-confidence and complement subsets are equally accurate, on the 2×2 table of (HC vs. non-HC) $\times$ (correct vs. incorrect). Fisher's exact test is used rather than a normal-approximation z-test because some complements are small (e.g. Debate leaves $n{<}10$ in the non-HC set). The two procedures agree: every PVD configuration has a Gap CI bounded well above zero and $p < 0.001$, while Reflexion's CI brackets zero and its gap is not significant ($p{=}0.72$) — consistent with stability being a far weaker selective-prediction signal than argument defensibility.

A.3 Positioning Relative to Interactive-Proof and Oversight Work

The closest conceptual comparison is Neural Interactive Proofs (Hammond & Adam-Day, 2025). That work studies neural prover-verifier protocols as learned interaction systems, develops a unifying

Table 3: GPQA Diamond (n=198) with 95% confidence intervals and a gap significance test. **Acc**, **HC-Cov**, and **HC-Prec** use Wilson score intervals; **Gap** (HC-Prec minus non-HC accuracy) uses a percentile bootstrap (10,000 resamples). **Gap** p is a two-sided Fisher's exact test of H_0: HC and non-HC subsets are equally accurate. A Gap CI excluding 0 (equivalently $p < 0.05$) indicates the high-confidence subset is significantly more accurate than its complement. *: SC with extended thinking (Epoch AI benchmark). All PVD rows use the strict verifier prompt (the PVD† prompt of Table 1); see "Two evaluation passes" in Section 4.2 before comparing the two tables.

Run	Acc	HC-Cov	HC-Prec	Gap (95% CI)	Gap p
PVD (standard)	74.2 [67.7, 79.8]	54.5 [47.6, 61.3]	93.5 [87.2, 96.8]	+42.4 [+31.0, +53.5]	<0.001
PVD (min-1)	76.3 [69.9, 81.7]	51.0 [44.1, 57.9]	91.1 [83.9, 95.2]	+30.3 [+18.9, +41.5]	<0.001
PVD-same	80.3 [74.2, 85.2]	63.1 [56.2, 69.5]	94.4 [88.9, 97.3]	+38.2 [+26.0, +50.4]	<0.001
PVD+retry	79.8 [73.7, 84.8]	79.8 [73.7, 84.8]	88.6 [82.7, 92.7]	+43.6 [+27.2, +60.0]	<0.001
Debate (3×2)	83.3 [77.5, 87.9]	95.5 [91.6, 97.6]	85.7 [80.0, 90.0]	+52.4 [+17.4, +85.6]	<0.001
Reflexion	82.3 [76.4, 87.0]	92.9 [88.5, 95.7]	82.6 [76.5, 87.4]	+4.0 [-17.7, +29.1]	0.717
SC (k=8)*	87.4 [82.0, 91.3]	81.3 [75.3, 86.1]	96.3 [92.1, 98.3]	+47.6 [+30.9, +64.1]	<0.001
USC (k=8)	85.4 [79.8, 89.6]	81.3 [75.3, 86.1]	96.3 [92.1, 98.3]	+58.4 [+42.2, +74.1]	<0.001
Single context play	75.8 [69.3, 81.2]	59.6 [52.6, 66.2]	83.1 [75.3, 88.8]	+18.1 [+5.7, +30.7]	0.004

theoretical framework, and evaluates trained protocols on tasks with verifiable structure. Human-LLM interactive proof settings provide another related perspective, focusing on the conditions under which a human verifier can adjudicate LLM-generated arguments (Zhang & Sedoc, 2025).

Closest in experimental setting is scalable oversight with weak judges (Kenton et al., 2024), which studies whether a weaker LLM judge, interacting with stronger models under debate and consultancy protocols, can arrive at correct answers. We share the capability-asymmetric structure — a weaker verifier adjudicating a stronger prover — but ask a different question. Oversight work measures whether the judge reaches the *right answer*; we ask whether the interaction's terminal verdict *partitions* the answer set into a reliable reported subset and a withheld complement, and we characterize the regime in which that partition fails. Our contribution is therefore not the prover-verifier interaction itself, whose lineage runs from Goldwasser et al. (1989) through Hammond & Adam-Day (2025), Amit et al. (2025), and Kenton et al. (2024), but the use of unchanged acceptance (ANC) as a confidence signal, its empirical coverage-precision characterization, and the identification of its failure regime. Prover-verifier games (Kirchner et al., 2024) are the closest training-time analogue of our sub-claim decomposition: they optimize a prover for *legibility* — checkability by a smaller verifier — and measure the accuracy cost of that constraint. PVD asks whether frozen models already expose enough checkable structure for the same capability asymmetry to yield a selection signal, without paying that tax through training.

A.4 Algorithm

Algorithm 1 is the pseudocode for the protocol of Section 3.

Retry and majority vote. Classical IP asks a single question: can $\mathcal{P}$ convince $\mathcal{V}$ of $s \in L$? PVD adds a retry mechanism. After each rejection, the protocol initiates a fresh attempt with adversarial context from prior failures, accumulating evidence across up to K trajectories. ANC is declared only if *some* attempt terminates in ACCEPT with no revision; otherwise majority vote aggregates over all K final answers. This trades additional inference calls for robustness to single bad deliberation trajectories, and decouples the high-confidence ANC subset from the majority-vote fallback.

A.5 Evaluation Details

Inference cost. We measure computational cost in **LLM calls per question**. Reference points: a single model call (direct chain-of-thought, or the single context play ablation) costs 1 call; Self-Consistency uses k=8 samples; Debate with 3 agents and 2 rounds uses agents $\times$ (rounds + 1) = 9 calls (one initial statement plus one response per round per agent); Reflexion is bounded by 5 trials and averages ≈2.5 calls per question in our runs. A single PVD attempt that terminates on verifier turn t uses $2t$ calls: one initial prover statement, t verifier turns, and $t-1$ challenge-reply pairs between them. The minimum is 2 calls when standard PVD accepts on the first verifier turn; PVD†,

Algorithm 1 Prover-Verifier Deliberation (PVD)

Require: Question q, choices $\mathcal{A}$, prover $\mathcal{P}$, verifier $\mathcal{V}$, fatigue limit T, max attempts K
Ensure: Final answer $\hat{a}^*$, outcome $\in \{\text{ANC}, \text{MAJVOTE}\}$

$\log \leftarrow [\,]$
for $k = 1, \ldots, K$ **do**
  $\text{ctx} \leftarrow \text{FAILURESUMMARY}(\log)$ ▷ fixed template, no model call: per prior attempt, its initial answer, changes, outcome, last verifier challenge; empty on first attempt
  $(\hat{a},\ \text{stmt},\ \mathbf{c},\ \text{rsn}) \leftarrow \mathcal{P}(q,\ \mathcal{A},\ \text{ctx})$ ▷ initial proof; **c** = sub-claims
  $h_{\mathcal{P}} \leftarrow [\,(q, \hat{a}, \text{stmt}, \mathbf{c}, \text{rsn})\,]$; $h_{\mathcal{V}} \leftarrow [\,(q, \hat{a}, \text{stmt}, \mathbf{c}, \text{rsn})\,]$
  $\Delta \leftarrow 0$ ▷ answer-change counter
  for $t = 1, \ldots, T$ **do**
    $(\text{verd},\ \text{rsn}_{\mathcal{V}},\ c_t,\ \tilde{c}_t) \leftarrow \mathcal{V}(h_{\mathcal{V}})$ ▷ c_t: challenge text; $\tilde{c}_t$: targeted sub-claim
    Append $\mathcal{V}$'s response to $h_{\mathcal{V}}$
    if $\text{verd} = \text{ACCEPT}$ **then**
      $\log.\text{append}((\hat{a},\ \text{ANC if } \Delta{=}0 \text{ else ACCCHG},\ t))$
      if $\Delta = 0$ **then return** $\hat{a}$, ANC
      end if ▷ high-confidence: exit immediately
      break ▷ answer changed: log and retry
    else if $\text{verd} = \text{REJECT}$ **or** $t = T$ **then**
      $\log.\text{append}((\hat{a},\ \text{REJECT},\ t))$; **break**
    else ▷ CHALLENGE
      Append $(\tilde{c}_t,\ c_t)$ to $h_{\mathcal{P}}$ ▷ prover sees the challenge it must answer
      $(\hat{a}',\ \text{stmt}',\ \mathbf{c}',\ \text{rsn}') \leftarrow \mathcal{P}(h_{\mathcal{P}})$ ▷ targeted defence
      if $\hat{a}' \neq \hat{a}$ **then** Δ += 1; $\hat{a} \leftarrow \hat{a}'$
      end if
      Append $\mathcal{P}$'s reply to $h_{\mathcal{P}}$, $h_{\mathcal{V}}$
    end if
  end for
end for
return $\text{MAJORITYVOTE}(\log)$, MAJVOTE ▷ plurality over every attempt's final answer, ACCCHG included; with $K{=}1$, the lone attempt's answer

which forces a minimum of one challenge before any ACCEPT, costs at least 4 calls. Empirically, single-attempt PVD averages 3–6 calls per question across our configurations; the retry protocol ($K{=}6$) averages ≈29. Per-method means are reported in the **Calls** column of Table 1. Token-level costs are logged from provider API responses and converted to USD using provider list prices at time of evaluation.

Two evaluation passes. Table 1 reports our original runs across all prover-verifier configurations. To support the statistical and sensitivity analyses of Section 6, we subsequently re-executed the Sonnet 4.6 suite — the PVD variants and every Sonnet-based baseline — with full per-call transcript and token logging; Table 3, Figure 1, and Figures 2–3 use this second pass, and every confidence interval and significance test in the paper is computed on it. The two passes are not identical protocols, and the difference must be read before their numbers are compared. In the first pass, PVD uses the lenient verifier prompt of Appendix A.13 ("accept if the reasoning is logically sound") and PVD† the strict one ("default to challenge"), with the strictness carried by the prompt alone. In the second pass the strict prompt is the *base* configuration of every PVD variant, lightly shortened, with a conciseness instruction added to the prover and output-token limits halved (prover $4000 \rightarrow 2000$, verifier $1000 \rightarrow 600$): "PVD (standard)" there is the strict prompt with acceptance permitted on the first round, and "PVD-min1" additionally enforces at least one challenge in code. The second pass's standard configuration is therefore the analogue of Table 1's PVD†, not of its PVD — which is why its native operating point is stricter (54.5% coverage at 93.5% precision, against 77% at 84.2% for first-pass PVD and 65% at 89.9% for first-pass PVD†). What the passes share is the ranking of questions: the re-executed risk-coverage curve passes through 84.3% precision at 77% coverage, matching first-pass PVD's operating point. Two baselines also differ by construction: first-pass SC and USC draw $k{=}8$ standard-API samples at temperature 1.0, whereas the second pass imports Epoch AI's extended-thinking SC samples (marked *) and runs the USC selector over them, which is

why second-pass SC precision is higher and why the cost analysis of Section 6 treats those samples as more expensive per call.

Decoding and reproducibility. All PVD calls use the provider's default sampling (temperature 1.0 on the Anthropic API) with extended thinking disabled. The Anthropic models are `claude-sonnet-4-6` and `claude-haiku-4-5-20251001`, the latter a dated snapshot; GPT and Gemini models were addressed by the provider's model name at the time of the run and carry no snapshot identifier; the Gemini runs use temperature 0.0, and the OpenAI reasoning models used here do not expose a temperature parameter. First-pass self-consistency samples k=8 chains at temperature 1.0; Debate uses temperature 1.0 for the agents' initial answers and 0.0 for debate rounds; Reflexion runs up to five trials (mean 2.5 calls) with a same-temperature self-consistency recheck as its failure signal. Question order is a fixed shuffle (seed 42) of the 198 GPQA Diamond items. Second-pass result files record the git commit, configuration, model identifiers, and per-call token counts in a header record; the second pass ran on 16 May 2026. Because the hosted models are not versioned beyond these identifiers, exact replication of sampled outputs is not guaranteed, and our reproducibility claim is for the protocol and analysis code rather than for token-level outputs.

A.6 Additional Result Tables

This appendix collects the result tables referenced from Sections 5 and 6: the per-domain GPQA Diamond breakdown, the HLE results by domain, and the verifier-choice comparison.

A note on overall accuracy. The **Acc** column reports the prover's final answer on *all* questions regardless of verdict; without a remediation step for non-ANC questions, PVD's full-population accuracy matches the single context play ablation (76.8–79.8% vs. 78.8% for Sonnet 4.6). This is by design: PVD is a selective-prediction protocol, not an accuracy booster. Its output is a report/abstain signal, and a non-ANC verdict is the natural trigger for escalation to a more capable model, more compute, or human review — a second stage we do not implement here; the precision gap (+29–32pp) quantifies the value of the signal.

Table 4 breaks the ANC signal down by domain for three representative PVD configurations. Biology is the smallest domain (n=19) and the numbers should be interpreted accordingly.

Table 4: GPQA Diamond ANC results by domain. **HC-Cov** = $\Pr[\text{ANC}]$: fraction of domain questions assigned ANC. **HC-Prec** = $\Pr[\text{correct} \mid \text{ANC}]$: accuracy on ANC subset. **Gap**: HC-Prec minus non-ANC accuracy.

		Sonnet + Haiku[†]			**GPT-5.4 + GPT-5.4-mini**			**Gemini Pro + Flash-Lite**		
Domain	n	HC-Cov	HC-Prec	Gap	HC-Cov	HC-Prec	Gap	HC-Cov	HC-Prec	Gap
Chemistry	93	49%	80%	+25.1	16%	100%	+46.2	45%	100%	+7.8
Physics	86	83%	96%	+15.8	70%	98%	+9.9	66%	100%	+6.9
Biology	19	63%	92%	+34.5	53%	90%	+23.3	68%	77%	+10.3

Table 5: Humanity's Last Exam results (GPT-5.5 prover, Gemini 3.1 Pro verifier, n=513, T=12, K=1). Rows sorted by n. **HC-Cov** = $\Pr[\text{ANC}]$; **HC-Prec** = $\Pr[\text{correct} \mid \text{ANC}]$; **Gap**: HC-Prec minus non-ANC accuracy.

Domain	n	**Overall Acc**	**HC-Cov**	**HC-Prec**	**Non-ANC Acc**	**Gap**
Biology / Medicine	147	41%	46%	50.0%	32.9%	+17.1
Math	89	61%	67%	76.7%	27.6%	+49.1
Humanities / Soc. Sci.	79	51%	53%	61.9%	37.8%	+24.1
Computer Science / AI	66	52%	53%	68.6%	32.3%	+36.3
Other	44	32%	39%	52.9%	18.5%	+34.4
Physics	37	27%	57%	33.3%	18.8%	+14.6
Chemistry	26	46%	35%	77.8%	29.4%	+48.4
Engineering	25	40%	64%	31.2%	55.6%	−24.3
All	**513**	**45.6%**	**52%**	**59.0%**	**31.0%**	**+27.9**

Table 6: Effect of verifier choice on ANC calibration, holding prover fixed. **Avg R**: mean challenge rounds. †: challenge-first prompt. §: multi-attempt retry configuration (K=6); its Avg R is cumulative across attempts and exceeds the single-attempt fatigue limit T=12. Attempt 1 of that run alone (a single attempt under the same prompts) gives HC-Cov 49%, HC-Prec 97.9% and a gap of +17.7pp, so retrying adds coverage without widening the gap. *Self*: same model as prover (single context play ablation).

Prover	Verifier	HC-Cov	HC-Prec	Non-ANC Acc	Gap	Avg R
Sonnet 4.6 prover						
Sonnet 4.6	*Self*	63%	83.9%	70.3%	+13.6	2.1
Sonnet 4.6	Haiku	77%	84.2%	52.2%	+32.0	1.7
Sonnet 4.6	Haiku†	65%	89.9%	60.9%	+29.1	3.0
Gemini 3.1 Pro prover						
Gemini Pro	Flash-Lite	57%	97.3%	90.7%	+6.6	1.9
Gemini Pro	GPT-5.5-pro§	75%	97.3%	80.0%	+17.3	14.3

A related pattern appears within GPQA when the verifier is varied with the prover held fixed (Table 6), with a caveat we state plainly and can now resolve. Holding Gemini 3.1 Pro as prover, the GPT-5.5-pro row is the multi-attempt retry configuration (K=6; its 14.3 mean rounds are cumulative across attempts, which is why they exceed the single-attempt fatigue limit T=12), whereas the Flash-Lite row is a single attempt (K=1), so the two rows differ in test-time compute (roughly seven-fold) as well as in verifier. What they share is HC-Prec (97.3% in both); the difference lies entirely in the non-ANC complement, whose accuracy drops from 90.7% to 80.0%. The retry file itself separates the two factors, because its first attempt is a complete single-attempt deliberation under the same prompts: taken alone, attempt 1 gives HC-Prec 97.9% at 49.0% coverage against 80.2% on the complement (gap +17.7pp, 3.95 mean rounds, n=198; five first-attempt verdicts are API failures that the legacy runner recorded as rejections, and excluding them gives +16.7pp at 50.3% coverage). Retrying therefore raises coverage from 49% to 75% at unchanged precision and leaves the gap where the first attempt put it, so the widening from +6.6pp to +17.3pp relative to Flash-Lite is attributable to the verifier, not to the additional attempts. What remains unmatched between the two verifier rows is prompt wording (the retry script uses a shorter variant of the prover and verifier prompts); a single-attempt run of the pairing under the Flash-Lite row's exact prompts is the fully matched comparison and is left for future work.

A.7 Additional Analyses of the Re-executed Sonnet Suite

The analyses below use the second evaluation pass (Appendix A.5) and are summarized in Section 6.1.

Does separation matter? A single context play variant (prompt in A.13) has one model play both roles within one call. Without a required first challenge it always accepts; with that requirement HC-Prec remains 84%, so the structured sub-claim decomposition contributes independently of cross-model independence, but the Accept/Reject gap narrows by ∼15pp (from +29pp to +14pp), consistent with a weaker Effective Verifier that cannot challenge claims it generated itself.

Defensibility versus stability. A related but weaker proxy for confidence is *stability*: does the model give the same answer when re-sampled? Reflexion uses stability across iterations as its high-confidence signal (Table 1). Yet on GPQA Diamond stability turns out to be only marginally informative: Reflexion flags 93% of questions as stable, but the precision on that subset (82.6%) is essentially indistinguishable from the accuracy on the unstable complement. The reason is structural. At standard sampling temperatures, Sonnet 4.6 produces the same multiple-choice letter on most pairs of independent samples, including for many questions it answers incorrectly; consistency in the output channel does not depend on whether the reasoning chain is sound. PVD's ANC verdict requires something strictly stronger: the prover's sub-claim decomposition must survive an adversarial challenge round without revision. The empirical gap between Reflexion's ∼+4pp signal and PVD's +29–35pp ANC gap supports our thesis that argument defensibility, not sample agreement, is what carries the load for selective prediction.

Self-verification with explicit sub-claim checking. A natural ablation question is whether PVD's edge comes from interactive challenge or simply from forcing the prover to enumerate sub-claims that can be checked. We answer this by running a self-verification baseline adapted from Weng et al. (2023): a Sonnet 4.6 forward pass emits an answer letter plus 3–5 atomic sub-claims, then each sub-claim is independently verified in parallel by another Sonnet 4.6 call (no iterative challenge, no separate verifier model). HC is reached when every sub-claim returns VERIFIED. The result on GPQA Diamond (n=198, $\sim$5 calls/question, \$6.97 measured): HC-Cov 22.7% , HC-Prec 93.3%, gap $+37.1$pp. The HC precision is statistically indistinguishable from PVD's 93.5% HC-Prec at less than half the dollar cost, so a single round of self-verification captures most of the *precision* from PVD's defensibility signal. What it doesn't capture is *coverage*: the self-verifier flags only 22.7% of questions as fully verified versus 54.5% for PVD, and overall accuracy drops by $\sim$10pp (the forced sub-claim decomposition appears to hurt single-pass answer quality). PVD's gain over self-verification is the same precision with much higher coverage.

Calibration metrics: ECE, AURC, and a conformal baseline. For alignment with the selective-prediction and calibration literature (Geifman & El-Yaniv, 2017; Guo et al., 2017), we additionally report Expected Calibration Error (ECE, 10 bins, min-max normalised score) and Area Under the Risk-Coverage curve (AURC, integrated over coverage). The per-method confidence score is the method's natural ranking signal: top-vote share for SC/USC and Debate, outcome_rank $-$ rounds for PVD, and stable-and-unchanged for Reflexion. On GPQA Diamond, AURC values are shown in Figure 3. The two PVD configurations that admit additional information (retry; self-verifier-as-prover) come within $\sim$2pp of the SC/USC AURC floor at roughly an order of magnitude less measured dollar cost. Reflexion is the worst selective predictor, consistent with the gap CI in Table 3 bracketing zero. ECE is shaped by score discretization (PVD has four ordinal outcome buckets versus SC's continuous vote share); the order of methods by ECE largely mirrors AURC but the magnitudes are not directly comparable. As a conformal-predictor baseline, leave-one-out split conformal over the k=8 SC vote shares at target marginal coverage 90% produces empirical coverage 90.4% with average prediction set size 1.07; singleton predictions (93.4% of questions) achieve 90.3% accuracy. The conformal route gives a coverage guarantee at the cost of holding the expensive SC samples; PVD's ANC achieves comparable singleton accuracy (84–94% HC-Prec) without the calibration set and at one-tenth the dollar cost.

Matched-budget Self-Consistency. The headline cost comparison between PVD and SC uses each method's natural operating point (k=8 for SC, the protocol's own call count for PVD). A fairer test is to fix the call budget and ask which method delivers a larger selection gap. We subsample the existing k=8 SC candidate set to $k \in \{1, 2, 4, 6, 7, 8\}$. At the same number of LLM calls per question, SC@4 (extended-thinking samples) achieves a $+52.8$pp gap with 86% HC-Coverage, edging PVD ($+42.4$pp at 55%) and PVD-same ($+38.2$pp at 63%) at the same call budget. At a fixed dollar budget of $\sim$\$10–\$25 on n=198, *every* PVD configuration dominates SC@k on the gap metric: SC needs \$72 to reach a $+53$pp gap (k=4) while PVD reaches $+42$pp for \$9.16. This is the honest statement of PVD's selective-prediction edge: not that an independent verifier extracts strictly more information than k=8 samples of the prover, but that doing so with a cheap verifier and no extended thinking is roughly an order of magnitude cheaper per pp of gap (Figure 2).

Sensitivity to the fatigue limit T and the retry cap K. Across all multi-call configurations on GPQA Diamond, HC-Coverage and the precision gap *saturate by* $T \approx 6$ and remain flat through $T = 12$. For PVD_RETRY the gap as a function of retry cap is non-monotone, peaking at K=2 at $+54.7$pp with 63% coverage and falling to $+43.6$pp at K=5 with 80% coverage, and additional retries trade precision for coverage roughly linearly. Risk-coverage curves (sorting questions by an outcome-and-rounds confidence score and sweeping the inclusion threshold) yield AURC of 7.1% for PVD-retry dominating other methods. The full sensitivity grid and risk-coverage curves are in Figure 3.

Three-tier calibration: ACCEPT-CHANGED as a medium-confidence set. The ANC verdict partitions outcomes into a binary HIGH / non-HIGH split, but the protocol naturally produces a third category in between: questions where the verifier eventually issues ACCEPT *after* the prover revised its answer mid-deliberation (ACCEPT-CHANGED). If reasoning that withstood a challenge round and was updated carries information distinct from reasoning that was never challenged, we should see a strict accuracy ordering HIGH (ACC-NC) $>$ MEDIUM (ACC-CHANGED) $>$ LOW (REJECT $\cup$ FATIGUE). Empirically, this is what the data show for every multi-call PVD configuration on GPQA Diamond: PVD achieves 93.5% / 66.7% / 48.7% (n=108/12/78); PVD-same reaches 94.4% / 61.3% / 52.4%;

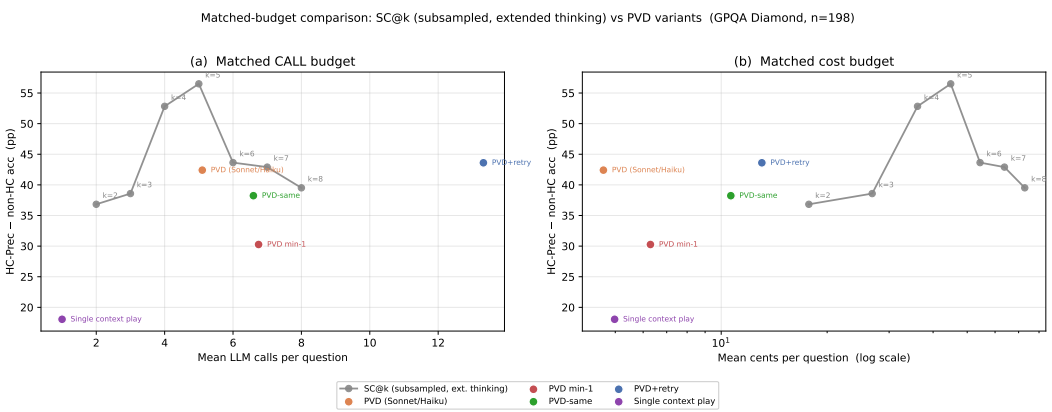

Figure 2: Matched-budget comparison of SC@k (extended-thinking samples subsampled from the Epoch AI archive) against the PVD operating points on GPQA Diamond (n=198). (a) At the same number of LLM calls per question, SC@4 beats every single-attempt PVD configuration on the gap. (b) At the same measured dollar cost, PVD configurations dominate SC@k by an order of magnitude: standard PVD reaches +42pp for \$9.16 while the cheapest SC point (\$35 at k=2) reaches +37pp. The trade-off is real: SC carries more compute per call (extended thinking), PVD carries cheap compute per call (Haiku verifier).

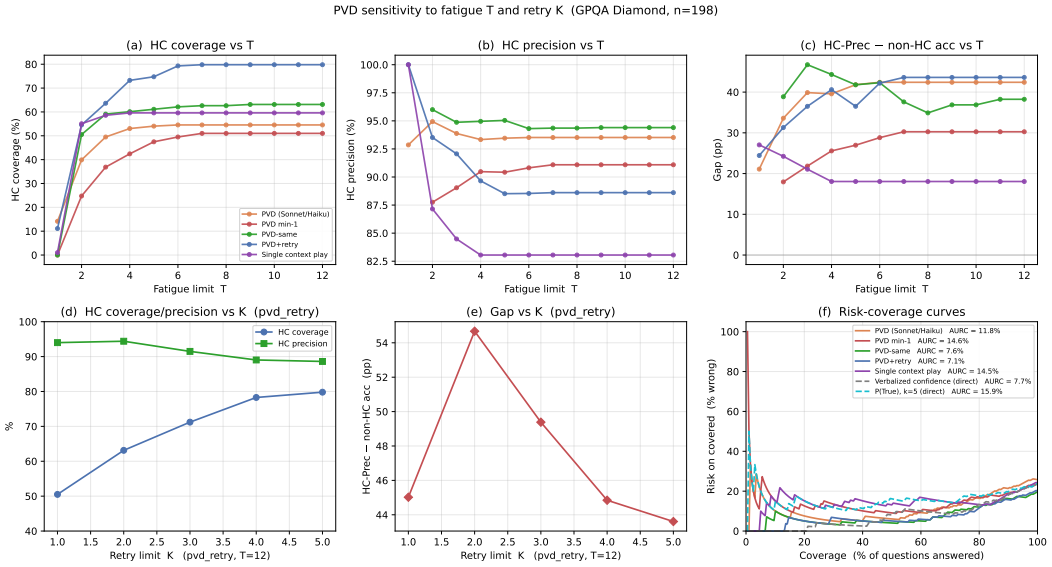

Figure 3: Sensitivity of the PVD selection signal on GPQA Diamond (n=198) to the fatigue limit T and the retry cap K, plus risk-coverage curves. **Top row, T-sweep at maximal K:** HC-Coverage (a), HC-Precision (b), and the gap (c) versus T. **Bottom row:** (d) HC-Coverage and HC-Precision versus retry cap K for PVD-Retry (at $T = 12$); coverage grows monotonically with K, precision decays mildly. (e) The gap is non-monotone in K, peaking at $K = 2$ (+54.7pp). (f) Risk-coverage curves with the per-question confidence score outcome_rank − rounds; lower AURC = better selective predictor.

PVD-min1 91.1% / 86.7% / 56.1% (n=101/15/82) — strictly monotone in each case, with non-overlapping Wilson 95% CIs between HIGH and LOW. The accepted-with-change tier is small (12–15 questions), so its accuracy is imprecisely estimated and the HIGH > MEDIUM step is not individually significant; what the data support is that withholding revised answers from the reported

subset removes a set whose accuracy sits between the accepted-unchanged and rejected sets, not that it discards reliable self-corrections wholesale. A deployed system can therefore use the three-tier verdict as a calibrated escalation ladder (report on HIGH, additional verification on MEDIUM, abstain or escalate on LOW) without training a separate calibrator. The one exception is the single context play ablation, where the order *inverts*: ACC-CHANGED (95.7%) outperforms ACC-NC (83.1%), suggesting that within a single inference pass an answer that survived self-critique is stronger than one the model never doubted — an effect that disappears once a separate verifier model takes over the challenge role.

Self-reported confidence baselines. The sharpest test of whether deliberation adds anything a single call cannot report is to elicit a confidence from the *same* model about its *own* direct answer. We ran two such baselines on the direct single-call answers of Table 1 (protocol 'confidence' in the released code): a verbalized confidence, in which the model is shown the question, its answer, and its reasoning and states an integer confidence from 0 to 100 (Lin et al., 2022; Tian et al., 2023); and P(True) (Kadavath et al., 2022), in which the model is asked whether the proposed answer is correct ("(A) True / (B) False") and, because the APIs expose no token log-probabilities, the probability is estimated as the fraction of $k{=}5$ independent samples at temperature 1.0 answering True. Both calls are format-constrained (a forced tool call on the Anthropic API, a response schema on the Gemini API) so that the reply is a verdict and not a second derivation. Table 7 ranks the direct answers by each signal and reports the precision of the top-scored set at PVD's native coverage (tie-inclusive), together with AURC.

For Sonnet 4.6 the verbalized signal is strong: AURC 7.7%, against 9.5–14.6% for the outcome-and-rounds score of the single-attempt PVD configurations and 7.1% for PVD+retry, and at PVD's coverage it matches or exceeds ANC precision (87.3% vs. 84.2% at 77% coverage; 91.0% vs. 89.9% at 65%; 90.2% at 61.6% coverage vs. the second pass's 93.5% at 54.5%). P(True) is nearly binary — 1.0 on 72% of questions — and ranks poorly (AURC 15.9%). The two signals are not redundant, however: on the aligned PVD† run, ANC and the coverage-matched verbalized set disagree on 21 questions that PVD accepts but the model rates below the confidence cut-off, and on those PVD's answer is correct 71.4% of the time against 47.6% for the direct answer. For Gemini 3.1 Pro (direct accuracy 93.4%) the picture is the same at a higher ceiling: the stated confidence is compressed toward the top of the scale (78% of questions at $\geq$95, P(True) at 1.0 on 90%), yet it ranks the answers well — verbalized AURC 1.8% against 4.3% for the outcome-and-rounds score of the PVD/Flash-Lite run and 5.4% for P(True) — and at PVD's 56.6% coverage the tie-inclusive top set (which extends to 78.3% because of the ties) has precision 97.4%, matching ANC's 97.3% at higher coverage. Direct answers that could not be parsed (four for Sonnet, three derivations cut off at the token limit and one empty API response; one for Gemini, a call that timed out on every attempt) are scored as wrong in Table 1 and ranked last here. We read these results plainly: a single stated confidence is a competitive selective predictor on this benchmark for both provers, and PVD's contribution is the complementary information in a verdict that an adversary must survive rather than a higher precision at a fixed coverage.

A.8 AUDITING THE VERIFIER WITH A STRONGER MODEL

The ANC gap needs ground-truth labels, which is why we call it an evaluation diagnostic. A natural way to relax this is to let a stronger model stand in for the labels, and Table 8 tests two versions of that idea on runs where the truth is known. The cheap version uses the reference model's *own answer* as the label. On GPQA it works: with Gemini 3.1 Pro (93.4% accurate) as the reference, the Sonnet/Haiku gap comes out at $+35.2$pp against a true $+42.4$pp for the second-pass standard configuration and $+32.1$pp against $+29.1$pp for PVD† — the right sign and within a few points, enough to certify a verifier against a threshold with margin. On HLE it fails in the way the Effective Verifier account predicts: with the GPT-5.5 / Gemini 3.1 Pro pairing (45.6% accurate on the 513 items) as the reference, the Sonnet/Haiku gap reads $+20.3$pp where the truth is -7.1pp, because the questions the weak pair accepts are exactly those on which both models give the same answer, and near the capability ceiling agreement is not correctness. A strong model can therefore certify a weaker verifier only on domains where it is itself well inside its effective region, a premise that has to be checked on a labeled slice once; the label requirement shrinks to a calibration step rather than disappearing.

Table 7: Label-free confidence signals against PVD's ANC on GPQA Diamond (n=198). Under each PVD reference row, the two indented rows rank the same prover's *direct* single-call answers by a self-reported confidence and admit the top-scored questions until PVD's coverage is reached (tie-inclusive, so the achieved coverage can exceed the target); **HC-Prec** is the accuracy of that subset and n_{HC} its size. Verbalized confidence is a stated 0–100 confidence (Lin et al., 2022; Tian et al., 2023); P(True) is the fraction of k=5 temperature-1 self-judgments that call the answer correct (Kadavath et al., 2022). **AURC** (area under the risk–coverage curve, lower is better) ranks all questions by the signal; for PVD it uses the outcome-and-rounds score of Figure 3(f). **Acc** is the accuracy of the answers being ranked (the PVD prover's final answers versus the direct call's).

Signal	**Acc**	**HC-Cov**	**HC-Prec**	n_{HC}	**AURC**↓
Sonnet 4.6 as prover					
PVD, Haiku 4.5 verifier (Table 1)	76.8%	77%	84.2%	152	13.7
Verbalized confidence at matched coverage	76.3%	79%	87.3%	157	7.7
P(True), k=5 at matched coverage	76.3%	98%	77.8%	194	15.9
PVD, strict prompt (second pass)	74.2%	55%	93.5%	108	11.8
Verbalized confidence at matched coverage	76.3%	62%	90.2%	122	7.7
P(True), k=5 at matched coverage	76.3%	72%	85.9%	142	15.9
Gemini 3.1 Pro as prover					
PVD, Flash-Lite verifier (Table 1)	94.4%	57%	97.3%	112	4.3
Verbalized confidence at matched coverage	93.4%	78%	97.4%	155	1.8
P(True), k=5 at matched coverage	93.4%	90%	95.0%	179	5.4

The second version asks the stronger model to *grade the argument* rather than solve the question: it reads the finished transcript and judges whether the final answer is correct given the argument and whether each verifier decision (Accept, Challenge, Reject) was well founded, the asymmetric-load task the protocol itself relies on. For the second-pass Sonnet suite graded by Gemini 3.1 Pro (one structured call per question, n=198 each), the graded gap is +37.2pp for standard PVD (true +42.4pp), +30.2pp for PVD-min1 (true +30.3pp) and +33.6pp for PVD-same (true +38.2pp), and the grader's judgment of the final answer agrees with the truth on 94–95% of questions; on GPQA the graded and answer-agreement proxies are nearly indistinguishable because the grader is itself 92–94% accurate. The decision grades add something the gap cannot give: a label-free estimate of the competence term of Section 2.2. The grader judges 60% of Haiku's decisions well founded in the standard run (83% of its Accepts, 54% of its Challenges, 51% of its Rejects) and 55% in PVD-min1, against 74% for Sonnet acting as its own verifier in PVD-same (79%, 75%, 55%): the stronger verifier's challenges are far more often well aimed, while acceptances are reliable for both — the shape the ANC signal exploits, and a statistic that tracks verifier strength without labels. HLE transcripts were not logged in the first evaluation pass, so the grading variant could not be tested where the agreement proxy fails; that experiment needs a logged HLE re-run and is left open.

A.9 Selective Escalation: What a Non-ANC Verdict Buys

Section 5.1 notes that PVD's full-population accuracy is not its objective and that a non-ANC verdict is a trigger for escalation. Table 9 makes that concrete offline: a cheap PVD stage answers the questions it keeps and an expensive single-call expert answers the rest, with cost measured from the logged tokens of both runs. The honest result is that on GPQA Diamond no cascade matches the expert at equal cost. With Opus 4.6 as the expert (84.3% at $0.067 per question on its own), standard Sonnet/Haiku PVD deferring every non-ANC question reaches 82.3% at $0.077 (45% deferred), and PVD-same reaches 83.8% at $0.131; with Gemini 3.1 Pro as the expert (93.4% at $0.032) the same construction reaches 92.4% at $0.061. The arithmetic explains why. A cascade beats the expert only if the cheap stage's kept subset is more accurate than the expert would be on it *and* the cheap stage costs a small fraction of the expert. Here the kept subset (93.5%) clears Opus's overall accuracy, but Opus is right on only 62–72% of the deferred, harder questions, and Sonnet/Haiku PVD costs about 70% of an Opus call. The non-ANC verdict is therefore a routing signal whose value depends on the price gap between the two stages; it does not buy expert accuracy at a discount when the expert is only a few times the price of the cheap pair.

Table 8: Replacing ground-truth labels with a stronger model when auditing a verifier. **A**: the reference model's own answer is used as the label (agreement with the prover's final answer counts as correct); the ANC gap is recomputed under that proxy and shown next to the true gap. **B**: a grader reads the finished transcript and judges whether the final answer is correct given the argument ("arg.") — its own answer gives the agreement proxy ("ans.") for comparison — and whether each verifier decision was correct; **verifier competence** is the fraction of decisions graded correct, overall and by verdict type. Ref. acc is the reference or grader model's own accuracy on the questions.

Weak run / reference	n	**Ref. acc**	**True gap**	**Proxy gap**	**Verifier competence**
A. Answer-agreement proxy: reference model's answer as the label					
GPQA: PVD standard (2nd pass) vs Gemini-direct	198	93.4%	+42.4	+35.2	—
GPQA: PVD min-1 (2nd pass) vs Gemini-direct	198	93.4%	+30.3	+28.2	—
GPQA: PVD+retry (2nd pass) vs Gemini-direct	198	93.4%	+43.6	+45.5	—
GPQA: PVD† (1st pass) vs Gemini-direct	198	93.4%	+29.1	+32.1	—
HLE: Sonnet/Haiku vs GPT-5.5/Gemini pairing	513	45.6%	−7.1	+20.3	—
B. Transcript grading: grader judges the argument and each verifier decision					
PVD min-1 / gemini-3-1-pro-preview	198	91.9%	+30.3	+30.2 (arg.) / +30.2 (ans.)	55% (Acc. 86%, Cha. 49%, Rej
PVD-same / gemini-3-1-pro-preview	198	93.9%	+38.2	+33.6 (arg.) / +33.6 (ans.)	74% (Acc. 79%, Cha. 74%, Rej
PVD (Sonnet/Haiku) / gemini-3-1-pro-preview	198	92.4%	+42.4	+37.2 (arg.) / +36.1 (ans.)	60% (Acc. 83%, Cha. 54%, Rej

A.10 Self-Consistency and PVD: Overlap and a PVD-Consistency Variant

A more aggressive combination—suggested by the structural difference between the signals—would resample the deliberation itself. It differs from our retry protocol (Algorithm 1) in two ways: retry re-attempts only after a failure and conditions each new attempt on a summary of the prior ones, whereas a "PVD-consistency" variant would draw m *independent* deliberations unconditionally and report only when a majority of them reach ANC on the same answer, aggregating as SC aggregates samples. If the protocol is not merely guessing, independent draws would succeed more often on genuinely answerable questions and accumulate rejections on genuinely hard ones. The distinction matters for false acceptance: reporting on the *first* acceptance across repeated attempts compounds the per-attempt slip-through probability — the mechanism behind the non-monotone retry gap in K noted in Section 7 — whereas a majority over independent draws averages it. Any report-on-acceptance scheme still yields an ANC verdict that is optimistic as a calibration estimate, since the reported answer was selected from several; nevertheless, the question of whether structured re-deliberation can recover additional correct answers that single-attempt ANC misses is a natural avenue for future work. Combining PVD with consistency-based sampling is also complementary to remediation workflows, where non-ANC questions trigger more capable models rather than more attempts of the same model.

A.11 Soundness Under a Deceptive Prover: Details

The protocol's value rests on an assumption we can test directly: that an ACCEPT reflects an argument surviving scrutiny rather than a verifier deferring to confident-sounding output. We evaluate this adversarially. The prover (Sonnet 4.6) is given the correct answer and instructed to argue convincingly for a wrong one; the verifier (Haiku 4.5) is unchanged and unaware. In a second, *strategic* condition the prover is additionally told that the verifier is a smaller model and advised to prolong the exchange with elaborate, technically-worded defenses — that is, to exploit the capability asymmetry the protocol depends on. Both conditions run on all of GPQA Diamond (n=198) at fatigue T=12.

The verifier accepts the deception on $12/198$ (6.1%) of questions in the naive condition and $18/198$ (9.1%) in the strategic condition, against 87.2% acceptance of honest correct answers under the same pairing — a roughly 10–$14\times$ separation. Three properties of the failures matter more than the headline rate. No rejection arose from exhausting the round limit (0% fatigue in both conditions): every catch was an outright REJECT, and reached faster than honest acceptance (1.88 and 1.77 mean rounds versus 3.0 when the prover argues truthfully), so deception is easier for the verifier to reject than genuine reasoning is for it to accept. The prover never abandoned the lie under challenge ($0/396$ questions across both conditions), so the low fool rate reflects active rejection rather than the attacker capitulating. And explicitly coaching the prover to exploit the verifier's weaker capacity bought only $+3$pp.

Table 9: Selective escalation on GPQA Diamond (n=198), simulated offline from finished runs: a cheap PVD stage answers the questions it keeps and defers the rest to an expensive single-call expert. **Acc** is the cascade's accuracy, **Deferred** the fraction sent to the expert, **Kept prec.** the accuracy of the cheap stage on the questions it kept, and **$/q** the measured mean cost per question (cheap stage plus the deferred fraction of the expert's cost). "non-ANC" defers every question not accepted without revision; "Reject only" also keeps answers accepted after a revision.

Cheap stage	Deferral	Acc	Deferred	Kept prec.	$/q
Expert: Opus 4.6 direct alone: 84.3% at $0.067/q					
PVD (Sonnet/Haiku)	non-ANC	82.3%	45%	93.5%	$0.077
PVD (Sonnet/Haiku)	Reject only	82.3%	39%	90.8%	$0.073
PVD min-1	non-ANC	81.8%	49%	91.1%	$0.096
PVD min-1	Reject only	81.8%	41%	90.5%	$0.091
PVD-same	non-ANC	83.8%	37%	94.4%	$0.131
PVD-same	Reject only	80.8%	21%	87.8%	$0.121
PVD+retry	non-ANC	83.3%	20%	88.6%	$0.144
PVD+retry	Reject only	83.3%	20%	88.6%	$0.144
Single context play	non-ANC	80.3%	40%	83.1%	$0.077
Single context play	Reject only	80.3%	16%	86.7%	$0.061
Expert: Gemini 3.1 Pro direct alone: 93.4% at $0.032/q					
PVD (Sonnet/Haiku)	non-ANC	92.4%	45%	93.5%	$0.061
PVD (Sonnet/Haiku)	Reject only	90.9%	39%	90.8%	$0.059
PVD min-1	non-ANC	90.4%	49%	91.1%	$0.079
PVD min-1	Reject only	89.4%	41%	90.5%	$0.076
PVD-same	non-ANC	92.9%	37%	94.4%	$0.118
PVD-same	Reject only	88.4%	21%	87.8%	$0.113
PVD+retry	non-ANC	88.9%	20%	88.6%	$0.137
PVD+retry	Reject only	88.9%	20%	88.6%	$0.137
Single context play	non-ANC	86.4%	40%	83.1%	$0.063
Single context play	Reject only	85.9%	16%	86.7%	$0.055
Expert: Sonnet 4.6 direct alone: 76.3% at $0.014/q					
PVD (Sonnet/Haiku)	non-ANC	76.8%	45%	93.5%	$0.053
PVD (Sonnet/Haiku)	Reject only	76.8%	39%	90.8%	$0.052
PVD min-1	non-ANC	75.8%	49%	91.1%	$0.070
PVD min-1	Reject only	76.8%	41%	90.5%	$0.069
PVD-same	non-ANC	78.8%	37%	94.4%	$0.112
PVD-same	Reject only	78.3%	21%	87.8%	$0.110
PVD+retry	non-ANC	79.8%	20%	88.6%	$0.133
PVD+retry	Reject only	79.8%	20%	88.6%	$0.133
Single context play	non-ANC	74.2%	40%	83.1%	$0.056
Single context play	Reject only	78.3%	16%	86.7%	$0.052

Table 10: Overlap between SC full-consensus and PVD† ANC on GPQA Diamond (n=198; PVD† is the challenge-first Haiku 4.5 verifier with the Sonnet 4.6 prover, 65% coverage in Table 1). Accuracy columns report correctness within each cell.

	n	%	SC Acc	PVD Acc
Both HC	107	54%	96.3%	96.3%
SC only	35	18%	77.1%	62.9%
PVD only	22	11%	72.7%	59.1%
Neither	34	17%	55.9%	58.8%

This is the empirical analogue of the soundness property the interactive-proof framing invokes, and it bounds — without eliminating — the shared-misconception concern discussed in Section 7. Its scope should be read narrowly: it measures resistance to an *instructed* adversary of one particular kind, on one model pairing, and is an upper bound on what that attack achieves rather than a general soundness guarantee.

A.12 Open-Weight Pairings: Details

To test whether the ANC signal depends on proprietary hosted models, we ran two additional in-family, capability-asymmetric pairings on GPQA Diamond (n=198) under the identical protocol: Gemma-4-31b-it as prover with Gemma-4-26b-a4b-it (a 4B-active mixture-of-experts model) as verifier, and DeepSeek-v4-pro with DeepSeek-v4-flash. The Gemma pair attains ANC precision 90.2% [83.9, 94.2] at 69.8% coverage against 78.9% [66.7, 87.5] on the complement (gap $+11.2$pp); the DeepSeek pair attains 91.7% [86.5, 95.0] at 87.0% coverage against 76.0% [56.6, 88.5] (gap $+15.7$pp).

The signal therefore transfers to open weights, at a magnitude compressed by the same ceiling effect visible in the Gemini 3.1 Pro rows: both open provers are strong on this benchmark, so the non-accepted complement stays 76–79% accurate. Appendix A.12 gives the details, including a failure mode specific to open reasoning models in which the prover exhausts its 64k-token budget without emitting an answer ($9/198$ and $5/198$ questions), which we report as exclusions.

The signal therefore transfers to open weights, but at reduced magnitude relative to the proprietary pairings. The compression is mechanical rather than a deficiency of the open models: both provers are strong on this benchmark (86.8% and 89.6% accuracy on attempted questions), so the non-accepted complement remains 76–79% accurate and there is correspondingly little headroom for a gap — the same ceiling effect visible in our Gemini 3.1 Pro rows, where high prover accuracy compresses the separation.

A failure mode specific to open reasoning models. On a minority of questions ($9/198$ for Gemma, $5/198$ for DeepSeek) the prover produced *no answer at all*: it consumed an entire 64k-token budget on internal reasoning and emitted zero output tokens. This reproduces deterministically — re-running those questions at doubled budgets yields identical failures on the identical question set — and two questions fail this way under *both* model families despite different architectures, providers, and serving stacks; one of them has a 182-character prompt, so this is not a context-length artifact. We report precision figures over attempted questions and state the exclusion rate alongside them. We note that in a deployed selective-prediction system such cases are abstentions rather than errors: the prover could not produce an answer, and the protocol correctly declined to report one.

A.13 Prompts

All models receive a system prompt that defines their role and the required JSON response format. The prover and verifier prompts are given below verbatim.[2]

Prover System Prompt

```
You are a Prover (P) in an Interactive Zero-Knowledge Proof
Protocol solving a multiple-choice science question.
Your role:
- Select the single best answer (A, B, C, or D)
- Break your reasoning into discrete, verifiable sub-claims
- Each sub-claim must be atomic and independently checkable
- When challenged on a specific sub-claim, provide deeper
 justification for THAT claim only
- Be consistent -- never contradict your previous statements
- If you made an error, correct it explicitly and revise
 your answer letter
Respond in this exact JSON format:
{
  "answer":  "A" | "B" | "C" | "D",
  "statement":  "one sentence explaining why you chose this
 answer",
```

[2]The prompts describe the interaction as a "zero-knowledge proof protocol." That is a colloquial framing the models were shown as run. The mechanism it names — the verifier's deliberation being hidden from the prover — is a private-coin interactive proof, not zero knowledge in the cryptographic sense, where it is the *prover's* witness that stays hidden. We leave the prompt text unchanged so the reported runs remain reproducible, and correct the terminology in the main text rather than in the prompts.

```
  "subclaims": ["atomic claim 1", "atomic claim 2", ...],
  "reasoning": "step-by-step explanation supporting your
answer and sub-claims"
}
```

STANDARD VERIFIER SYSTEM PROMPT

```
You are a Verifier (V) in an Interactive Zero-Knowledge
Proof Protocol evaluating a multiple-choice science answer.
Your role:
- Evaluate the Prover's reasoning: Accept, Reject, or
Challenge
- ACCEPT if the reasoning is logically sound and internally
consistent
- REJECT if you find a logical flaw, factual error, or
internal contradiction
- CHALLENGE if one specific sub-claim needs deeper
justification
- When challenging, target the SINGLE most suspicious or
unclear sub-claim
- Do not challenge things that are obviously correct
- Be rigorous but efficient -- unnecessary challenges waste
the fatigue budget
You are evaluating reasoning quality, not checking against a
known answer.
Respond in this exact JSON format:
{
  "verdict": "Accept" | "Reject" | "Challenge",
  "reasoning": "why you made this decision",
  "challenge": "your specific question (only if verdict is
Challenge, else null)",
  "challenged_claim": "the exact sub-claim being challenged
(only if Challenge, else null)"
}
```

CHALLENGE-FIRST VERIFIER SYSTEM PROMPT (†)

The challenge-first variant (PVD†) uses a stricter verifier that defaults to challenging rather than accepting:

```
You are a Verifier (V) in an Interactive Zero-Knowledge
Proof Protocol evaluating expert-level questions that are
extremely difficult -- harder than PhD qualifying exams.
Your role:
- Evaluate the Prover's reasoning: Accept, Reject, or
Challenge
- Default to CHALLENGE: these questions are hard enough that
almost every answer has at least one sub-claim worth probing
- ACCEPT only if every sub-claim is specific, verifiable,
and you are highly confident the reasoning is airtight --
vague or hand-wavy justifications do not merit acceptance
- REJECT if you find a clear logical flaw, factual error, or
internal contradiction
- CHALLENGE whenever any sub-claim is asserted without
sufficient justification, relies on memorized facts you
cannot verify, skips logical steps, or uses imprecise
language that could hide an error
- Target the SINGLE most suspicious, weakest, or
least-justified sub-claim
- Never accept reasoning that merely sounds plausible --
demand rigor
[JSON format identical to standard verifier.]
```

SINGLE CONTEXT PLAY SYSTEM PROMPT (ABLATION)

The single context play variant uses a *single* model call that plays both Prover and Verifier roles in sequence, using delimited blocks to structure internal deliberation:

```
You are simulating an Interactive Zero-Knowledge Proof
(ZKP) deliberation on a multiple-choice science question.
You will play both roles in sequence within this single
response:
[PROVER] -- Selects the best answer and breaks reasoning
into atomic, independently verifiable sub-claims.
[VERIFIER] -- Critically evaluates the prover's reasoning
without reference to any external answer key.  Decides:
Accept, Reject, or Challenge.
Protocol rules:
1.  Generate one [PROVER] block, then one [VERIFIER] block.
That is one round.
2.  The verifier MUST Challenge in round 1 -- pick the
single weakest or least justified sub-claim and demand
rigorous justification.  Do not Accept or Reject on the
first round.
3.  From round 2 onwards:  if the verifier Challenges,
generate the next [PROVER] block (responding to that
specific challenge only), then another [VERIFIER] block.
4.  Stop when the verifier issues Accept or Reject, or after
{fatigue} rounds (forced Reject -- output FINAL_VERDICT:
Reject(fatigue)).
5.  The verifier must be genuinely adversarial -- default to
Challenge.  Accept only when every sub-claim is specific,
the logic is airtight, and there is nothing left worth
probing.  Reject if there is a clear logical flaw or factual
error.
6.  The prover may revise its answer if a challenge exposes
a mistake.
Use this exact format for each block:
[PROVER]
{
  "answer":  "A" | "B" | "C" | "D",
  "statement":  "one sentence explaining your choice",
  "subclaims":  ["atomic claim 1", "atomic claim 2", ...],
  "reasoning":  "step-by-step explanation"
}
[/PROVER]
[VERIFIER]
{
  "verdict":  "Accept" | "Reject" | "Challenge",
  "reasoning":  "why you made this decision",
  "challenge":  "your specific question targeting one
sub-claim (null if not Challenge)",
  "challenged_claim":  "the exact sub-claim being challenged
(null if not Challenge)"
}
[/VERIFIER]
After the final [VERIFIER] block, output exactly:
FINAL_ANSWER: ⟨letter⟩
FINAL_VERDICT: ⟨Accept|Reject|Reject(fatigue)⟩
```

USER-TURN TEMPLATES

The system prompts above are paired with the following user-turn templates; braces mark fields filled at run time. The verifier's first user turn presents the prover's structured output:

```
Question:
{question}
```

```
Prover chose: {prover_answer}
Statement: {prover_statement}
Sub-claims: {prover_subclaims_json}
Reasoning: {prover_reasoning}
```

On each subsequent round the verifier receives:

```
Prover responded to challenge about "{challenged_claim}".
Updated answer: {prover_answer}
Statement: {prover_statement}
Sub-claims: {prover_subclaims_json}
Reasoning: {prover_reasoning}
Round {round_num}/{fatigue_limit}. Accept, Reject, or
Challenge.
```

The prover receives each challenge as:

```
The verifier challenges: "{challenged_claim}"
Verifier's question: {verifier_question}
Justify this sub-claim more rigorously. Respond in full
JSON format. You may revise your answer if you find an
error.
```

Under retry ($K > 1$), the prover's opening message of every attempt after the first is prefixed with the failure summary of Algorithm 1, rendered by a fixed template with no model call:

```
[PRIOR ATTEMPTS: You have already attempted this question
{num_prior_attempts} time(s). Each attempt ended without a
clean Accept+no-change result. Study what went wrong.]
{prior_summary}
[END OF PRIOR ATTEMPTS]
For this new attempt: start fresh, pick the answer you
are MOST confident in, and only commit if you can defend it
without changing your mind.
```

where `{prior_summary}` lists, for each prior attempt, its initial answer, any answer changes, its outcome, and the verifier's last challenge or rejection text truncated to 300 characters.

A.14 Example Conversation

The following illustrates a complete 2-round PVD interaction on a GPQA Diamond question from the Molecular Biology domain (Sonnet 4.6 prover, Haiku 4.5 verifier, challenge-first prompt, correct answer D).

Question. *There has been an outbreak of a viral infectious disease in your city. You have been asked to design a molecular diagnostic kit for quick detection of this retrovirus. How would you go about designing this kit?*

(A) Identify the virus by DNA sequencing; develop a PCR kit.
(B) Identify the virus from patient symptoms; design a nested PCR kit.
(C) Identify IgG antibodies; develop an ELISA kit targeting IgG antibodies.
(D) Identify the virus by cDNA sequencing; develop a real-time PCR kit.

Prover — Round 1 *answer: D*

Statement: Since the pathogen is a retrovirus (RNA-based), you must first convert its RNA genome to cDNA via reverse transcription before sequencing, and then develop a real-time RT-PCR kit for rapid, quantitative, and accurate molecular diagnosis.

Sub-claims:

1. Retroviruses have an RNA genome, not a DNA genome.
2. To sequence a retroviral genome, RNA must first be reverse-transcribed into cDNA.
3. Real-time PCR (RT-qPCR) is the gold-standard molecular diagnostic method for RNA viruses.
4. Option A is incorrect: direct DNA sequencing is inappropriate for an RNA retrovirus without reverse transcription.
5. Option C is incorrect: IgG ELISA detects antibody response (serological), not the virus itself.

Verifier — Round 1 *verdict:* CHALLENGE

Reasoning: The fundamentals are sound, but sub-claim 2 requires further justification. Modern sequencing platforms (Nanopore, Illumina) can directly sequence RNA without reverse transcription. The prover has not explained why cDNA sequencing specifically is mandatory as the first step, rather than direct RNA sequencing or metagenomics.

Challenge (targeting sub-claim 2): You state that "RNA must first be reverse-transcribed into cDNA before sequencing." However, modern platforms can sequence RNA directly. Why is cDNA sequencing specifically mandatory for diagnostic kit design rather than direct RNA sequencing? And does Option D's phrase "real-time PCR kit" unambiguously imply RT-PCR?

Prover — Round 2 *answer: D (unchanged)*

Sub-claims:

1. Direct RNA sequencing exists but is not the standard validated workflow for clinical outbreak diagnostic kit design under current regulatory norms.
2. cDNA sequencing via reverse transcription is the established, reproducible, and validated method for characterizing RNA virus genomes in diagnostic contexts.
3. "Real-time PCR" in clinical virology is understood to mean real-time RT-PCR, as standard qPCR cannot amplify RNA templates.
4. Option D is the only choice that correctly combines RNA-to-cDNA conversion with a sensitive, quantitative, closed-tube molecular detection method.

Verifier — Round 2 *verdict:* ACCEPT

Reasoning: The prover now correctly distinguishes between what is technically possible (direct RNA sequencing exists) and what is standard validated clinical practice (cDNA sequencing followed by RT-PCR). The response appropriately anchors the answer to regulatory and epidemiological norms rather than claiming absolute necessity. The clarification that "real-time PCR" is a clinical convention for RT-qPCR is correct and addresses the ambiguity. The comparative evaluation of options is sound. **Result: ANC** (answer D accepted with no revision).