Separating Answer Selection from Computation in Adaptive Language-Model Policies

Anonymous authors
Paper under double-blind review

Abstract

A confidence-guided policy can improve answered accuracy by choosing which answers to return as well as by allocating additional reasoning. We test a fixed controller through interventions that preserve cached branch outcomes, exact query counts and answer coverage. Across two open-weight backbones on MMLU and ARC Challenge, the 25%-query, 80%-coverage controller exceeds state scrambles by 7.50 macro accuracy points. Query-only and abstain-only controls yield gains of 1.55 and 6.94 points at their respective budgets. The results establish that item-aligned state matters for this controller under fixed budgets.

Keywords: test-time computation, selective prediction, confidence-based routing, causal evaluation, language models

TL;DR: State-scramble interventions show that item-aligned state improves a fixed controller's selective accuracy at unchanged query and coverage budgets.

1 Introduction

Adaptive language-model systems decide both how much computation to spend and which answers to return. Confidence can guide either decision (Snell et al., 2025; Alomrani et al., 2025; Zhai et al., 2026; Geifman & El-Yaniv, 2017). Their combination creates an attribution problem: a policy can appear better at reasoning because it uses confidence to reject difficult answers.

We separate these decisions while holding their budgets fixed. For every question, we cache the direct answer and a deliberated answer. A fitted controller predicts the correctness of both branches from pre-deliberation features, queries exactly 25% of items and answers exactly 80%. Reassigning complete state vectors changes the controller's information about individual questions while preserving its parameters, the available branch outcomes and both budgets. Intervening separately at query allocation and answer selection then identifies their contributions to the joint state-alignment effect.

Answer selection accounts for most of the observed effect. The controller exceeds state scrambles by 7.50 accuracy points across two models and two tasks. An exploratory decomposition at the same budgets assigns 6.35 points to answer selection and 1.12 to querying. Adding queries to the no-query selective policy improves accuracy by 0.94 points.

Our contribution is a decision-level audit with paired counterfactual branch outcomes. It measures the separate effects of computation allocation and answer selection, tests them against stronger first-pass-confidence baselines, and repeats the audit under prompt and branch-length changes. The two-model, two-task evaluation measures state dependence, and the decision interventions locate its contribution to each allocation decision.

2 Related Work

Permutation importance and model reliance. Reassigning a feature measures how a fixed predictor relies on its association with other variables and outcomes; it does not by itself identify the feature's natural causal meaning (Fisher et al., 2019). We apply this perspective to two decision stages with cached branch outcomes and exact allocation budgets. The paired branches and exact budgets permit separate query and answer-selection interventions. Section 5 and Appendix G also distinguish fitting uncertainty from sensitivity to the reassignment distribution.

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.

Adaptive test-time compute. Test-time scaling can improve reasoning by choosing how much sampling, search, or verification to spend as a function of the problem (Snell et al., 2025). A growing literature learns compute policies under explicit budgets or constraints (Alomrani et al., 2025; Zhai et al., 2026). Confidence-guided refinement makes the control signal explicit (Jin et al., 2026), while interpretable adaptive sampling studies which observable features drive allocation (Kashaniyan & Jannesari, 2026). We hold the fitted controller and branch capability fixed, then intervene on item-specific state to measure its contribution to routing and answer selection.

The closest concurrent direction is DIAL, which uses counterfactual exploration to learn how the utility of a signal varies across environments, with evaluation across multiple backbones (Li et al., 2026). Our paired-branch intervention holds the fitted policy unchanged and reassigns state under exact query and coverage budgets, separating changes in query allocation from answer selection. A domain-specific uncertainty-routing ablation has also been studied in armed-conflict forecasting (Nemkova & Indukuri, 2026).

Confidence, calibration, and selective answering. Language models expose uncertainty through option probabilities, verbal reports, or second-pass correctness estimates (Kadavath et al., 2022; Xiong et al., 2024). Pre- and post-reasoning confidence can refer to different information sets, so calibration targets must be stage-aligned (Finlay et al., 2026). Selective classification evaluates the risk--coverage tradeoff when a model may reject items (Geifman & El-Yaniv, 2017). Our answered-accuracy measure belongs to this selective setting: it conditions on the exact 80% answer set. Full-coverage comparisons use their own matched budget.

Self-correction and refinement. More reasoning is not uniformly beneficial. Self-refinement can improve outputs (Madaan et al., 2023), but surveys and controlled studies emphasize that models may fail to correct without reliable external feedback and can corrupt initially correct answers (Huang et al., 2024; Kamoi et al., 2024; Yang et al., 2025). Confidence-weighted self-consistency can improve candidate aggregation (Taubenfeld et al., 2025), yet aggregation, branch generation, routing, and abstention are distinct decisions. Our branch-transition counts retain both wrong-to-right and right-to-wrong cases and let the controller estimate two potential-outcome probabilities rather than assuming deliberation is always helpful.

3 Testing State Use in a Fixed Controller

3.1 Fixed branches and pre-deliberation state

For item i, let X_i contain non-state input features, S_i the explicit state computed before deliberation, and $Y_i(0), Y_i(1) \in \{0, 1\}$ the correctness of the direct and deliberated answers. We materialize both branches for every item using deterministic option scoring and greedy generation. The potential outcomes are defined by this fixed two-action procedure.

The five state components are

$$S_i = (1 - p_{\max},\ 1 - \text{margin},\ H_{\text{norm}},\ 1 - v,\ |v - p_{\max}|),$$

where $p_{\max}$, margin, and normalized entropy come from the direct option distribution, and v is a second-pass estimate, aligned to that same answer, that the direct argmax is correct. The three input features are question characters, total choice characters, and choice count. The term uncertainty state denotes these five measured confidence features.

3.2 Controller and exact allocation

For each backbone, we pool its MMLU and ARC development rows and fit two standardized L2 logistic models,

$$\hat{m}_a(X_i, S_i) \approx P(Y_i(a) = 1 \mid X_i, S_i), \qquad a \in \{0, 1\}.$$

The query value is $\hat{\tau}_i = \hat{m}_1 - \hat{m}_0$. The exact top $\lfloor qn + 0.5 \rfloor$ items by $\hat{\tau}_i$ receive the deliberated branch. The chosen branch's predicted correctness is then $\hat{m}_{Q_i}$, and the exact top $\lfloor cn + 0.5 \rfloor$ items by this

score are answered. Seeded ranks break ties. At the primary allocation, $q = 0.25$ and $c = 0.80$; this means 100 queries and 320 answers for each 400-item ARC cell, and 175 queries and 560 answers for each 700-item MMLU cell.

Answered accuracy is

$$A(Q, R) = \frac{\sum_i R_i\{Q_i Y_i(1) + (1 - Q_i)Y_i(0)\}}{\sum_i R_i},$$

where R_i indicates that the system answers. We also report net utility $(N_{\text{correct}} - N_{\text{incorrect}} - 0.05N_{\text{query}})/n$, but answered accuracy is the primary measure.

3.3 State-scramble intervention and estimand

For each evaluation cell, a scramble draws a permutation π within choice-count strata and replaces S_i with $S_{\pi(i)}$. The five components move together, preserving their internal dependence; X_i, both branch outputs, both outcomes, fitted model parameters, and exact budgets remain fixed. We then rerun the unchanged predictor and allocator. This intervention destroys item alignment while preserving the empirical state distribution and the controller's capacity.

The primary contrast is the unweighted four-cell macro mean

$$\Delta = \frac{1}{4}\sum_{g=1}^{4} \left[A_g(S) - E_\pi A_g(S_\pi)\right].$$

We use 2,000 scrambles and the plus-one one-sided value $(1 + \#\{A_\pi \geq A(S)\})/2001$, where both terms are four-cell macro accuracies. The 2.5th--97.5th percentiles shown in figures describe variation across state permutations within the evaluated models and tasks.

The intervention identifies the effect of item--state alignment on the fixed controller's allocation and outcomes. The state can carry confidence, answer identity, difficulty and other output-correlated information; its internal origin is outside this estimand.

3.4 Decision criteria and decompositions

The criteria were fixed before any evaluation row was generated: (i) $\Delta \geq 0.020$; (ii) one-sided $p \leq 0.05$; and (iii) at least three of four condition effects nonnegative. Two secondary allocations were fixed alongside them. **Query only** sets $q = 0.25, c = 1$ to isolate branch routing. **Abstain only** sets $q = 0, c = 0.80$ to isolate answer selection from extra reasoning. Every cell is reported regardless of sign.

4 Experimental Protocol

4.1 Models, tasks, and data separation

We test Qwen3-4B and Granite-3.3-2B-Instruct, two open-weight decoder backbones with separately fitted controllers. MMLU provides broad multiple-choice knowledge questions (Hendrycks et al., 2021); ARC Challenge provides a distinct science-question family (Clark et al., 2018). For each model, development contains 700 MMLU validation and 700 ARC train items. Evaluation contains 700 MMLU test items at the pre-specified offset 5600 and 400 ARC Challenge test items at offset 700. Content hashes and model revisions identify the data and checkpoints. Development and evaluation sets have disjoint question-plus-choice hashes.

The design therefore has 2,800 development and 2,200 evaluation model--item rows, 5,000 in total. The two models use the same question windows. The unweighted macro averages the four model/task strata.

4.2 Branch generation

For each row, a single-token option scorer produces the direct distribution and argmax. A target-aligned verifier scores Yes versus No on the question of whether that exact argmax is correct. The model then receives one fixed deliberation prompt, generates greedily for at most 96 new tokens, and the options are rescored after the continuation. Generation uses BF16, item scoring batch one, and generation batch four on an A800-SXM4-40GB. Both branches are generated for every row before any controller is fitted, ensuring that every evaluated policy replays the same outputs.

Caching both branches reveals $Y(0)$ and $Y(1)$ for paired comparison. A deployed policy would generate only selected branches, while the verifier requires a separate scoring pass for every item.

4.3 Controls

The **state-only** controller uses the five components of S; the **input-only** controller uses only the three length/count features. Both use the identical fitting and allocation code. A seeded **random matched** policy has the same query and answer counts. **Direct no query** and **deliberate all** set q to zero and one, respectively, while retaining 80% coverage for the main control table.

The **content-aware input** control operates on the cached branch bank. It receives the question, ordered choices, and the three surface features, but no option probabilities, verifier output, state feature, or outcome at evaluation. For each backbone, word TF--IDF unigrams/bigrams and vocabulary are fit only on its pooled development questions (`min_df=2`, at most 12,000 terms), then separate L2 logistic models predict direct and deliberated correctness. The unchanged allocator evaluates this controller at all three budget settings. This bag-of-words baseline uses the same evaluation branches as the full controller.

Two constant-state interventions replace all evaluation states by the development-set componentwise 10th or 90th percentile, then rerun the fixed full controller. They test how allocation responds to a globally low or high feature vector. Joint componentwise quantiles can produce a feature vector outside the observed state distribution.

4.4 Analysis design

The 25% query and 80% coverage values were selected on development data before evaluation. The content-aware control, feature correlations, discarded-query analysis, first-pass-confidence baselines, same-budget decomposition, joint allocator, and prompt/cap/checkpoint comparisons are exploratory analyses specified after inspecting the primary results. They reuse the evaluation questions; the primary test and its two secondary allocations use the criteria in Section 3.4.

5 Results

5.1 The joint effect holds in every condition

At 25% query rate and 80% coverage, controller answered accuracy is 0.7694 versus a state-scramble mean of 0.6945. The contrast is +0.0750, with one-sided $p = 0.0005$ and four of four nonnegative conditions. All three criteria are met.

Table 1. Held-out results. Direct and deliberate-all accuracies use every item; controller and scramble accuracies use their exact 80% answer sets. The controller-minus-scramble column compares policies at matched coverage.

Model / task	N	Direct	Deliberate all	Controller	Scramble	Delta	p
Granite / ARC	400	0.7075	0.7100	0.8000	0.7092	+0.0908	0.0005
Granite / MMLU	700	0.5371	0.5486	0.5929	0.5397	+0.0532	0.0005
Qwen / ARC	400	0.8600	0.8975	0.9438	0.8644	+0.0794	0.0005
Qwen / MMLU	700	0.6543	0.6900	0.7411	0.6646	+0.0765	0.0005
Macro	--	**0.6897**	**0.7115**	**0.7694**	**0.6945**	**+0.0750**	**0.0005**

The permutation intervals are wholly positive for every cell (Figure 1a). Route-set Jaccard similarity between intact and scrambled allocations is only 0.151--0.213; answer-set similarity is 0.668--0.680. Re-pairing state changes both decision sets.

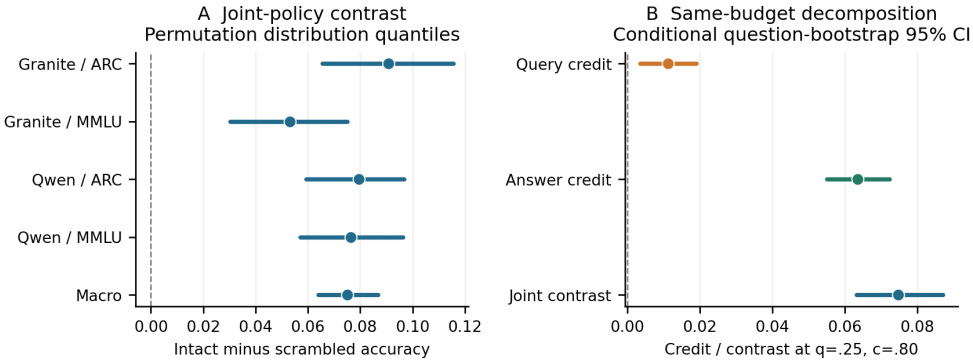

Figure 1. State alignment improves selective accuracy mainly through answer selection. A: Controller-minus-scramble contrasts; bars describe the permutation distribution. B: Path-averaged decision credits at $q = 0.25, c = 0.80$, with marginal question-bootstrap 95% intervals conditional on development fits.

5.2 A SAME-BUDGET DECOMPOSITION LOCALIZES DECISION RELIANCE

The four-cell analysis crosses intact/scrambled state assignment separately at the query and answer stages, keeping $q = 0.25, c = 0.80$ in every cell. Path-averaged credits divide each decision's marginal effect equally across the two possible intervention orders. They sum exactly to the joint alignment contrast in this analysis.

Table 2. Marginal 95% question-bootstrap intervals conditional on the development fit. Query and answer credits sum to 0.0747; the interaction is a separate contrast, not a third additive credit. The 0.0747 estimate uses the decomposition resampling scheme; the primary permutation estimate is 0.0750. Appendix F gives the four cells and resampling definition.

Four-cell estimand	Macro [95% CI]
Query path credit	+0.0112 [+0.0036, +0.0191]
Answer path credit	+0.0635 [+0.0550, +0.0723]
Interaction	-0.0109 [-0.0205, -0.0009]
Joint alignment contrast	+0.0747 [+0.0632, +0.0870]

The query-only ($c = 1$) and abstain-only ($q = 0$) contrasts are +0.0155 and +0.0694. They answer functional questions at different budgets and cannot be added to decompose the primary effect. Query-only effects are heterogeneous, including a slightly negative Granite/MMLU value; the full secondary table remains in Appendix F.4.

5.3 State features improve allocation beyond the fixed input controls

Table 3. Unweighted macro across conditions. The content-aware baseline is exploratory; the other rows were specified before evaluation. Net utility assigns a cost of 0.05 per deliberation call.

Policy at 80% coverage	Query rate	Macro accuracy	Net utility
Full controller	0.25	0.7694	0.4186
State only	0.25	0.7710	0.4211
Input only	0.25	0.7232	0.3446
Content-aware input	0.25	0.7296	0.3548
Random matched	0.25	0.6907	0.2927
Direct, no query	0.00	0.7600	0.4161
Deliberate all	1.00	0.7692	0.3807
State scramble	0.25	0.6945	0.2986
Constant low state	0.25	0.7204	0.3402
Constant high state	0.25	0.7202	0.3398

State-only is numerically similar to, and slightly above, the full controller; the three-feature input-only baseline is lower. The content-aware input controller reaches 0.7296, leaving the full controller +0.0398 higher at the primary setting. The four cell gaps are +0.0625, +0.0357, +0.0594, and +0.0018 (Granite/ARC, Granite/MMLU, Qwen/ARC, Qwen/MMLU), so the direction is consistent but nearly absent in one cell. At full coverage the query-only gap is only +0.0106; with no queries the abstain-only gap is +0.0329. Replacing every state with low or high development quantiles yields almost identical macro accuracy near 0.720. The numerical state features improve allocation beyond both tested input controls, primarily through answer selection.

The state contains option-distribution and candidate-verifier features. The lexical baseline nearly matches the full controller in Qwen/MMLU, indicating that the advantage of numerical state features depends on the task and model.

First-pass confidence is a stronger baseline than coarse input features. The feature comparison fits the same two correctness predictors on development data using eight feature sets, with exact $q = 0.25, c = 0.80$ at evaluation. Removing both verifier-derived features gives macro accuracy 0.7750 versus 0.7694 for the full controller. Full-minus-baseline intervals adjusted for seven comparisons include zero for every first-pass-confidence baseline. The full controller exceeds verifier-only and surface-input controls. Appendix F reports all baselines, risk–coverage curves, and the query-budget grid. First-pass confidence provides a competitive allocation signal; the intervals leave the verifier's incremental contribution unresolved.

Item alignment retains a 0.0725 accuracy contrast under input-stratified reassignment, close to 0.0747 with choice-count strata alone. A two-stage bootstrap resamples development questions, refits both controllers and independently resamples evaluation questions. Its full-minus-no-verifier interval is [-0.0105, +0.0102]. Alignment therefore remains useful under the finer strata, while the refitted comparison leaves the verifier's incremental contribution uncertain (Appendix G).

5.4 Deliberation has heterogeneous potential outcomes

Uniform deliberation yields both corrections and corruptions. Wrong-to-right versus right-to-wrong transitions are 31/30 for Granite/ARC, 78/70 for Granite/MMLU, 25/10 for Qwen/ARC, and 65/40 for Qwen/MMLU. Granite gains one correct answer on ARC and eight on MMLU; Qwen gains 15 and 25, respectively. Query allocation must account for both potential outcomes.

Within the selected query sets, correction and corruption rates also vary (Appendix B). Yet at 80% coverage, adding queries to the no-query selective policy raises macro accuracy by only 0.94 points and net utility by 0.0025. This incremental gain measures the benefit of adding reasoning at matched coverage.

5.5 Token accounting exposes a deployment inefficiency

The primary design holds the number of deliberation calls constant. Of 2,200 held-out deliberation traces, 2,164 hit the 96-token cap; only 36 Granite traces ended earlier. The controller selects 52,713

generated tokens across 550 calls. Relative to state scrambles, selected tokens are identical in both Qwen cells and only 0.27% (Granite/ARC) and 0.31% (Granite/MMLU) higher than the scramble means. The query-only comparison is consequently nearly generated-token matched, but prompt lengths, scoring work, attention cost, and hardware latency are not measured.

Of 550 queried items, 284 are later abstained: 54, 60, 77, and 93 in the four table rows. A joint allocator uses the same predicted branch correctness and exact budgets but requires every queried item to be answered. This removes all 284 discarded queries and changes macro accuracy from 0.7694 to 0.7769; the paired difference is +0.0075 [95% CI −0.0015, +0.0122]. The constraint redirects the fixed query budget to returned answers, with an accuracy interval that includes zero. Appendix F gives all four cells and the optimization constraint.

5.6 Longer branches and a third checkpoint

The prompt and branch-length comparison crosses prompt A with 96/192/384/512-token caps and prompt B with 96/384-token caps in both original checkpoints, refitting each controller. All 24 model/task/setting alignment contrasts are positive. At 25% queries and 80% coverage, every full-minus-no-verifier 95% interval includes zero after development refitting. Longer branches can still improve the available capability: at full coverage and 25% queries, A/384 improves macro accuracy over A/96 by 0.0313 [five-setting-adjusted 99% CI 0.0114, 0.0498]. The corresponding 80%-coverage interval includes zero.

Qwen still reaches every cap, including 512 tokens. Granite's A/512 mean length is 304 tokens on MMLU and 239 on ARC, with 8.1% and 1.8% reaching the cap. The longest setting reduces truncation for Granite; Qwen remains capped. All six settings, four budgets and paired refit intervals appear in Appendix H.

Phi-3.5-mini supplies a third A/96 bank. Deliberation lowers MMLU accuracy from 0.6771 to 0.5771 and ARC accuracy from 0.8750 to 0.7125, while the controller retains positive alignment contrasts of 0.0760 and 0.0522. An alignment effect can coexist with harmful average branch outcomes. The Phi comparison uses the same questions as the two-model evaluation.

5.7 Decision credits persist across branch settings

We repeat the same-budget four-cell intervention across all six existing prompt/cap settings, refitting each controller on that setting's development rows. At 25% queries and 80% coverage, macro answer-selection credits range from 5.89 to 6.90 accuracy points; query credits range from 0.84 to 2.12. Paired answer-minus-query credits are 3.89--6.07 points, with all six adjusted intervals above zero (Table I3). At A/384, the query credit is 2.12 [six-setting-adjusted 99.167% CI 1.06, 3.35] and the answer credit is 6.07 [4.88, 7.19]. These intervals pair questions across models and condition on the development fits.

Longer branches also expose available gains that the fitted query policy leaves unused. With the same 25% query budget at full coverage, hindsight allocation reaches 76.13--82.05% accuracy across settings, versus 70.00--74.33% for the fitted policy (Figure 2B). The oracle uses both recorded outcomes, so it measures branch capability rather than a deployable routing rule. At A/384, its 81.81% ceiling exceeds the fitted policy's 74.30%. The smaller query credit therefore coexists with useful branch alternatives. Appendix I gives the complete credits, oracle construction and uncertainty definition.

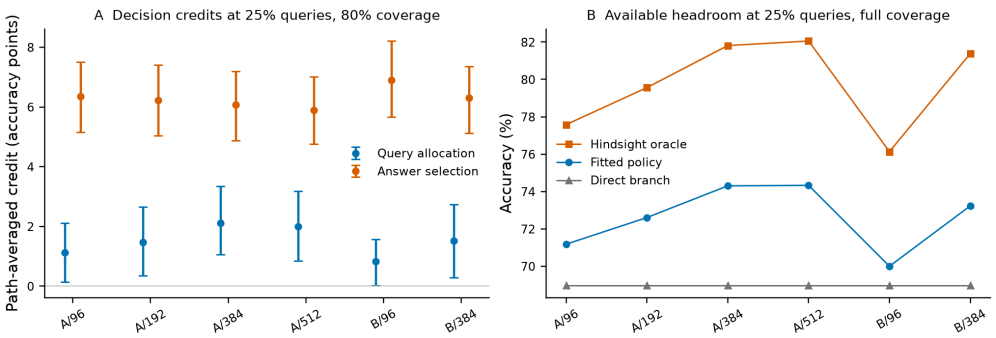

Figure 2. Answer selection remains the larger contribution as branch settings change. A: Decision credits at fixed 25% queries and 80% coverage; bars are six-setting-adjusted 99.167% question-bootstrap intervals conditional on development fits. B: Exact-budget hindsight ceilings and fitted-policy accuracy at full coverage, with the direct branch for reference. A/B identify prompts and numbers identify token caps. All six settings reuse the same evaluation questions.

6 State Use, Answer Selection, and Deliberation

Item-state alignment causally improves selective accuracy. When the same fitted module receives another item's complete state, with capability and budgets held fixed, its outcome deteriorates in every cell. Constant-state interventions and low route overlap confirm substantial behavioral dependence.

The dominant demonstrated function is answer selection. At the same 25% query and 80% coverage budgets, the answer credit is 6.35 points and query credit is 1.12 points, summing to that analysis's 7.47-point contrast. The 6.94- and 1.55-point comparisons use different budgets and are not additive credits.

Branch quality and allocation quality can be separated. The six-setting analysis retains a larger answer-selection credit while the oracle exposes unused gains from the available branches. This identifies a concrete target for stronger learned policies (Zhai et al., 2026; Jin et al., 2026; Li et al., 2026): selecting useful branch changes, evaluated separately from deciding which answers to return.

First-pass confidence supplies a competitive allocation signal. The controller without verifier features reaches 0.7750 macro accuracy, compared with 0.7694 for the full controller, with adjusted difference intervals spanning zero. These baseline comparisons distinguish reliance on item-aligned state from the incremental contribution of the verifier.

Caching both branches and intervening at the two decision stages separates how well the model can answer from how the policy allocates computation and selects its returned answers.

7 Limitations

The primary evaluation covers two selected models and two multiple-choice tasks, with question windows shared across models. Its 2,000-permutation test measures dependence within those conditions. Query and coverage budgets were selected using development data; the held-out result applies to that fixed policy. Selective and full-coverage accuracy use different denominators.

The original input control uses three surface features. The TF--IDF baseline remains 3.98 accuracy points below the full controller, without a sampling interval, and nearly closes the gap in one condition. Stronger text encoders may recover more of the allocation. State features are correlated operational measures of confidence and disagreement; they can also encode answer identity or difficulty. Componentwise low/high interventions can create states outside the observed support.

Baseline intervals in Appendix F condition on fitted predictors and adjust for seven comparisons; Appendix G includes development-refitting uncertainty. Curves use pointwise intervals. Query counts and coverage are exact, while token counts are nearly matched when generations reach the cap. Runtime, FLOPs and energy are unmeasured; the verifier adds work to every item. More than half of the

sequential policy's deliberations are later rejected. The joint alternative changes this allocation but retains the same call count. The utility coefficient of 0.05 is a specified analysis cost.

Caching both outcomes supports paired interventions on fixed questions. Interactive agents require trajectory-level counterfactuals because actions change later observations. The present audit supplies a decision-level test for a policy with observable branch alternatives.

8 Reproducibility

The supplement provides all 5,000 branch rows, model revisions, data hashes, state definitions, controller settings and exact-budget rules. Appendix D describes the software environments, data validation and relationship to concurrent submissions. Appendix F provides the baselines, risk–coverage curves, decision credits and joint-allocation estimates, with their resampling definitions.

The supplement also contains all six prompt/cap settings and the Phi checkpoint, including every tested budget, paired setting contrast and development-refitting interval.

AI use statement

Generative AI tools were used to critique and refine hypotheses and experimental methodology; suggest experimental parameters; implement and edit analysis and experiment code; support statistical auditing and result interpretation; generate plotting code; search for and format literature; and draft and edit manuscript text. No generative AI tool supplied benchmark labels or served as an outcome judge in this study. All AI-assisted analyses are checked against trial-level artifacts and automated tests. The authors take responsibility for the final content and verified the disclosure against their use of these tools.

Ethics statement

This study analyzes open-weight language models and public benchmark data. It does not involve private user data or deployment on people.

Reproducibility statement

The appendix and anonymous supplement specify data partitions, model revisions, precision, seeds, intervention sites, statistical criteria, and per-trial records. The released test suite covers branch construction, probability and target validation, exact-budget allocation, state scrambling, the content-aware control, manifest locks, and holdout separation.

References

Mohammad Ali Alomrani, Yingxue Zhang, Derek Li, Qianyi Sun, Soumyasundar Pal, Zhanguang Zhang, Yaochen Hu, Rohan Deepak Ajwani, Antonios Valkanas, Raika Karimi, Peng Cheng, Yunzhou Wang, Pengyi Liao, Hanrui Huang, Bin Wang, Jianye Hao, and Mark Coates. Reasoning on a budget: A survey of adaptive and controllable test-time compute in LLMs, 2025. URL https://arxiv.org/abs/2507.02076.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try ARC, the AI2 reasoning challenge, 2018. URL https://arxiv.org/abs/1803.05457.

Conor Finlay, Joshua Kurien, Saurabh Dash, Marzieh Fadaee, and Beyza Ermis. CALIBER: Calibrating confidence before and after reasoning in language models, 2026. URL https://arxiv.org/abs/2606.24281.

Aaron Fisher, Cynthia Rudin, and Francesca Dominici. All models are wrong, but many are useful: Learning a variable's importance by studying an entire class of prediction models simultaneously. *Journal of Machine Learning Research*, 20(177):1–81, 2019. URL https://www.jmlr.org/papers/v20/18-760.html.

Yonatan Geifman and Ran El-Yaniv. Selective classification for deep neural networks. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations*, 2021.

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

Chen Jin, Ryutaro Tanno, Tom Diethe, and Philip Teare. CoRefine: Confidence-guided self-refinement for adaptive test-time compute, 2026. URL https://arxiv.org/abs/2602.08948.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, Scott Johnston, Sheer El-Showk, Andy Jones, Nelson Elhage, Tristan Hume, Anna Chen, Yuntao Bai, Sam Bowman, Stanislav Fort, Deep Ganguli, Danny Hernandez, Josh Jacobson, Jackson Kernion, Shauna Kravec, Liane Lovitt, Kamal Ndousse, Catherine Olsson, Sam Ringer, Dario Amodei, Tom Brown, Jack Clark, Nicholas Joseph, Ben Mann, Sam McCandlish, Chris Olah, and Jared Kaplan. Language models (mostly) know what they know, 2022.

Ryo Kamoi, Yusen Zhang, Nan Zhang, Jiawei Han, and Rui Zhang. When can LLMs actually correct their own mistakes? a critical survey of self-correction of LLMs. *Transactions of the Association for Computational Linguistics*, 12:1417–1440, 2024. doi: 10.1162/tacl_a_00713. URL https://aclanthology.org/2024.tacl-1.78/.

Mobina Kashaniyan and Ali Jannesari. Interpretable adaptive sampling for LLM test-time scaling, 2026. URL https://arxiv.org/abs/2608.03961.

Ziming Li, Jiatan Huang, Xiaoguang Guo, Guilin Wang, and Chuxu Zhang. Same signal, opposite meaning: Direction-informed adaptive learning for LLM agents, 2026. URL https://arxiv.org/abs/2605.06908.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. Self-refine: Iterative refinement with self-feedback. In *Advances in Neural Information Processing Systems*, volume 36, 2023. doi: 10.52202/075280-2019. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/91edff07232fb1b55a505a9e9f6c0ff3-Abstract-Conference.html.

Poli Apollinaire Nemkova and Haeshitha Indukuri. What drives LLM self-reflection? a controlled ablation of uncertainty routing in armed conflict forecasting, 2026. URL https://arxiv.org/abs/2608.12322.

Charlie Victor Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling LLM test-time compute optimally can be more effective than scaling parameters for reasoning. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=4FWAwZtd2n.

Amir Taubenfeld, Tom Sheffer, Eran Ofek, Amir Feder, Ariel Goldstein, Zorik Gekhman, and Gal Yona. Confidence improves self-consistency in LLMs. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 20090–20111. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.findings-acl.1030. URL https://aclanthology.org/2025.findings-acl.1030/.

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 *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=gjeQKFxFpZ.

Zhe Yang, Yichang Zhang, Yudong Wang, Ziyao Xu, Junyang Lin, and Zhifang Sui. Confidence v.s. critique: A decomposition of self-correction capability for LLMs. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3998–4014. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.acl-long.203. URL https://aclanthology.org/2025.acl-long.203/.

Zhiyuan Zhai, Bingcong Li, Bingnan Xiao, Ming Li, and Xin Wang. Adaptive test-time compute allocation for reasoning LLMs via constrained policy optimization, 2026. URL https://arxiv.org/abs/2604.14853.

A Exact State, Prompts, and Allocation

A.1 Feature definitions

Let the normalized direct option distribution be p. The five state features are:

Feature	Definition
decision uncertainty	$1 - \max_j p_j$
inverse margin	$1 - (p_{(1)} - p_{(2)})$
normalized entropy	$-\sum_{j:p_j>0} p_j \log p_j / \log K$
verifier uncertainty	$1 - P(\text{Yes} \mid q, z = \arg\max p)$
monitor disagreement	$\lvert P(\text{Yes} \mid q, z) - \max_j p_j \rvert$

Question characters, summed choice characters, and number of choices form X. The full controller receives all eight features. Feature values are computed from the stored option and verifier distributions.

Table A1. Pearson correlations between state features, numbered in the order defined above. Each column uses its model's held-out questions: 400 for ARC and 700 for MMLU. The table describes within-cell dependence among the five features.

Feature pair	Granite/ARC	Granite/MMLU	Qwen/ARC	Qwen/MMLU
1, 2	0.992	0.990	0.993	0.990
1, 3	0.973	0.961	0.968	0.964
1, 4	0.243	0.238	0.277	0.283
1, 5	0.072	0.082	0.254	0.251
2, 3	0.959	0.934	0.946	0.938
2, 4	0.233	0.212	0.255	0.282
2, 5	0.070	0.078	0.244	0.250
3, 4	0.265	0.280	0.360	0.327
3, 5	0.103	0.111	0.312	0.276
4, 5	0.938	0.843	0.929	0.855

A.2 Branch procedure

The direct prompt presents the question and labeled alternatives and asks for the answer. The single-token scorer evaluates only the option labels. The verifier prompt presents the same question and choices, names the direct argmax as the candidate, and asks whether that candidate is correct; Yes/No probabilities are normalized. The fixed deliberation prompt asks the model to solve the problem carefully, check each option, and end by identifying the best answer. After at most 96 greedy tokens, the original options are rescored conditioned on that continuation. The controller never receives generated text, deliberated probabilities, deliberated identity, generated length, or either correctness label at evaluation time.

A.3 Exact budget and tie behavior

Rates are converted to counts by $\lfloor rn + 0.5 \rfloor$. Query scores are sorted first; a second sort ranks selected-branch correctness for coverage. A seeded pseudorandom permutation provides a deter-

ministic secondary key for exact ties rather than adding floating-point jitter. Scrambled, input-only, state-only, and random controls all call the same allocator.

B Full Branch and Policy Diagnostics

B.1 Branch transitions

Cell	Wrong to right	Right to wrong	Net all-deliberation change
Granite / ARC	31	30	+0.0025
Granite / MMLU	78	70	+0.0114
Qwen / ARC	25	10	+0.0375
Qwen / MMLU	65	40	+0.0357

These counts cover all evaluation items, not only queried or answered subsets. They show why ranking predicted branch advantage can matter even when the all-deliberation mean is close to direct accuracy.

B.2 Primary queried-subset diagnostics

Cell	Correction rate	Corruption rate	Revision rate	Queried then abstained
Granite / ARC	0.170	0.060	0.290	54/100
Granite / MMLU	0.143	0.143	0.497	60/175
Qwen / ARC	0.210	0.070	0.310	77/100
Qwen / MMLU	0.223	0.120	0.451	93/175

Correction and corruption rates are proportions within the queried set. Revision counts any change in option identity, whether or not correctness changes. The last column counts queried items excluded from the returned-answer set.

B.3 State interventions and overlaps

Cell	Scramble query Jaccard	Scramble answer Jaccard	Low-state accuracy	High-state accuracy
Granite / ARC	0.198	0.673	0.7531	0.7500
Granite / MMLU	0.213	0.680	0.5446	0.5375
Qwen / ARC	0.154	0.668	0.8875	0.8969
Qwen / MMLU	0.151	0.680	0.6964	0.6964

Scramble Jaccards are means across the 2,000 permutations. Constant-state query sets also differ markedly from the observed allocation; low and high vectors can produce similar rankings because every item receives the same state and the remaining variation comes from X and model intercept/interaction-free linear scores.

The content-aware input control gives the following primary answered accuracies:

Cell	Full controller	Content-aware input	Full − content
Granite / ARC	0.8000	0.7375	+0.0625
Granite / MMLU	0.5929	0.5571	+0.0357
Qwen / ARC	0.9438	0.8844	+0.0594
Qwen / MMLU	0.7411	0.7393	+0.0018
Macro	**0.7694**	**0.7296**	**+0.0398**

The corresponding macro full-minus-content differences are +0.0106 for query only and +0.0329 for abstain only. These descriptive comparisons have no prespecified decision threshold or sampling interval.

B.4 Generated-token counts

Cell	Traces below cap	Controller tokens	Scramble mean	Difference	Queried then abstained
Granite / ARC	22/400	9,523	9,497.6	+25.4	54
Granite / MMLU	14/700	16,790	16,738.6	+51.4	60
Qwen / ARC	0/400	9,600	9,600.0	0.0	77
Qwen / MMLU	0/700	16,800	16,800.0	0.0	93

Only generated continuation tokens are counted. Direct scoring, verifier scoring, deliberation-prompt prefill, and post-deliberation option scoring are omitted, so these totals cannot be converted into an end-to-end speedup.

C Decision criteria

The configuration was specified at `2026-09-04T11:05:00Z`, before evaluation generation. Query rate and coverage were selected on development data. The decision criterion uses the disjoint evaluation questions from both models and tasks.

The decision criterion is conjunctive. The observed values are:

Requirement	Threshold	Observed	Decision
Macro controller minus scramble	at least +0.020	+0.074959	pass
One-sided permutation p	at most 0.05	0.000500	pass
Nonnegative cells	at least 3/4	4/4	pass

D Data and Software

D.1 Model and data versions

- Qwen3-4B revision: `1cfa9a7208912126459214e8b04321603b3df60c`.
- Granite-3.3-2B-Instruct revision: `707f574c62054322f6b5b04b6d075f0a8f05e0f0`.
- Configuration hash: `sha256:bc81159148be226b22ac635fe24f444a16e0d413ca2bf57f76660ece7a46ca53`.
- Independent 13-file lock SHA-256: `781623b09b957a48b647a39c7b16120b9cf550d807055302e08e9f165c1c95c2`.
- Analysis-data digest excluding machine-specific input paths: `c92a053483090428879ffa0db26247be1e3938079f0b294d255fe7d21d58580d`.

The four held-out data slices have normalized question/choice hashes. The supplement contains eight JSONL branch files and eight manifests recording row counts, model revisions, task IDs, prompt IDs, and source/configuration hashes.

D.2 Environments

Generation ran on one NVIDIA A800-SXM4-40GB under Python 3.11.16, PyTorch 2.13.0+cu130, Transformers 4.57.6, and Datasets 4.8.5. Canonical analysis used NumPy 2.4.6, scikit-learn 1.9.0, and SciPy 1.17.1. Two Tesla V100-SXM2-32GB validation hosts replayed the unchanged analyzer with NumPy 1.26.4, scikit-learn 1.6.1, and SciPy 1.14.1.

Both validation hosts reproduced the local summaries: `summary.csv` has SHA-256 `1e862e5b...da31e`, and `results.md` has SHA-256 `f15d5f4b...ad77`. Analysis data share the digest above after excluding machine-specific input paths.

D.3 Data validation

Validation checks model revisions, dataset hashes, dtypes, prompts, scorer modes, source snapshots, unique in-range indices, finite normalized option probabilities, derived state features, candidate/outcome consistency, manifest hashes, and disjoint development/evaluation content.

Concurrent-submission relationship. One concurrent anonymous submission studies candidate-selection dependence in second-pass correctness verification; another studies activation-induced false event reports in fictional contexts. This paper shares the general QA scorer, candidate-verifier convention, some model families, manifest infrastructure, and methodological vocabulary with the former, and only broader infrastructure and vocabulary with the latter. The candidate-selection paper's A1 and A1b banks share no normalized question stems with this paper's MMLU evaluation. Its earlier C-T1H window shares three stems, two with identical choices; its C-T3a evaluation shares one stem with different choices. Source-row windows, generated branches, fitted controllers, interventions and reported estimates remain separate. The relationships will be disclosed in the submission system, with anonymous cross-references only if the live system makes the corresponding records reviewer-accessible.

E Interpreting the Intervention Contrasts

The joint contrast measures the effect of item-state alignment on the fitted controller at matched query and answer budgets. The same-budget decomposition divides this effect between query allocation and answer selection. Query-only and abstain-only comparisons instead change the budgets to isolate each function.

State reassignment measures dependence on the supplied features. Its interpretation depends on the information those features encode, including confidence, answer identity and difficulty. Population generalization, calibrated confidence and deployment cost require separate evaluation.

F Confidence Baselines and Allocation

F.1 Feature scope and paired uncertainty

All predictors use the same development-only logistic fits and cached evaluation branch outcomes. Full uses three surface features plus all five state features. No verifier removes verifier uncertainty and monitor disagreement, retaining three first-pass and three surface features. Max probability, margin, entropy and verifier each use one feature; first-pass state uses the first three state features; surface input uses the three length/count features. No evaluation outcome is used to fit a predictor or choose a query.

Table F1. Macro accuracy at $q = 0.25, c = 0.80$ and discrete area under the risk–coverage curve (AURC; lower is better). Intervals condition on development-fitted predictors. Accuracy has marginal 95% intervals; full-minus-baseline contrasts use 99.286% intervals to adjust for seven comparisons. All four model–task cells receive equal macro weight; question resamples are shared across models within each benchmark. The complete cell-wise intervals, including AURC intervals, are in the accompanying JSON.

Predictor inputs	Accuracy [95% CI]	AURC	Full minus row [99.286% CI]
Full	0.7694 [0.7459, 0.7929]	0.1473	--
No verifier	0.7750 [0.7502, 0.7958]	0.1494	-0.0056 [-0.0147, +0.0080]
Max probability	0.7602 [0.7375, 0.7834]	0.1410	+0.0093 [-0.0043, +0.0227]
Margin	0.7609 [0.7369, 0.7839]	0.1415	+0.0085 [-0.0054, +0.0234]
Entropy	0.7642 [0.7418, 0.7871]	0.1395	+0.0052 [-0.0095, +0.0182]
Verifier	0.7164 [0.6916, 0.7404]	0.2239	+0.0530 [+0.0327, +0.0731]
First-pass state	0.7667 [0.7436, 0.7895]	0.1388	+0.0027 [-0.0100, +0.0162]
Surface input	0.7232 [0.6981, 0.7452]	0.2787	+0.0462 [+0.0269, +0.0692]

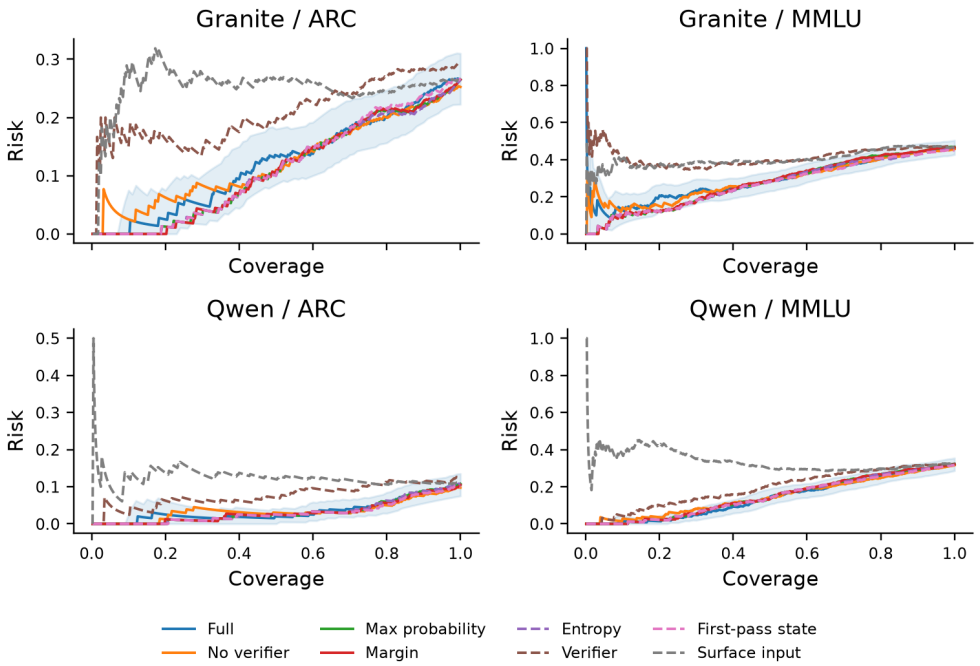

Figure F1. Every attainable coverage at a fixed 25% query fraction. Shading shows only the full controller's pointwise 95% intervals for readability; intervals for all eight predictors are included in the data view. These are not simultaneous confidence bands.

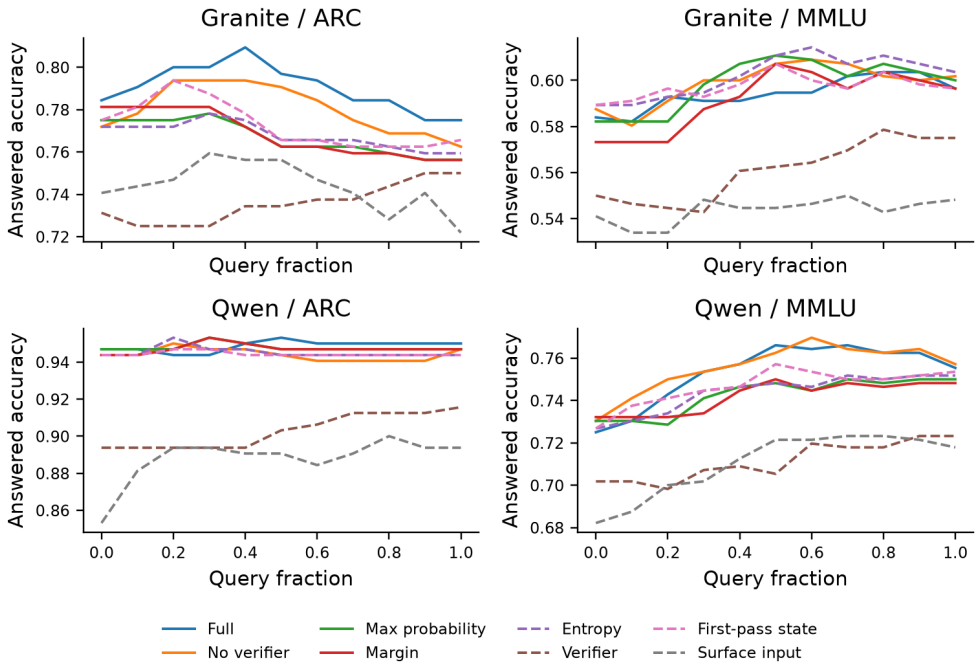

Figure F2. Descriptive query-budget sweep at fixed 80% coverage on the same branch bank. The grid $q = 0, 0.1, \ldots, 1$ uses unchanged development-fitted predictors. Points show accuracy at each query rate without sampling intervals. The primary budget was selected on development data.

F.2 Same-budget joint allocation

Let p_i^D, p_i^F be the full controller's predicted direct and deliberated correctness. Choose disjoint direct-answer and queried-answer sets D, F to maximize $\sum_{i \in D} p_i^D + \sum_{i \in F} p_i^F$, subject to $|F| = \lfloor 0.25n + 0.5 \rfloor$ and $|D| + |F| = \lfloor 0.80n + 0.5 \rfloor$. All remaining items abstain. Sorting by $p_i^F - p_i^D$ and evaluating prefix/suffix top-k sums solves this fixed-score assignment. The objective is predicted correctness, not evaluation correctness. The original sequential comparator permits queried items to abstain, so the two feasible allocation sets differ despite equal query and answer counts.

Table F2. Same fitted full-controller predictors, exact budgets and branch bank. Intervals use 2,000 paired question bootstraps and hold the development fit fixed. The joint constraint eliminates discarded queries at unchanged query and answer counts. The macro accuracy interval spans zero.

Cell	Sequential	Joint	Difference [95% CI]	Discarded queries: seq/joint
Granite/ARC	0.8000	0.8063	+0.0062 [-0.0062, +0.0188]	54 / 0
Granite/MMLU	0.5929	0.5893	-0.0036 [-0.0143, +0.0143]	60 / 0
Qwen/ARC	0.9437	0.9531	+0.0094 [-0.0094, +0.0156]	77 / 0
Qwen/MMLU	0.7411	0.7589	+0.0179 [-0.0036, +0.0286]	93 / 0
Macro	0.7694	0.7769	+0.0075 [-0.0015, +0.0122]	284 / 0

F.3 Four-cell alignment decomposition

The four-cell decomposition crosses intact/scrambled state assignment at the query decision with intact/scrambled assignment at the answer decision while retaining $q = 0.25, c = 0.80$ in every cell. It differs from the original query-only ($c = 1$) and abstain-only ($q = 0$) tests. The two path-averaged credits sum exactly to the joint intact-minus-scrambled contrast. Point estimates average 512 state permutations; conditional question-bootstrap intervals use 32 permutations per resample and do not include controller-fitting uncertainty.

The numerical credits and intervals are in Table 2 and Figure 1B.

The 95% intervals in Table 2 describe question-sampling uncertainty conditional on the fitted controller.

F.4 Comparisons across different budgets

Table F4. Separate functional comparisons. These rows use different budgets and are not additive effect components. Query only always answers and therefore isolates which branch is selected. Abstain only never deliberates and isolates which direct answers are returned.

Analysis	Query rate	Coverage	Controller	Scramble	Delta	p	Nonnegative
Joint primary	0.25	0.80	0.7694	0.6945	**+0.0750**	0.0005	4/4
Query only	0.25	1.00	0.7118	0.6963	**+0.0155**	0.0010	3/4
Abstain only	0.00	0.80	0.7600	0.6907	**+0.0694**	0.0005	4/4

At full coverage, query-only macro accuracy 0.7118 is numerically close to 0.7115 for deliberating on every item. This numerical proximity is descriptive; equivalence was not tested. The four query-only alignment contrasts are +0.0236, -0.0013, +0.0242 and +0.0154 in the primary table order.

G Refitting and Input-Stratified Reassignment

We compare the full and no-verifier controllers with resampling of the existing branch bank. Development and evaluation questions are independently resampled within each benchmark, with the same draws shared across backbones. Each development resample refits both original logistic correctness models using the fixed regularization and preprocessing; each evaluation resample reruns exact-budget allocation. All 2,000 draws are retained. The macro point contrast is -0.0056.

Table G1. Paired percentile intervals. The two-stage interval incorporates development-fitting and evaluation-question uncertainty; the other rows isolate their respective resampling scopes. The intervals condition on the selected tasks, features and branch procedure.

Resampling scope	Full minus no-verifier [95% CI]
Evaluation questions only	[-0.0115, +0.0041]
Development refits only	[-0.0067, +0.0068]
Development and evaluation	[-0.0105, +0.0102]

For a separate reliance sensitivity analysis, complete states are reassigned within choice count crossed with development-defined question-length quartiles. Quartile cuts are retained only if every development bin has at least 40 questions; otherwise that choice group uses no length subdivision. Evaluation groups, including sparse groups, are retained. Every source state remains in the same defined input stratum, while other input/state dependence can still change.

Table G2. Each row uses 2,000 permutations and the original fitted controller. The interval column describes reassignment variability; MC SE quantifies uncertainty in the simulated mean. Preserving question-length strata retains a substantial alignment contrast. The accompanying JSON gives all stratum counts, cutoffs, cell results and resampling definitions.

Reassignment strata	Alignment contrast	Permutation 2.5–97.5%	Mean MC SE
Choice count	0.0747	[+0.0634, +0.0857]	0.000125
Choice count and question length	0.0725	[+0.0621, +0.0837]	0.000125

H Matched-runtime prompt, cap and checkpoint sensitivity

We generate complete branch banks at prompt A caps 96, 192, 384 and 512, and prompt B caps 96 and 384, for Qwen3-4B and Granite-3.3-2B. Phi-3.5-mini adds an A/96 bank. All banks retain the original development/evaluation questions and BF16 runtime. The 13 model/settings each contain 2,500 branches; the 32,500 generations reuse the same questions. Controllers are refitted separately within each setting. A two-stage bootstrap uses 2,000 paired development/evaluation resamples. Five nonbaseline cap/prompt contrasts receive 99% intervals, with multiplicity adjustment within each budget. Reassignment quantiles describe variation across shuffled allocations.

Setting	Model	Task	Direct	Deliberated	Mean tokens	At cap	Alignment
a096	Qwen3-4B	mmlu	0.6543	0.6900	96.0	1.000	0.0767
a096	Qwen3-4B	arc	0.8600	0.8975	96.0	1.000	0.0792
a096	Granite-3.3-2B	mmlu	0.5371	0.5486	95.7	0.980	0.0524
a096	Granite-3.3-2B	arc	0.7075	0.7100	95.1	0.945	0.0899
a192	Qwen3-4B	mmlu	0.6543	0.7300	192.0	1.000	0.0933
a192	Qwen3-4B	arc	0.8600	0.9100	192.0	1.000	0.0824
a192	Granite-3.3-2B	mmlu	0.5371	0.5571	182.2	0.807	0.0561
a192	Granite-3.3-2B	arc	0.7075	0.7600	172.9	0.688	0.0759
a384	Qwen3-4B	mmlu	0.6543	0.7486	384.0	1.000	0.0913
a384	Qwen3-4B	arc	0.8600	0.9225	384.0	1.000	0.0809
a384	Granite-3.3-2B	mmlu	0.5371	0.5943	283.6	0.260	0.0708
a384	Granite-3.3-2B	arc	0.7075	0.7750	234.1	0.068	0.0842
a512	Qwen3-4B	mmlu	0.6543	0.7514	512.0	1.000	0.0865
a512	Qwen3-4B	arc	0.8600	0.9125	512.0	1.000	0.0757
a512	Granite-3.3-2B	mmlu	0.5371	0.5971	304.0	0.081	0.0688
a512	Granite-3.3-2B	arc	0.7075	0.7775	238.7	0.018	0.0838
b096	Qwen3-4B	mmlu	0.6543	0.6571	96.0	1.000	0.0776
b096	Qwen3-4B	arc	0.8600	0.8650	96.0	1.000	0.0842
b096	Granite-3.3-2B	mmlu	0.5371	0.5471	95.6	0.969	0.0495
b096	Granite-3.3-2B	arc	0.7075	0.6775	95.1	0.938	0.0979
b384	Qwen3-4B	mmlu	0.6543	0.7414	384.0	1.000	0.0993
b384	Qwen3-4B	arc	0.8600	0.9150	384.0	1.000	0.0735
b384	Granite-3.3-2B	mmlu	0.5371	0.5814	281.5	0.296	0.0660
b384	Granite-3.3-2B	arc	0.7075	0.7800	249.4	0.128	0.0736

Accuracy is full-coverage branch accuracy; alignment uses q=0.25 and coverage 0.80. All tested budgets and macro contrasts follow.

H.1 Query 0, coverage 0.8

Setting	Contrast	Change	95% CI	99% CI
a096	full − no V	0.0017	[-0.0052, 0.0086]	[-0.0074, 0.0102]
a192	full − no V	0.0017	[-0.0052, 0.0086]	[-0.0074, 0.0102]
a192	setting − A96	0.0000	[0.0000, 0.0000]	[0.0000, 0.0000]
a384	full − no V	0.0017	[-0.0052, 0.0086]	[-0.0074, 0.0102]
a384	setting − A96	0.0000	[0.0000, 0.0000]	[0.0000, 0.0000]
a512	full − no V	0.0017	[-0.0052, 0.0086]	[-0.0074, 0.0102]
a512	setting − A96	0.0000	[0.0000, 0.0000]	[0.0000, 0.0000]
b096	full − no V	0.0017	[-0.0052, 0.0086]	[-0.0074, 0.0102]
b096	setting − A96	0.0000	[0.0000, 0.0000]	[0.0000, 0.0000]
b384	full − no V	0.0017	[-0.0052, 0.0086]	[-0.0074, 0.0102]
b384	setting − A96	0.0000	[0.0000, 0.0000]	[0.0000, 0.0000]

H.2 Query 0, coverage 1

Setting	Contrast	Change	95% CI	99% CI
a096	full − no V	0.0000	[0.0000, 0.0000]	[0.0000, 0.0000]
a192	full − no V	0.0000	[0.0000, 0.0000]	[0.0000, 0.0000]
a192	setting − A96	0.0000	[0.0000, 0.0000]	[0.0000, 0.0000]
a384	full − no V	0.0000	[0.0000, 0.0000]	[0.0000, 0.0000]
a384	setting − A96	0.0000	[0.0000, 0.0000]	[0.0000, 0.0000]
a512	full − no V	0.0000	[0.0000, 0.0000]	[0.0000, 0.0000]
a512	setting − A96	0.0000	[0.0000, 0.0000]	[0.0000, 0.0000]
b096	full − no V	0.0000	[0.0000, 0.0000]	[0.0000, 0.0000]
b096	setting − A96	0.0000	[0.0000, 0.0000]	[0.0000, 0.0000]
b384	full − no V	0.0000	[0.0000, 0.0000]	[0.0000, 0.0000]
b384	setting − A96	0.0000	[0.0000, 0.0000]	[0.0000, 0.0000]

H.3 Query 0.25, coverage 0.8

Setting	Contrast	Change	95% CI	99% CI
a096	full − no V	-0.0056	[-0.0097, 0.0103]	[-0.0135, 0.0134]
a192	full − no V	0.0017	[-0.0097, 0.0090]	[-0.0144, 0.0124]
a192	setting − A96	0.0052	[-0.0075, 0.0156]	[-0.0117, 0.0182]
a384	full − no V	-0.0003	[-0.0116, 0.0086]	[-0.0155, 0.0116]
a384	setting − A96	0.0169	[0.0016, 0.0275]	[-0.0042, 0.0309]
a512	full − no V	-0.0002	[-0.0124, 0.0089]	[-0.0167, 0.0131]
a512	setting − A96	0.0147	[0.0002, 0.0268]	[-0.0047, 0.0305]
b096	full − no V	0.0019	[-0.0103, 0.0108]	[-0.0155, 0.0140]
b096	setting − A96	-0.0002	[-0.0185, 0.0068]	[-0.0230, 0.0108]
b384	full − no V	-0.0019	[-0.0155, 0.0075]	[-0.0205, 0.0106]
b384	setting − A96	0.0112	[-0.0061, 0.0220]	[-0.0105, 0.0260]

H.4 Query 0.25, coverage 1

Setting	Contrast	Change	95% CI	99% CI
a096	full − no V	-0.0075	[-0.0123, 0.0055]	[-0.0166, 0.0079]
a192	full − no V	-0.0012	[-0.0090, 0.0052]	[-0.0117, 0.0069]
a192	setting − A96	0.0143	[0.0009, 0.0273]	[-0.0035, 0.0316]
a384	full − no V	-0.0004	[-0.0103, 0.0061]	[-0.0132, 0.0082]
a384	setting − A96	0.0312	[0.0154, 0.0448]	[0.0114, 0.0498]
a512	full − no V	0.0003	[-0.0118, 0.0060]	[-0.0146, 0.0083]
a512	setting − A96	0.0315	[0.0141, 0.0445]	[0.0099, 0.0487]
b096	full − no V	0.0029	[-0.0110, 0.0096]	[-0.0157, 0.0129]
b096	setting − A96	-0.0118	[-0.0322, -0.0022]	[-0.0377, 0.0019]
b384	full − no V	-0.0053	[-0.0162, 0.0041]	[-0.0190, 0.0072]
b384	setting − A96	0.0205	[0.0042, 0.0366]	[-0.0005, 0.0436]

H.5 Phi-3.5-mini

Task	Direct	Deliberated	Corrects	Corrupts	Controller	Scrambled	Alignment
mmlu	0.6771	0.5771	0.0671	0.1671	0.7518	0.6758	0.0760
arc	0.8750	0.7125	0.0400	0.2025	0.9219	0.8697	0.0522

Every Phi trace reaches the 96-token cap. The branch loses accuracy on both tasks, yet state alignment remains beneficial to the selective controller. This separates alignment from the value of querying. The full-minus-no-verifier comparison at q=0.25, coverage 0.80 is unresolved after development refitting; all four budgets are available in the accompanying scientific data.

I Decision credits across prompt and branch-length settings

We use all 30,000 branch records from the six two-model settings in Appendix H. Each setting has 700 MMLU and 700 ARC development questions per model and 700 MMLU plus 400 ARC evaluation questions. Question identities are shared across models and settings. We use the same logistic predictors, feature order, fit seed (20260904), exact-budget tie rules and within-choice-count state permutations as in the setting comparisons.

For each of the four model/task cells, 2,000 permutations estimate the four decision-intervention means at $q = 0.25, c = 0.80$. Credits use the path-averaged decomposition in Section 5.2. The 2,000 question-bootstrap draws share question indices across models within each benchmark and average 20 within-choice-count permutations in each draw. These intervals condition on development fits. The 99.167% intervals adjust for six settings within each credit family.

Table I1. Complete macro decision credits in accuracy percentage points. Query and answer credits sum to the joint contrast.

Setting	Query credit [99.167% CI]	Answer credit [99.167% CI]	Joint contrast
A/96	1.12 [0.13, 2.11]	6.35 [5.15, 7.51]	7.47
A/192	1.48 [0.36, 2.65]	6.23 [5.03, 7.40]	7.71
A/384	2.12 [1.06, 3.35]	6.07 [4.88, 7.19]	8.19
A/512	2.00 [0.85, 3.18]	5.89 [4.77, 7.01]	7.89
B/96	0.84 [0.02, 1.57]	6.90 [5.66, 8.21]	7.74
B/384	1.52 [0.28, 2.73]	6.31 [5.12, 7.36]	7.83

For the hindsight ceiling, let D count direct-correct items, G count wrong-to-right changes and N count items whose binary correctness is unchanged by deliberation. With exactly k queries, the maximum number of correct selected branches is $D + \min(k, G) - \max(0, k - G - N)$. With exactly a returned answers, oracle accuracy is the smaller of this count and a, divided by a. This rule is checked against exhaustive query/answer allocation on small binary branch banks. The ceiling assumes knowledge of both outcomes for every question.

Table I2. Fitted and oracle accuracies (%) at 25% queries. Each pair uses the stated common answer coverage.

Setting	Policy, 80% coverage	Oracle, 80% coverage	Policy, full coverage	Oracle, full coverage
A/96	76.94	93.15	71.18	77.58
A/192	77.47	95.00	72.61	79.56
A/384	78.63	96.56	74.30	81.81
A/512	78.42	96.79	74.33	82.05
B/96	76.92	91.96	70.00	76.13
B/384	78.06	96.03	73.23	81.38

Table I3. Direct paired answer-minus-query credit in accuracy percentage points. Each draw subtracts its query credit from its answer credit. The adjusted column covers this six-setting contrast family, conditional on the development fits.

Setting	Answer minus query credit [95% CI]	Six-setting-adjusted 99.167% CI
A/96	5.23 [4.16, 6.34]	[3.83, 6.74]
A/192	4.75 [3.57, 5.90]	[3.17, 6.27]
A/384	3.96 [2.73, 4.98]	[2.21, 5.35]
A/512	3.89 [2.80, 5.05]	[2.49, 5.31]
B/96	6.07 [4.97, 7.28]	[4.65, 7.58]
B/384	4.79 [3.70, 5.89]	[3.35, 6.28]

The supplement provides 2,000 paired draws per setting. At A/96, the paired difference is 5.23 [six-setting-adjusted CI 3.83, 6.74] points. These intervals use the resampling scheme defined above; Table 2 uses the scheme in Appendix F.